universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Assessing risk factors for depression during the second wave of COVID-19 in Japan: A machine learning approach

verfasst von | submitted by

Julian Weinhuber BSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2026

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 840
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Psychologie
Betreut von | Supervisor:	Univ.-Prof. Dr. Frank Scharnowski MSc
Mitbetreut von | Co-Supervisor:	Dipl.-Ing. Dr.techn. David Steyrl

Abstract

This thesis replicates and extends the study of Fukase et al. (2021), which examined depression, risk factors, and coping strategies during the second wave of the COVID-19 pandemic in Japan. Using the publicly available dataset, descriptive and inferential analyses of the original study were successfully reproduced, with only minor deviations likely due to rounding errors or software differences. In the process, undocumented analytical decisions, such as the exclusion of more than one fifth of participants and several coping strategies, were identified, highlighting the importance of transparent reporting.

Beyond replication, this thesis applies a Machine Learning framework to the same data. Logistic and linear regression models were implemented within a predictive pipeline using nested cross-validation and hyperparameter optimization, and their performance was compared to Gradient Boosted Decision Trees (LightGBM). The results show that the LightGBM models achieved modest but consistent improvements in predictive accuracy over the traditional regression models.

The findings strengthen confidence in the original study, highlight the methodological trade-offs between traditional statistical and modern Machine Learning approaches, and emphasize the value of integrating rigorous replication practices with predictive modeling. This work contributes both to the reliability of psychological research and to its methodological development.

Zusammenfassung

Diese Masterarbeit repliziert und erweitert die Studie von Fukase et al. (2021), die Depression, Risikofaktoren und Bewältigungsstrategien während der zweiten Welle der COVID-19-Pandemie in Japan untersuchte. Unter Verwendung des öffentlich verfügbaren Datensatzes konnten die deskriptiven und inferenzstatistischen Analysen der Originalstudie erfolgreich reproduziert werden, mit nur geringfügigen Abweichungen, die wahrscheinlich auf Rundungsdifferenzen oder unterschiedliche Software zurückzuführen sind. Im Zuge der Replikation wurden nicht dokumentierte analytische Entscheidungen identifiziert, wie der Ausschluss von mehr als einem Fünftel der Teilnehmenden sowie mehrerer Bewältigungsstrategien, was die Bedeutung transparenter Dokumentation unterstreicht.

Über die Replikation hinaus wendet diese Arbeit ein Machine-Learning-Framework auf denselben Datensatz an. Logistische und lineare Regressionsmodelle wurden in einer prädiktiven Pipeline mit verschachtelter Kreuzvalidierung und Hyperparameter-Optimierung implementiert und ihre Leistung mit Gradient Boosted Decision Trees (LightGBM) verglichen. Die Ergebnisse zeigen, dass die LightGBM-Modelle moderate, aber konsistente Verbesserungen der Vorhersagegenauigkeit im Vergleich zu den traditionellen Regressionsmodellen erzielen konnten.

Die Ergebnisse stärken das Vertrauen in die ursprüngliche Studie, verdeutlichen die methodischen Abwägungen zwischen traditionellen statistischen und modernen Machine-Learning-Ansätzen und betonen den Wert der Integration strenger Replikationspraktiken mit prädiktiver Modellierung. Diese Arbeit leistet somit einen Beitrag sowohl zur Zuverlässigkeit psychologischer Forschung als auch zu deren methodischer Weiterentwicklung.

Declaration on the Use of Artificial Intelligence (AI)

I hereby declare that I have used the following AI tools in the preparation of my Master's thesis entitled *Assessing Risk Factors For Depression During The Second Wave Of COVID-19 In Japan: A Machine Learning Approach*. I confirm that the disclosure of the AI tools used is complete, including their purpose and product names, as well as example prompts or outputs. I take full responsibility for the selection and adoption of all results generated by the AI tools I have used.

Purpose of Use	**AI Tool(s)**	**Example Prompt / Output**
Writing Literature research Feedback Idea development	GPT-3.5 GPT-4o GPT-4.1 GPT-4.5 GPT-5	”Find sources that are relevant in the context of Gradient Boosted Decision Trees” ”Adapt this table to APA style” ”What do you think about this paragraph? Do you suggest any improvements?”
Coding assistance in Visual Studio Code	GPT-4o GPT-4.1 Claude Sonnet 3.5 Claude Sonnet 4	”Add StandardScaler to all machine learning models for methodological consistency” ”Fix the singular matrix error in Box-Tidwell test with proper error handling”
Writing support in Overleaf	Writefull	Automatic grammar, style, and phrasing suggestions in text drafts

Note: Tools used solely for grammar and spell checking, or for the automated creation of the bibliography, are excluded from this declaration.

Contents

Chapter 1

Introduction

Depression is a mood disorder that affects approximately 280 million people worldwide (Cuijpers et al., 2025). It causes a persistent feeling of sadness and loss of interest (Chand et al., 2023). Common symptoms include sadness, emptiness, and irritable mood, and can be accompanied by somatic and cognitive changes that affect the individual's ability to function normally (Chand et al., 2023).

The mortality risk for people suffering from depression is more than 20 times higher than in the general population (Lépine & Briley, 2011). It is also a risk factor for cardiovascular death and is associated with functional impairment and disability. In addition, it increases the risk of reduced productivity and absence from the workplace, which can lead to lower income or unemployment (Lépine & Briley, 2011).

Even if the disorder can be successfully treated, it still imposes a considerable burden on the affected person. Most patients retain residual symptoms such as cognitive impairment and social dysfunction, and the risk of relapse can reduce quality of life. Depression in a parent and especially maternal depression can affect the development and health of the baby and have long-term consequences on its mental health (Lépine & Briley, 2011). An estimated 40% of children born to mothers with depression develop depression or other mental health problems (Farahzadi, 2017).

Depression is also the most common psychiatric disorder in people who commit suicide

(Hawton et al., 2013). Approximately nine out of ten individuals appear to have had a psychiatric disorder at the time of their death, and depression is the most common of these disorders (Hawton et al., 2013). Half to two thirds of people who commit suicide are estimated to suffer from depression. Although it is a very common disorder, the detection of individuals at risk of suicide can be difficult.

Given its widespread impact, depression has been a critical focus in public health research. These concerns became especially urgent during the COVID-19 pandemic, when social isolation, economic uncertainty, and health-related stressors intensified mental health challenges around the world. The following sections will examine depression in the context of COVID-19, with particular attention to the situation in Japan, and discuss the importance of replication in psychological research.

1.1 COVID-19 and Mental Health in Japan

The outbreak of COVID-19 created unprecedented challenges for public health systems and disrupted everyday life around the world. Lockdowns and limits on social contact interrupted routines, increased isolation, and generated stressors such as fear of infection and concern for the health of loved ones, all of which are known risk factors for anxiety and depression (Choi et al., 2020). Global estimates suggest that in 2020 the number of cases of major depressive disorder increased by 27.6% and anxiety disorders by 25.6% (Santomauro et al., 2021). However, longitudinal studies indicate that these estimates may exaggerate the long-term impact. Symptoms of anxiety and depression rose sharply in the early months of the pandemic, but for many individuals they declined again and returned to pre-pandemic levels after a few months (Daly & Robinson, 2022; Robinson et al., 2022). However, certain groups, such as health workers, women, young adults, and those facing financial difficulties, showed more persistent mental health problems (Olaya et al., 2021; Racine et al., 2021).

Japan provides an important case for studying these developments. Before the pandemic, the prevalence of Major Depressive Disorder (MDD) was consistently lower than in many western countries, with 12-month rates between 1–2% and lifetime rates between 3–7% (Kawakami, 2007). More recent surveys estimate a lifetime prevalence

of 5.7% and a 12-month prevalence of 2.7% between 2013 and 2015 (Cho et al., 2021). Broader surveys of mental disorders reported a lifetime prevalence of 20.3% and a 12-month prevalence of 7.6%, with MDD among the most common disorders (Ishikawa et al., 2016). At the same time, treatment rates in Japan have been low. Among those with major depression, only about 27% sought medical help and only about 14% consulted a psychiatrist (Kawakami, 2007).

During the pandemic, surveys in Japan showed that psychological distress increased even under relatively mild lockdown conditions (Yamamoto et al., 2020). Economic insecurity further contributed to mental health problems, particularly among workers experiencing income loss and among those with low social support (Hori et al., 2025). University students reported worsening mental health in 2021, although some recovery was observed in 2022 (Nagib et al., 2023). Suicide rates in Japan followed a complex pattern: deaths initially declined in early 2020 but increased later that year, especially among women (Ueda et al., 2022).

These findings suggest that while Japan mirrored the global pattern of an initial increase in symptoms followed by partial recovery, the effects of the pandemic varied strongly between subgroups. Given the historically low prevalence of depression in Japan and the large treatment gap, it is important to examine more closely how the pandemic affected mental health in this context.

1.2 The Importance of Replication

Replication plays a central role in the scientific process. It enables researchers to verify empirical findings, assess the reliability of theoretical claims, and accumulate robust knowledge over time. In psychological science, replication is particularly relevant, as human behavior is complex and variable (Nosek et al., 2022).

Despite the foundational importance of replication, psychology has faced a profound replication crisis in recent decades (Korbmacher et al., 2023). This crisis has cast doubt on the robustness of findings in psychological research while at the same time stimulating reform efforts aimed at transparency and reproducibility (Korbmacher et al., 2023). A very important study in the replication debate was the publication of the OpenScience

Collaboration's large-scale replication project in 2015, which systematically tested the reproducibility of findings in psychological science (Collaboration, 2015). The project attempted to replicate 100 experimental findings from studies published in 2008 in one of three high-ranking psychological journals. Each study was subjected to high-fidelity replication, including consistent methods, materials, and analytical strategies.

The results showed a great discrepancy in the originally reported findings and those made in the replication. Although 97% of the original studies reported statistically significant results, only 36% of the replication attempts reported significant results. The average effect size of the replication attempts was approximately half the magnitude of the original findings (r = 0.197 replication vs. r = 0.403 original). These discrepancies occurred despite rigorous efforts to follow the original protocols and maintain methodological fidelity. The researchers conclude the study like this: "A large portion of replications produced weaker evidence for the original findings despite using materials provided by the original authors, review in advance for methodological fidelity, and high statistical power to detect the original effect sizes." (Collaboration, 2015).

These findings are consistent with broader concerns that psychological science has systematically produced inflated effect sizes and results that fail to generalize. As Hengartner (2018) argues, structural issues such as publication bias and selective reporting have contributed to an accumulation of evidence that often overstates the strength of psychological phenomena. However, the authors of the OpenScience Collaboration (Collaboration, 2015) emphasize that the goal of replication is not to disprove previous findings but to distinguish robust results from those that are more dependent on sampling variation or contextual specificity. They also introduced the idea that reproducibility can be viewed as a continuous measurement, not as a binary distinction of success or failure. Instead, attention should be shifted to the analysis of effect sizes, confidence intervals, and broader patterns of evidence across studies.

In the context of clinical and applied psychology, the implications are especially significant. Many treatment recommendations and public health strategies are derived from empirical findings whose reproducibility may not have been tested. Tackett et al. (2019) highlight that many foundational findings used to inform diagnostics and interventions have not undergone systematic replication, which might erode their reliability

and applicability. They argue for a stronger adoption of open science practices, such as preregistration and data sharing, to improve the robustness of clinical research. Furthermore, they note that replication is especially important in clinical psychology given the potential real-world consequences of unstable or overstated findings. In the case of mental health studies that focus on the context of pandemics, such as the study by Fukase et al. (2021) discussed in this thesis, rigorous replication is essential to ensure that interventions are grounded in reliable data analysis.

1.3 The Original Study by Fukase et al. (2021)

This thesis seeks to replicate and extend the findings of a study conducted by Fukase et al. (2021), which examined the psychological impact of the COVID-19 pandemic in Japan during its second wave. The study aimed to assess the prevalence of depressive symptoms, identify associated risk factors, and evaluate coping strategies among the general population. The authors surveyed 2,764 participants in an online questionnaire that included standardized measures such as the Patient Health Questionnaire-9 (PHQ-9) for depressive symptoms (Kroenke et al., 2001), the State–Trait Anger Expression Inventory (STAXI) for state anger and anger control, and the Brief Coping Orientation to Problems Experienced (Brief COPE) for coping strategies.

Fukase et al. (2021) reported a high prevalence of depressive symptoms, with approximately 18.4% of participants scoring above the clinical threshold of $PHQ \geq 10$. They also identified several risk factors for depression, including existing disease, younger age, lower income, and unemployment. In addition, they found that adaptive coping strategies, such as planning and use of instrumental support, were associated with lower depression scores, while coping strategies such as behavioral disengagement and self-blame were associated with higher depression scores.

The authors concluded that the social dislocations caused by the pandemic had a measurable impact on mental health in Japan. They emphasized the importance of identifying protective factors that could inform interventions.

1.4 Research Questions & Hypotheses

Fukase et al. (2021) made their anonymised dataset publicly available (Fukase, 2020), which constitutes good scientific practice in accordance with open science recommendations (Grant et al., 2022). This thesis will use the available data and try to verify and extend the existing analysis. It is based on several research questions and hypotheses.

- **Research question 1**: Are the descriptive findings of Fukase et al. (2021) reproducible? This involves verifying descriptive statistics such as the prevalence of depression and the distributions of variables. It also involves simple inference statistics such as t-tests for continuous variables and chi-square-tests for categorical variables. Essentially, this research question is related to the replication of Additional Table 1 of the original study.
  - **Hypothesis 1.1**: The results of the descriptive analysis will be consistent with the findings of the previous study, with a precision of two decimal points.
  - **Hypothesis 1.2**: The results of the simple inference statistics will be consistent with the findings of the previous study with a precision of two decimal points.
- **Research question 2**: Are the inferential findings of Fukase et al. (2021) reproducible? This involves verifying whether the preconditions for a logistic regression are met. It also involves verifying the results of the logistic regression, particularly the protective factors and risk factors identified by the Odds Ratios (ORs). Essentially, this research question is related to the replication of Additional Table 2 of the original study.
  - **Hypothesis 2**: The results of the inferential logistic regression analysis will be consistent with the findings of the previous study, with a precision of two decimal points.
- **Research question 3**: When using a machine learning approach to evaluate the performance of predictive models, how do logistic regression and linear regres-

sion compare? What changes when the depression score is not dichotomized into a categorical variable but is used as a continuous variable?

  - **Hypothesis 3.1**: The logistic regression model and the linear regression model identify the same set of predictors as statistically significant.

- **Research question 4**: Can a more sophisticated machine learning model that allows non-linear relationships, such as Gradient Boosted Decision Trees (GBDT), provide better accuracy in predicting depression than the simpler regression models?

  - **Hypothesis 4**: The GBDT models for classification and regression perform at least as well as the traditional models. For classification, performance is measured by accuracy and Area Under The Receiver Operating Characteristic Curve (AUC). For regression, performance is measured by R^2 and Root Mean Squared Error (RMSE).

The first two research questions address the replication of existing results, whereas the last two research questions extend the analysis to new methodological territory. Answering these questions will help assess the reliability of the findings of Fukase et al. (2021) and potentially gain new insights by using more sophisticated analytical models.

1.5 Software

All analyses were performed in Python 3.12.3 (Van Rossum, Drake et al., 1995). Data handling relied on pandas (McKinney, 2010; team, 2025) and NumPy (Harris et al., 2020), while statistical modeling used statsmodels (Seabold & Perktold, 2010) and SciPy (Virtanen et al., 2020). Machine learning was implemented with scikit-learn (Pedregosa et al., 2011) and Microsoft's Light Gradient Boosting Machine (LightGBM) library (Ke et al., 2017; Shi et al., 2025), with hyperparameter tuning via Optuna (Akiba et al., 2019). Visualizations were created using matplotlib (Hunter, 2007) and seaborn (Waskom, 2021), and Excel data was imported with openpyxl (Gazoni & Clark, 2024). The versions of these packages and the full list of installed packages can be found in Appendix A.

1.6 Structure

The remainder of this thesis is organized into four chapters that build on each other sequentially. Chapter 2 explains the replication analysis in more detail. The descriptive and inferential methods are explained and the methodological shortcomings of the original analysis by Fukase et al. (2021) are discussed. Chapter 3 introduces the predictive modeling approach. The Machine Learning (ML) approach is defined and methodological concepts such as pipelines, penalization, and non-linear interactions are discussed. Chapter 4 displays the results of the replication and advanced analyses. Descriptive and comparative metrics are reported. Chapter 5 contextualizes the results and critically reflects on the methodological limitations. The different approaches and models are discussed. Chapter 6 highlights the contributions of this thesis, offers recommendations for data analysis, and outlines directions for future research. Tables and figures that are relevant for this thesis but could not be included in the main part due to lack of space can be found in Appendix A.

Chapter 2

Replication of the analysis by Fukase et al. (2021)

2.1 Introduction to the Replication

The following chapter details a replication study of the methods and findings of Fukase et al. (2021), titled "Depression, risk factors, and coping strategies in the context of social dislocations resulting from the second wave of COVID-19 in Japan". The original study was a web-based survey conducted in Japan in July 2020 (during the second wave of COVID-19) with 2,708 adult participants from 13 prefectures under special pandemic precautions. The authors collected comprehensive data on demographics (for example, age, sex, employment status, income), pandemic-related factors, mental health measures, and coping behaviors. They measured depression using the PHQ-9 (Kroenke et al., 2001), as well as state anger and anger control using the STAXI (Spielberger, 1988) and the Brief COPE (Carver, 1997). The purpose of the original study was to identify risk factors associated with probable depression and to suggest coping strategies to manage prolonged pandemic-related distress. To this end, Fukase et al. (2021) performed descriptive analyses to compare participants with and without depression, followed by a multivariate logistic regression to identify significant predictors of depressive symptoms.

Replication studies play a critical role in the scientific process by verifying and validating previous findings (Ioannidis, 2005). The purpose of this replication of the analyses of Fukase et al. (2021) is to confirm the reliability of their results and to evaluate the robustness of their methodological choices. The original open-access dataset is available on OPENICPSR (Fukase, 2020). Throughout this chapter, I will highlight any methodological issues encountered and suggest improvements to enhance replicability, generalizability, and statistical rigor.

2.2 Replication of Descriptive Analysis

2.2.1 Data Overview

I first replicated the descriptive statistics of the original study. The dataset includes demographic attributes, pandemic impact indicators, psychological scales, and coping strategy measures. An overview of the demographic variables can be found in Table 2.1

Table 2.1
Key demographic variables

Variable	Description
Age	Age in years
Sex	Male / Female
Employment status	Full-time, homemaker, not working, no regular employment
Residential areas	13 prefectures
Underlying disease	Without / With
Marital status	Married / Single
Household income	$<$ 2 million JPY; 2–8 million JPY; $>$ 8 million JPY; I do not know; non-response
Economic impact	Without; negative; positive

The underlying disease variable included diseases with a higher risk of a more severe infection, such as taking immunosuppressive medications, as well as underlying medical conditions, such as heart disease (Fukase et al., 2021).

Depression was assessed using the Japanese version of the PHQ-9 questionnaire, which consists of 9 items on a four-point scale (each 0 to 3) summed to a total score between

0 and 27 (Fukase et al., 2021). Consistent with Fukase et al. (2021), I defined "probable depression" as a PHQ-9 score $\geq$ 10. Participants scoring 9 or below were considered non-depressed, participants scoring 10 or above were considered probably depressed. Of the 2708 participants, 18.35% fell into the probable depression group, while 82.65% were considered not depressed. Thus, the overall prevalence of elevated depression symptoms in this dataset is much higher than the prevalence of around 2.0% that other studies suggest (Demiya et al., 2022).

State anger and anger control were measured with the Japanese version of the STAXI (Spielberger, 1988). The Japanese version of the STAXI was validated by Suzuki and Haruki (1994). The state anger scale consists of 10 items on a four-point scale (0 to 3) and a score of 0 to 27 in total. On this scale, higher scores indicate higher state anger. The anger control scale consists of 7 items on a four-point scale (0 to 3) and a score of 0 to 21 overall. On this scale, a higher score indicates a greater attempt to keep calm and restrain one's behavior.

Coping strategies were measured with the Japanese version of the Brief COPE (Carver, 1997). The Japanese version of the Brief COPE was validated by Otsuka et al. (2009). The question "how do you feel and deal with troubling unpleasant things in daily life" was adapted by Fukase et al. (2021) to "how do you feel and deal with social dislocations resulting from COVID-19 infection" to focus on the context of COVID-19. The Brief COPE consists of 28 items on a four-point scale (1 to 4). Since each coping style is evaluated by two items, the score for each coping style can range from 2 to 8. On this scale, a higher score indicates higher levels of the corresponding coping style. The Brief COPE measures 14 coping styles: Self-distraction, active coping, denial, substance use, use of emotional support, use of instrumental support, behavioural disengagement, venting, positive reframing, planning, humour, acceptance, religion, and self-blame.

2.2.2 Replication of Additional Table 1

I started my analysis by replicating the "Additional file 1: Table S1. Participant demographic characteristics, information related to social dislocations resulting from the COVID-19 pandemic, and scores from the PHQ-9, State Anger, Anger Control, and Brief COPE." (Fukase et al., 2021). This table contains mostly descriptive statistics,

along with some inferential statistics for group comparisons.

For the categorical variables Sex, Employment status, Residential areas, Underlying disease, Married, Household income, and Economic impact, I computed absolute and relative frequencies, clustered by depression. I also performed χ^2-tests to test the independence of categorical variables and depression. The χ^2-test can be used to check whether there is a significant association between two categorical variables (Pearson, 1900). It compares the observed frequencies with the frequencies expected if the variables were independent. This is done by calculating the differences between the observed and expected counts and summing them in a standardized way. The resulting value, the χ^2-statistic, is then compared with a critical value of the χ^2-distribution to assess significance. A larger difference results in a larger χ^2-statistic, which suggests that the variables are not independent.

For the metric variables Age, PHQ-9, State anger, Anger control, and each coping style, I computed the mean and standard deviation, clustered by depression. For each metric variable, I also conducted independent samples t-tests to compare the non-depression group and the probable depression group. The independent samples t-test is used to compare the means of two independent groups (Student, 1908). The test evaluates whether the difference in group means is greater than would be expected by chance, taking into account the variability within each group, as well as the sample sizes.

The independent samples t-test assumes that

1. **Independence of samples**: The observations in one group are independent of the observations in the other group

2. **Normality**: Both samples are approximately normally distributed

3. **Homoscedasticity**: Both samples have approximately the same variance

2.3 Replication of the Logistic Regression

2.3.1 Introduction to Logistic Regression

The logistic regression is a statistical modeling technique for predicting a binary outcome (Cox, 1958). It models the log-odds of the outcome as a linear combination of the predictor variables.

In mathematical terms, the logistic regression model can be written as

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k \tag{2.1}$$

where p is the probability of being in the probable depression group and $x_1, x_2, \ldots, x_k$ are the predictor variables. The coefficients $\beta_0, \beta_1, \ldots, \beta_k$ are estimated from the data. Different estimators can be used to compute these coefficients, one of the most common being Maximum Likelihood Estimation (MLE). Exponentiating a coefficient yields an Odds Ratio (OR), which is easier to interpret: An OR above 1 indicates that the predictor is associated with higher odds of depression, that is, it is a risk factor for depression. An OR below 1 indicates that the predictor is associated with lower odds of depression, that is, it is a protective factor. In the logistic regression analysis of the original study, the odds ratio for being married was around 0.53. This means that married participants were about 50% less likely to be categorized as having probable depression than those who were not married.

2.3.2 Shortcomings of the Original Logistic Regression Analysis

Exclusion of Participants and Coping Strategies

During the replication, it became evident that not all participants and coping strategies were included in the logistic regression analysis. First, participants who selected ”I do not know” or ”Non-response” as their Household income were excluded from the analysis. Together, these two groups represented 21.64% of the original sample (586 participants). This exclusion is not documented in the original article by Fukase et al. (2021), but I was only able to replicate their reported results by applying it. The

removal of such a substantial portion of the sample reduces statistical power and may have introduced bias if these values were systematically associated with socioeconomic status or other relevant factors and therefore should have been explicitly mentioned in the article.

Second, not all the coping strategies of Brief COPE were considered. The following eight strategies were excluded from the regression results and thus almost certainly from the logistic regression analysis itself: Acceptance, Active coping, Emotional support, Positive reframing, Religion, Self-distraction, Substance use, and Venting. I discovered this during the replication process when reconstructing the set of predictors included in the logistic regression models. Their omission is not addressed in the original study, yet it narrows the scope of coping strategies considered as potential protective or risk factors for depression.

These exclusions highlight an important limitation in the transparency of the original analysis. Without explicit documentation, key preprocessing decisions such as the treatment of missing values and the exclusion of predictors are difficult to assess. To remain consistent with Fukase et al. (2021), I therefore excluded the same participants and coping strategies from my analyses, including the replication of their logistic regression, as well as the linear regression and LightGBM models.

Binarization of the Outcome

The original study by Fukase et al. (2021) assessed the depression score using the PHQ-9, a validated self-report questionnaire comprising 9 items (Kroenke et al., 2001). The score for each item can range from 0 to 3. Thus, the cumulative score of the questionnaire ranges from 0 to 27. This continuous score represents varying levels of severity of depression. The authors chose to dichotomize this continuous score into two distinct categories: Probable depression and not probable depression. A threshold of PHQ-9 $\geq$ 10 was used, which had been established in previous studies (Fukase et al., 2021). Participants with a PHQ-9 score of 10 or above were classified as probably depressed, while those with a score of 9 or less were classified as probably not depressed.

The cutoff value of a PHQ-9 score of 10 is clinically meaningful and is described in the

original PHQ-9 validation paper by Kroenke et al. (2001). It simplifies the interpretation of the analysis and helps to clearly communicate the results. However, binarization also leads to loss of information about the severity of symptoms. Individuals with scores close to the threshold, e.g. 9 and 10, may differ only slightly in symptom severity, yet they are categorized into two distinct groups. In addition, the scores of individuals within the same group can vary. For example, a person with a PHQ-9 score of 9 might feel moderately depressed but is labeled as probably not depressed, while a person who reports very few symptoms with a score of 1 receives the same label. Binarization might also obscure nuanced risk-factor relationships. For example, certain predictors might strongly influence severe cases, but may be less relevant for moderate cases.

Given these limitations, I replicated the original logistic regression model using the established cutoff value of PHQ-9 ≥ 10 to allow for a direct methodological comparison. Recognizing the shortcomings of this binary classification, I will also explore alternative approaches such as linear regression and GBDT that use the continuous PHQ-9 score. This approach aims to identify subtle relationships, improve predictive performance, and improve understanding of the patterns underlying depression during Japan's second wave of COVID-19.

Standardization of Variables

Standardization is a preprocessing method that transforms variables to have a mean of 0 and a standard deviation of 1. This enables a direct comparison of the coefficients of different predictors that are measured on different scales.

Fukase et al. (2021) do not mention the use of standardization in their logistic regression analysis. SPSS, the statistical software they used for the analysis, does not standardize variables by default in the logistic regression analysis (IBM Corp., 2013). I was able to replicate the original results almost perfectly using unstandardized predictors, while using standardized predictors yielded different results. The results of the logistic regression using unstandardized predictors can be found in Table 4.2. Thus, it is reasonable to infer that unstandardized predictors were used in their original metric scale. This limits direct interpretability of the relative importance of predictors, especially for variables measured on different scales (for example, age versus coping strategies).

Standardization offers several advantages. It simplifies interpretability, because the standardized coefficients clearly reflect the predictor's strength regardless of the original unit. It also enhances numerical stability and reduces potential computational issues, especially when predictors have vastly different scales. In addition, standardization helps identify and address multicollinearity as the correlations among standardized variables become clearer.

Given these advantages, I use standardization in all analyses that are not part of the replication.

Feature Encoding

Feature encoding is a preprocessing step in regression and machine learning models that use categorical data. Since most algorithms, including logistic regression, linear regression, and LightGBM, require numerical input, categorical variables must be converted into a numerical format.

Fukase et al. (2021) do not provide detailed documentation on their variable encoding process. In the raw dataset, the categorical variables are already numerically encoded. However, for variables on an ordinal scale such as Economic influence, the encoding does not always reflect the correct order (see Table 2.2). In this case, *Negative economic influence* should be assigned a lower numerical value than *Without economic influence*, and *Without economic influence* should be assigned a lower numerical value than *Positive economic influence*. In addition, the intervals between the values are inconsistent.

Table 2.2
Feature encoding for predictor Economic influence

Categorical value	Numerical value
Without economic influence	0
Negative economic influence	1
Positive economic influence	3

For other categorical predictors, there is no natural order of the values. For example, the categories of Employment status (see Table 2.3) are qualitatively different and cannot be meaningfully arranged in order.

Table 2.3
Feature encoding for predictor Employment status

Categorical value	Numerical value
Full-time	0
Homemaker	1
Not working	2
No regular	3

Fukase et al. (2021) present the results of their logistic regression in Additional Table 2. This table contains one reference category for each categorical predictor, i.e. the value *Without economic influence* for the predictor Economic influence. These observations and the usage of SPSS 22 (IBM Corp., 2013) make it reasonable to assume that dummy variables were used to prepare the categorical predictors for the logistic regression analysis.

Dummy variables can be used to convert categorical data into a binary array. Each category is represented by a unique binary vector with the value 1 in the position corresponding to the category and 0 in the other positions. One of the categories is used as a reference category and contains 0 in all positions or is not represented in the binary array. This helps in avoiding multicollinearity. The dummy variables for the predictor Economic influence with reference value *Without economic influence* are shown in Table 2.4.

Table 2.4
Dummy variables for predictor Economic influence

Categorical value	Numerical value	Dummy 1	Dummy 2
With economic influence	1	1	0
Without economic influence (ref)	0	0	0
Moderate economic influence	3	0	1

Missing Regularization

In their analysis, Fukase et al. (2021) used a traditional multivariate logistic regression in SPSS Statistics 22 (IBM Corp., 2013). Regularization is not included in the standard

multivariate logistic regression in SPSS Statistics 22, and the paper does not mention the use of any regularization methods. Thus, I also replicated the analysis using a standard multivariate logistic regression without any regularization penalties, such as Lasso or Ridge regularization, for the model coefficients. I used the same predictor variables as in the original analysis and identified significant predictors through p-values.

Computing an unpenalized model ensures that coefficient estimates are purely data driven via likelihood maximization. While regularization can help prevent overfitting, it also biases coefficients towards zero. Since the original study did not mention regularization, all selected predictors are retained in the model, and their full effects are estimated.

Assumptions of Logistic Regression

Fukase et al. (2021) do not explicitly report checking the assumptions of logistic regression. The assumptions of logistic regression, as reported by Stoltzfus (2011), are the following:

1. Independence of observations: The outcomes of all participants must be independent. For example, outcomes are dependent if there are repeated measures or clustered data

2. Linearity in the logit: Continuous independent variables must have a linear relationship with the log-odds of the outcome

3. Absence of multicollinearity: Predictor variables should not be highly correlated with each other, as this inflates standard errors and destabilizes coefficient estimates

4. Lack of strongly influential outliers: Observations with a disproportionate influence on the model can distort regression estimates. Outliers should be detected using diagnostic statistics and retained or excluded according to their impact

The assumption of independence of observations holds by design. In the study by Fukase et al. (2021), each individual participated only once and self-reported on dif-

ferent variables. Variables that belong to the same scale, for example PHQ-9, were aggregated and used as one variable.

Linearity in the logit can be tested with the Box-Tidwell transformation (Box & Tidwell, 1962). An interaction term between the continuous predictor and its natural logarithm is included in the logistic regression model:

$$logit(p) = \beta_0 + \beta_1 X + \beta_2 (X \cdot \ln(X)) \tag{2.2}$$

If the interaction term is significant, the assumption of a linear relationship with the log-odds of the outcome is violated.

Multicollinearity is tested with the Variance Inflation Factor (VIF) (O'brien, 2007; Stoltzfus, 2011). The VIF measures how much the variance of a regression coefficient is inflated due to collinearity with other predictors. To determine the VIF, each predictor is regressed on all other predictors:

$$VIF_i = \frac{1}{1 - R_i^2} \tag{2.3}$$

where R_i^2 is the coefficient of determination from a regression of predictor X_i on all other predictors. A high R_i^2 means that the predictor X_i is highly predictable from the other predictors, which means multicollinearity. The VIF quantifies how much the variance of a regression coefficient is inflated due to multicollinearity. For example, a VIF of 10 indicates that the variance of that coefficient is 10 times larger than it would be if the corresponding predictor were not correlated with any other predictors in the model (O'brien, 2007).

Influential outliers are tested with Cook's distance. It is used to assess the influence of each observation on the regression model. It combines information about the residual, i.e. how far off the prediction is, and leverage, i.e. how unusual the predictor values are. Cook's distance is computed as follows:

$$D_i = \frac{\sum_{j=1}^{n} (\hat{y}_j - \hat{y}_{j(i)})^2}{p \cdot MSE} \tag{2.4}$$

where: D_i is Cook's distance for observation i, $\hat{y}_j$ is the predicted value of the dependent variable for observation j when observation i is removed, p is the number of predictor variables that include the constant term and Mean Squared Error (MSE) is the average of the squares of the residuals for the model.

Chapter 3

Using Machine Learning to Predict Depression

3.1 Machine Learning

It is difficult to provide a generally accepted definition of ML (Christodoulou et al., 2019). I will summarize some ideas and approaches and try to explain the core concept of ML, especially in contrast to traditional statistics. I want to note that there is no agreement on a clear boundary between statistical modeling and ML. For example, Christodoulou et al. (2019) do not consider penalized regression embedded in a ML pipeline as a ML algorithm because regression models do not fit their definition of ”automatic learning from data” (Christodoulou et al., 2019). However, other researchers, such as He and Garcia (2009), disagree and consider any method that deviates from basic regression models, in this case penalized regression, ML (He & Garcia, 2009).

In this thesis, I want to contrast the two approaches. I want to focus on the difference between the traditional explanatory approach and the ML predictive approach. I want to differentiate the two as clearly as possible. Thus, I will consider anything beyond basic regression models ML, in a similar fashion as He and Garcia (2009). This includes models that are computed with ML methods, such as cross-validation (CV) and hyper parameter optimization (HPO), even if the underlying model is usually not considered

a ML model. Specifically, I will consider the logistic regression and linear regression analyses conducted in this thesis ML, as they are built in a similar ML pipeline as the GBDT.

ML refers to a family of computation methods that focus on automatically identifying patterns in data to make accurate predictions. Rather than relying on explicitly defined statistical theories, ML methods build predictive functions by optimizing performance on real data, often without strong assumptions about the underlying data generation process (Yarkoni & Westfall, 2017a).

Yarkoni and Westfall (2017a) argue that psychological science has traditionally favored explanation over prediction. Researchers focus on identifying statistically significant relationships among variables, rather than building models that perform well on new data. In ML, the primary goal is generalization, not explanation. This leads to several methodological differences:

- CV ensures that the model's predictive ability is not overestimated
- Model complexity is controlled by regularization, not exclusion of variables based on non-significant p-values
- Evaluation is external, i.e. based on new data, rather than internal, i.e. based on fit to training data

This predictive framework is especially valuable in the field of psychology, where many variables are correlated and noisy. As Yarkoni and Westfall (2017a) remark, ML techniques offer a robust way to build models that can detect meaningful patterns even in such high-dimensional and messy data. However, they also emphasize that ML is not meant to replace theory-driven research, but rather to complement it with tools that offer more accurate predictions:

> "In much of psychology, researchers privilege theoretical understanding over concrete prediction. However, in many applied domains – for example, much of industrial-organizational psychology, educational psychology, and clinical psychology – achieving accurate prediction is often the primary

stated goal of the research enterprise." (Yarkoni & Westfall, 2017b)

3.2 Two Core Components of the ML Pipeline

The two core components that I am using in my ML pipeline that separate it from traditional statistics are CV and HPO.

CV is a statistical method commonly used in ML to evaluate the performance of predictive models. It involves splitting the dataset into several subsets, called folds. In a typical k-fold CV, the dataset is partitioned into k subsets. The model is trained on k-1 subsets and then tested on the remaining subset. This process is repeated k times, each time using a different subset as the test set. The final performance metric is averaged across all folds, providing a reliable estimate of the model's generalization to unseen data (Raschka, 2018).

HPO refers to the systematic process of finding the optimal settings of the hyperparameters. Hyperparameters are model parameters that are set before training begins, such as regularization strengths, learning rate, or tree depth. Unlike model parameters that are learned directly from data during training, hyperparameters cannot be derived analytically and must instead be tuned through empirical search strategies, e.g. grid search, random search, or Bayesian optimization. Effective hyperparameter optimization is essential, as poorly chosen hyperparameters can lead to suboptimal performance or overfitting (Yang & Shami, 2020).

Nested CV combines both CV and HPO into a robust framework. It involves two nested loops: An inner loop and an outer loop. The inner loop is used for hyperparameter tuning, employing CV to identify the best hyperparameter values. The outer loop assesses the generalization performance of the tuned model by evaluating it on data completely separate from the hyperparameter optimization stage. By keeping hyperparameter tuning and performance evaluation strictly separated, nested CV produces an unbiased estimate of the model's predictive capabilities, reducing the risk of overly optimistic results that are common in simpler evaluation strategies. I provide a Pseudocode implementation example of nested CV in Fig. 3.1. This Pseudocode resembles the structure of the Python code that I used in my analysis (Bradshaw et al., 2023).

```
for outer_fold in outer_cross_validation:
    outer_train, outer_test = split_data(fold=outer_fold,
        data=data)

    best_model = None
    best_score = float('-inf')

    for hyperparams in hyperparameter_grid:
        scores = []

        for inner_fold in inner_cross_validation:
            inner_train, inner_val = split_data(fold=inner_fold,
                data=test_data)

            inner_model = train_model(data=inner_train,
                hyperparameters=hyperparams)

            score = evaluate_model(model=inner_model,
                data=inner_val)

            scores.append(score)

        avg_score = mean(scores)

        if avg_score > best_score:
            best_score = avg_score
            best_hyperparams = hyperparams

    outer_model = train_model(data=training_data,
        hyperparameters=best_hyperparams)

    test_score = evaluate_model(model=outer_model,
        data=test_data)

    store(score=test_score, model=outer_model)
```

Figure 3.1: Nested CV Pseudocode

3.3 Advantages of Machine Learning

Machine learning offers several methodological and practical advantages over traditional statistical approaches. Whereas classical statistical models emphasize estimation, inference, and interpretability, machine learning prioritizes predictive accuracy, model generalization, and data-driven discovery. This is particularly relevant in psychology, where datasets are often high-dimensional, noisy, and collected from diverse sources.

As Shmueli (2010) and Breiman (2001) argue, statistical modeling in fields such as psychology has traditionally focused on causal explanation, with the assumption that models with a good explanatory fit will also predict well. However, explanatory fit does not necessarily translate into predictive power (Yarkoni & Westfall, 2017a). Explanatory modeling is designed to test hypotheses about mechanisms and estimate the effects of predictors on outcomes, typically within a theoretical framework and under distributional assumptions. Predictive modeling prioritizes the construction of models that are generalized to new data, regardless of interpretability.

This distinction has practical implications. Explanatory models are often evaluated on the same dataset used for estimation, which can yield overly optimistic accuracy. Predictive modeling requires a strict separation of training and test data and uses validation techniques such as cross-validation to obtain unbiased estimates of generalization performance (Shmueli, 2010).

However, it is important to acknowledge that machine learning is not universally superior. In some domains, traditional methods such as logistic regression can perform just as well as or even better than more complex algorithms (Christodoulou et al., 2019). Machine learning should be seen as a complement to classical statistical approaches. In this thesis, I want to expand on the traditional explanatory regression model used by Fukase et al. (2021) with machine learning methods that emphasize predictive performance.

3.4 Regression in a Machine Learning Framework

I have addressed logistic regression as an explanatory model in Chapter 2. However, its implementation differs when used as a predictive model in a ML pipeline. Traditional

logistic regression focuses on estimating model coefficients and testing their statistical significance. The primary goal is to understand the relationship between predictors and the outcome.

3.4.1 Logistic Regression in Machine Learning

In traditional statistics, logistic regression is solved analytically using MLE. All coefficients are estimated from the full dataset, and the performance is evaluated by in-sample fit measures, such as the log-likelihood. In contrast, in ML the model performance is assessed by splitting the data into multiple datasets, such as training data, test data and validation data. This is done using techniques such as train-test splits, k-fold cross-validation, or nested cross-validation. These different datasets are then used to train and test the model and to simulate how the model might behave when seeing new data in the future (Raschka, 2018).

This also changes the way the performance of logistic regression is evaluated. Instead of focusing on p-values or confidence intervals, the model is assessed using predictive metrics. These might include accuracy, AUC, and F1-score. These metrics are better aligned with decision making and classification performance (Christodoulou et al., 2019).

Furthermore, logistic regression is often regularized in ML pipelines. Regularization is achieved by using Lasso (L1) and Ridge (L2) penalties to avoid overfitting, especially when the number of predictors is large or when there is multicollinearity (Tibshiranit, 1996; Zou, 2005). This leads to a simpler model and a better generalization over a perfect fit to the training data.

3.4.2 Linear Regression and Continuous Outcomes

Linear regression is one of the most fundamental and widely used statistical models for examining the relationship between a continuous outcome and one or more predictors (Ernst & Albers, 2017). It provides a model for understanding how changes in the predictors relate to changes in the outcome.

In the simplest form, simple linear regression, the model consists of only one predictor

(Kutner et al., 2005):

$$Y = \beta_0 + \beta_1 * X_1 + \varepsilon \tag{3.1}$$

where Y is the outcome, X_1 is the predictor, β_0 is the intercept, β_1 is the slope or regression coefficient of the predictor X_1, and ε is the error term, which is assumed to be normally distributed.

In multiple linear regression, the model is extended to include multiple predictors (Kutner et al., 2005):

$$Y = \beta_0 + \beta_1 * X_1 + \cdots + \beta_n * X_n + \varepsilon \tag{3.2}$$

The coefficients $\beta_1, \ldots, \beta_n$ represent the marginal effect of each predictor on the outcome, controlled for the other predictors in the model. Marginal effects are typically estimated using Ordinary Least Squares (OLS), which minimizes the sum of squared residuals between observed and predicted values.

Linear regression is popular because of its simplicity and interpretability. It is commonly used in psychology to test hypotheses about relationships between variables and to make predictions about outcomes (Ernst & Albers, 2017; Kutner et al., 2005).

The classical linear regression model is based on the following assumptions (Kutner et al., 2005):

- Linearity: The relationship between the predictors and the outcome is linear
- Independence: The observations are independent of each other
- Homoscedasticity: The variance of errors is constant across all the levels of the predictors
- Normality of errors: The residuals are normally distributed

Violations of these assumptions may lead to biased estimates or invalidate inference. Diagnostics such as residual plots and VIF can be used to evaluate these conditions (Kutner et al., 2005).

Linear regression is very well suited for scenarios where the relationships between predictors and outcome are approximately linear and the focus is on inference or explanation. However, it may perform poorly when relationships are non-linear, there is multicollinearity between predictors, or the model is overfitted due to too many predictors relative to observations. It can be augmented with regularization techniques to improve generalization and address multicollinearity (Kutner et al., 2005).

Although logistic regression is suitable for binary classification, it can be suboptimal when the outcome variable is continuous (Lovasi et al., 2012). The PHQ-9 questionnaire used in the original study by Fukase et al. (2021) produces a total score that represents the severity of depression. Fukase et al. (2021) dichotomized this outcome with the threshold PHQ-9 ≥ 10, categorizing the participants as probably depressed or probably not depressed. Although this approach is reasonable and was validated in previous studies (Fukase et al., 2021), it results in loss of information by compressing the continuous PHQ-9 score into two categories (Altman & Royston, 2006).

Altman and Royston (2006) highlight three problems of dichotomisation:

1. The loss of information can lead to reduced statistical power and may also increase the risk of false positives

2. There may be a lot of variation between the groups and inside of the groups

3. Non-linearity in the relationship between predictor and outcome gets concealed

In contrast, linear regression takes advantage of the full range of depression scores. It allows the model to distinguish between mild, moderate, and severe depression symptoms using the full range of the continuous PHQ-9 score. This can result in more nuanced predictions, especially for a mental disorder such as depression, where the severity of symptoms exists on a spectrum where categories are not clearly distinguishable.

3.4.3 Linear Regression in Machine Learning

In its traditional form, linear regression is estimated using OLS, and inference is typically based on statistical tests of the coefficients, such as t-tests (Kutner et al., 2005). However, in a machine learning pipeline, linear regression is often extended with regularization techniques such as L1 and L2 penalties or a combination of both in elastic net regularization (Tibshiranit, 1996; Zou, 2005). The goal of these methods is to prevent overfitting by shrinking coefficients, and thus improving the ability of the model to generalize to new data. These regularization parameters cannot be derived analytically. Instead, they must be selected through hyperparameter optimization, since their optimal values depend on the data and the balance between bias and variance.

3.5 Gradient Boosted Decision Trees

3.5.1 Decision Trees

The following section is based on the foundational book for Classification And Regression Trees (CART) by Breiman et al. (2017).

A decision tree is a predictive modeling technique that is frequently used in ML for both classification and regression problems. Decision trees partition the feature space into a set of disjoint regions. This is done using decision rules learned directly from the data. The rules are applied sequentially and form a hierarchical, tree-like structure. A very simple example decision tree is provided in Fig. 3.2. A decision tree consists of the following components:

- Internal node: Represents a decision based on the value of a specific feature, effectively splitting the data into subsets. The internal nodes are represented as ovals in the example decision tree.

- Branch: Stems from a node and represents an outcome of this decision rule, guiding the path to subsequent nodes. Branches are represented as labeled arrows in the example decision tree.

- Leaf node: Represents a final decision or prediction, typically determined by av-

eraging (regression) or voting (classification) over data points that fall within that node. The leaf nodes are represented as rectangles in the example decision tree.

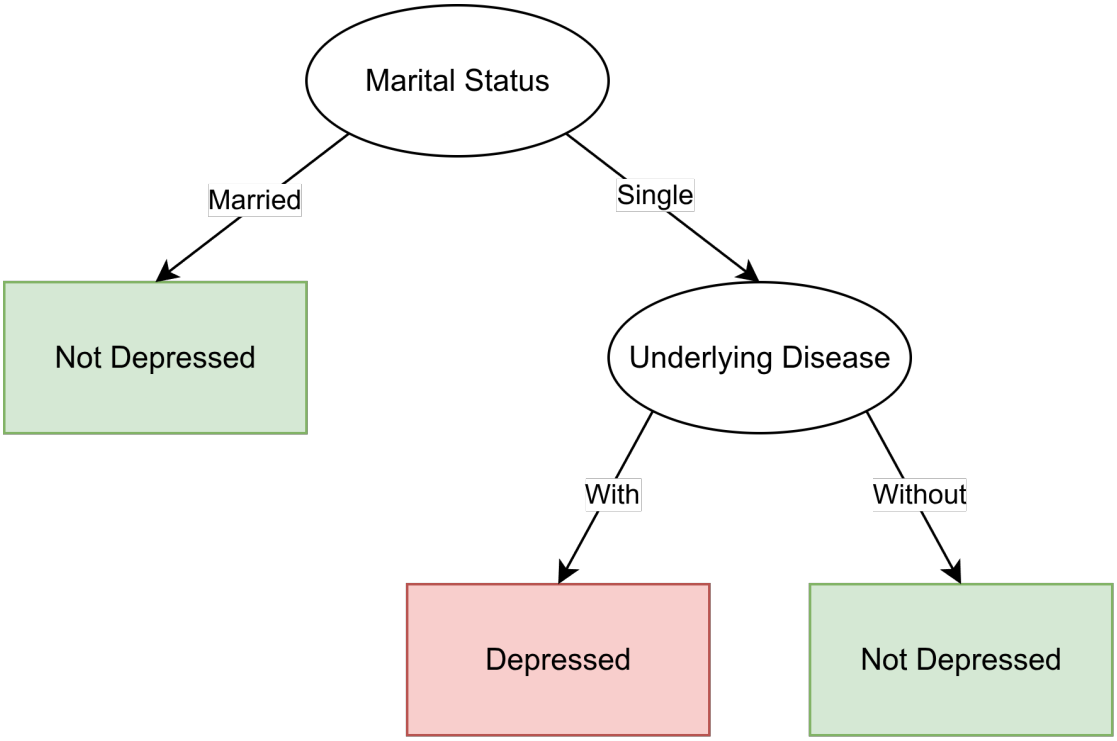

Figure 3.2: *Example decision tree.*

Decision trees are constructed through a recursive partitioning algorithm, often called recursive binary splitting. The algorithm repeatedly divides the data into subsets, striving to improve the accuracy of the predictions at each step. In a high-level overview, the construction of a decision tree works as follows:

1. Start with the given data points at the single node. In the first iteration, start with all data points at the root node.

2. Evaluate splits: For each predictor, consider potential splits. For categorical predictors, the splits are based on categories. For continuous predictors, midpoints between unique values or other split points can be chosen. Each potential split is evaluated using a statistical measure, for example, Gini impurity or entropy for classification and variance reduction for regression.

3. Select the best split: Choose the split that most effectively separates the data into subsets that are more homogenous or predictive of the outcome.

4. Recursive splitting: Check if a stopping criterion is reached. This might be the maximum tree depth or the minimum number of samples per node. There might also be no significant improvements from further splits. If no stopping criterion is reached, repeat the splitting process on each subset created by the previous split in Step 3. For each new node, start with its assigned data points in Step 1.

5. Prediction assignment: At each leaf node, assign a prediction. For classification, predict the majority class of data points within the leaf node. For regression, predict the mean value of the target variable for data points within the leaf node.

Decision trees have several advantages, such as interpretability and ease of visualization. However, individual decision trees tend to suffer from overfitting, especially when they are deep or complex. To address these limitations, ensemble methods, such as gradient boosting, combine multiple trees to improve predictive accuracy and generalization performance.

3.5.2 LightGBM

This section is based on the foundational papers on Gradient Boosting by Friedman (2001) and LightGBM by Ke et al. (2017).

Gradient Boosting is an ensemble learning technique that constructs a strong predictive model by combining the outputs of many weak learners, typically shallow decision trees, in a sequential manner. Gradient boosting trains each new tree to correct the errors made by the ensemble of previously trained trees. In a high-level overview, the construction of GBDT works as follows:

1. Compute the residual prediction errors of the current model.

2. Fit a new decision tree to the residual prediction errors.

3. Add the new tree to the model in a way that minimizes the residual prediction errors.

4. Continue steps 1 to 3 until a stopping criterion is reached.

This process continues iteratively, gradually improving performance by focusing on the instances the model currently gets wrong. Gradient boosting is particularly effective for structured/tabular data and supports a wide variety of loss functions, including log-loss for classification and MSE for regression.

GBDT are a specific application of gradient boosting where each weak learner is a decision tree, usually shallow (depth around 3 – 8). Each tree is trained to minimize a differentiable loss function using gradient descent, and the overall model is a weighted sum of all trees. GBDT have several advantages:

- They can capture complex, non-linear relationships
- They are robust to different types of features and scales
- They support both classification and regression tasks

However, GBDT require careful tuning of hyperparameters such as the number of trees, learning rate, and tree depth to avoid overfitting or underfitting.

LightGBM is an open source, high-performance implementation of GBDT developed by Microsoft. It introduces several innovations to make gradient boosting faster and more scalable:

- Histogram-based decision tree learning: Continuous features are bucketed into discrete bins to reduce memory usage and speed up computation.
- Leaf-wise growth strategy: Instead of growing trees level by level, LightGBM grows them leaf-wise with depth constraints, which often results in better accuracy.
- Efficient handling of large datasets: LightGBM is optimized for large-scale learning with support for parallel and GPU training.

LightGBM is widely used in applied ML and has become a default choice in many Kaggle competitions and industry applications due to its excellent balance of speed,

accuracy, and flexibility.

Chapter 4

Results

4.1 Replication

4.1.1 Descriptive Analysis (Additional Table 1)

The detailed results of the descriptive analysis can be found in Appendix A. To increase readability, I divided the original additional table 1 into three tables: Categorical variables in Table A.1, continuous variables in Table A.2, and age groups in Table A.3. Additional Table 1 of the original analysis by Fukase et al. (2021) was fully replicated, with the exception of two minor differences.

The two differences are highlighted in red in the corresponding tables. The first difference is the percentage of all participants from Ibaraki, which I calculated as 2.44% (2.5% in the original analysis). The second is the χ^2 statistic for age groups, which I calculated as 146.64 (146.65 in the original analysis). Since these deviations are minor, I assume that they are simple rounding errors. While the original analysis used SPSS Statistics 22 (IBM Corp., 2013), I used the Python programming language (Van Rossum, Drake et al., 1995) with relevant packages as described in Appendix A to perform my analysis. Thus, I find it highly likely that there might have been a small difference in the data processing methods or rounding conventions resulting from the use of the different tools.

4.1.2 Logistic Regression (Additional Table 2)

To examine the structure of the predictors, I first computed the predictor correlation matrix shown in Fig. 4.1. I used this correlation matrix as a general reference context for the rest of the analyses.

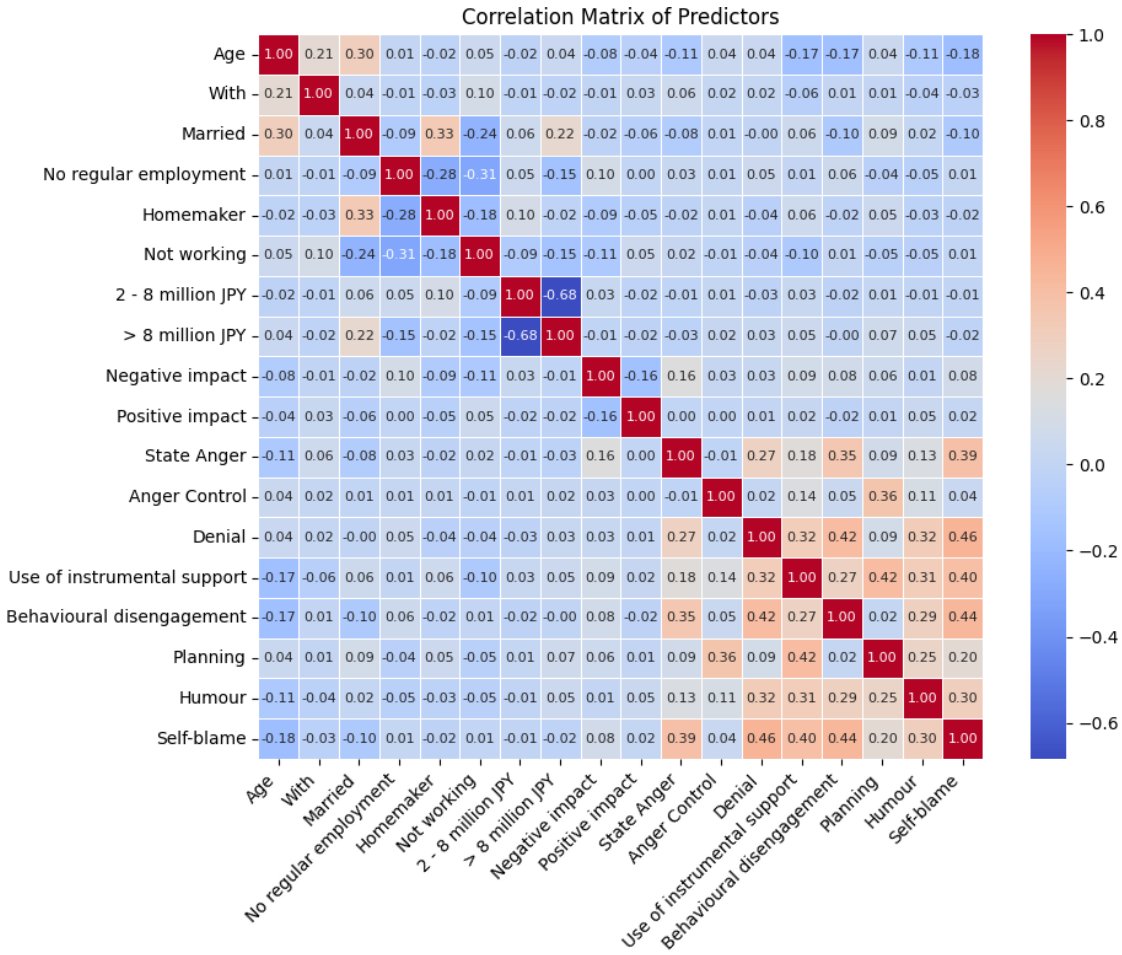

Figure 4.1: Predictor Correlation Matrix

The test results for the preconditions of the logistic regression are summarized in Table 4.1. VIF values were low across predictors (all < 2), indicating that there was no problematic collinearity. The Box–Tidwell interaction coefficients suggest that most continuous predictors met the linearity-in-the-logit assumption. However, significant deviations were observed for Humour (FDR-adjusted $p = 0.0006$), Instrumental support (FDR-adjusted $p = 0.0219$), and Planning (FDR-adjusted $p = 0.0137$). Cook's distance values were

very small (maximum $D = 0.016615$), suggesting that no individual case exerted undue influence on the logistic regression model.

Table 4.1
Logistic Regression Preconditions

Predictor	VIF	β (Box–Tidwell)	p-value	p-value (FDR)
Negative impact	1.10			
Positive impact	1.04			
Homemaker	1.37			
Not working	1.48			
No regular employment	1.42			
2 – 8 million JPY	2.26			
$>$ 8 million JPY	2.54			
Married	1.50			
With	1.06			
Age	1.32	−0.0437	0.1289	0.2900
State anger	1.28	−0.0190	0.7798	0.7798
Anger control	1.17	0.1224	0.2255	0.3382
Behavioural disengagement	1.50	0.3205	0.2206	0.3382
Denial	1.51	0.2362	0.3751	0.4822
Humour	1.28	0.9510	0.0001***	0.0006***
Instrumental support	1.53	0.6174	0.0073**	0.0219*
Planning	1.48	0.7505	0.0031**	0.0137*
Self blame	1.67	0.1675	0.4604	0.5180

Note. VIF = Variance Inflation Factor; β = Box–Tidwell interaction coefficient; FDR = False Discovery Rate (Benjamini–Hochberg corrected p-value). $^{*}p < .05$. $^{**}p < .01$. $^{***}p < .001$.

The complete replication of the logistic regression is shown in Table 4.2. In addition to the statistics reported by Fukase et al. (2021), I also report the p-value and VIF. Additional Table 2 of the original analysis was fully replicated, with the exception of one minor difference.

The difference is highlighted in red in the corresponding table. The Confidence Interval (CI) of the Humor coping style was marked by a significance star in the original analysis, indicating a p-value of less than 0.05. However, in my analysis, I found a p-value of 0.0501. This might be another rounding error as described in the previous Section 4.1.1, because the value is very close to the cutoff point of 0.05. Apart from this inaccuracy, all other statistics could be replicated.

Table 4.2
Logistic Regression Results

Predictor	β	SE	Wald	OR	95% CI		p-value	VIF
Intercept	−4.14	0.61	46.34	0.02	0.0	0.05	0.0000***	62.15
Negative impact	0.29	0.14	4.01	1.33	1.01	1.77	0.0452*	1.10
Positive impact	−0.53	0.45	1.41	0.59	0.25	1.41	0.2349	1.04
Homemaker	0.20	0.26	0.58	1.22	0.74	2.01	0.4450	1.37
Not working	0.62	0.21	8.49	1.85	1.22	2.8	0.0036**	1.48
No regular employment	0.04	0.18	0.04	1.04	0.73	1.47	0.8343	1.42
2 - 8 million JPY	−0.13	0.19	0.48	0.87	0.6	1.28	0.4877	2.26
> 8 million JPY	−0.80	0.29	7.46	0.45	0.25	0.8	0.0063**	2.54
Married	−0.64	0.17	14.10	0.53	0.38	0.74	0.0002***	1.50
With	0.67	0.20	11.07	1.96	1.32	2.92	0.0009***	1.06
Age	−0.03	0.00	38.61	0.97	0.96	0.98	0.0000***	1.32
State Anger	0.16	0.01	137.54	1.17	1.14	1.21	0.0000***	1.28
Anger Control	0.08	0.02	13.17	1.08	1.04	1.13	0.0003***	1.17
Behavioural Disengagement	0.25	0.06	15.90	1.28	1.13	1.44	0.0001***	1.50
Denial	−0.13	0.07	4.19	0.88	0.77	0.99	0.0408*	1.51
Humour	−0.11	0.06	3.84	0.89	0.8	1.0	0.0501	1.28
Instrumental Support	−0.17	0.06	8.41	0.85	0.76	0.95	0.0037**	1.53
Planning	−0.18	0.06	8.62	0.84	0.74	0.94	0.0033**	1.48
Self Blame	0.38	0.06	41.76	1.47	1.31	1.65	0.0000***	1.67

Note. β = unstandardized coefficient; SE = standard error; Wald = Wald statistic; OR = Odds Ratio; CI = confidence interval; VIF = Variance Inflation Factor. $^{*}p < .05$. $^{**}p < .01$. $^{***}p < .001$.

4.2 Simple Linear Regression

The results of the simple linear regression analysis are shown in Table 4.3. Simple linear regression analysis and simple logistic regression analysis show consistent patterns. All predictors show the same directional relationships. The simple linear model and the simple logistic model both achieved strong discriminative performance (R^2 = 0.42 and RMSE = 4.12 linear, AUC = 0.85 and balanced accuracy = 69% logistic). Key demographic factors also demonstrated consistent effects: Being married was a strong protective factor (β = -1.17 linear, β = -0.64 logistic), while having an underlying disease (”With”) substantially increased the risk of depression (β = 1.13 linear β = 0.67 logistic). Among the employment categories, ”Not working” emerged as the strongest risk factor in both models (β = 1.28 linear, β = 0.62 logistic), while other employment statuses showed more modest effects. Psychological factors were the most robust pre-

dictors in both analyses based on the Wald statistic. State anger showed the highest coefficients (β = 0.37 linear, β = 0.16 logistic), while Self-blame also showed strong positive associations (β = 0.85 linear, β = 0.38 logistic). Protective coping strategies including Denial, Humour, Instrumental support, and Planning consistently showed negative relationships with depression in both models. Statistical significance patterns were largely consistent.

Table 4.3
Linear Regression Results

Predictor	β	SE	Wald	95% CI		p-value
Constant	5.30	0.08	4437.78	5.14	5.46***	0.0000
Negative impact	0.44	0.08	27.78	0.28	0.60***	0.0000
Positive impact	−0.02	0.08	0.05	−0.18	0.14	0.8294
Homemaker	0.17	0.09	3.09	−0.02	0.35	0.0789
Not working	0.49	0.10	26.49	0.30	0.68***	0.0000
No regular employment	0.22	0.09	5.54	0.04	0.41*	0.0187
2 - 8 million JPY	−0.10	0.09	1.14	−0.28	0.08	0.2866
> 8 million JPY	−0.13	0.10	1.88	−0.32	0.06	0.1706
Married	−0.58	0.10	35.89	−0.77	−0.39***	0.0000
With	0.36	0.08	19.59	0.20	0.52***	0.0000
Age	−0.99	0.09	115.34	−1.17	−0.81***	0.0000
State Anger	2.01	0.09	490.41	1.83	2.19***	0.0000
Anger Control	0.23	0.09	7.04	0.06	0.40**	0.0080
Behavioural disengagement	0.73	0.10	57.97	0.55	0.92***	0.0000
Denial	−0.55	0.10	32.33	−0.74	−0.36***	0.0000
Humour	−0.34	0.09	14.53	−0.52	−0.17***	0.0001
Instrumental Support	−0.35	0.10	12.89	−0.54	−0.16***	0.0003
Planning	−0.35	0.10	13.34	−0.55	−0.16***	0.0003
Self blame	1.19	0.10	136.11	0.99	1.40***	0.0000

Note. β = standardized coefficient; SE = standard error; Wald = Wald statistic; CI = confidence interval. $^{*}p < .05$. $^{**}p < .01$. $^{***}p < .001$.

4.3 Machine Learning Model Performance

To evaluate the predictive performance of the models, I performed nested CV with 10 outer folds and Optuna HPO in the inner loop for both classification and regression tasks.

Table 4.4 shows a summary of the classification results. The LightGBM classifier achieved a slightly higher mean balanced accuracy of $\sim$ 68% compared to $\sim$ 67% of

the logistic regression. The LightGBM classifier also demonstrated a higher mean AUC compared to logistic regression (~ 0.87 vs. ~ 0.84). This suggests that while the overall accuracy was similar, the LightGBM classifier provided better discriminative ability between the outcome classes.

Table 4.4
Nested Cross-Validation Performance (Classification)

Algorithm	Accuracy (SD)	AUC (SD)
Logistic Regression	0.6732 (0.0362)	0.8406 (0.0348)
LightGBM Classifier	0.6830 (0.0355)	0.8668 (0.0274)

Note. Mean $\pm$ SD computed across outer folds of nested CV using the best hyper-parameters per fold.

Table 4.5 shows a summary of the regression results. The Elastic Net model achieved a mean R^2 of ~ 0.41, a mean RMSE of ~ 4.14, and a mean Mean Absolute Error (MAE) of ~ 3.10. The LightGBM regressor achieved a mean R^2 of ~ 0.44, a mean RMSE of ~ 4.02, and a mean MAE of ~ 2.98. These results suggest that, while both models explained a moderate proportion of variance ($\sim 40\%$), the LightGBM regressor showed modest improvements over the Elastic Net regressor in all three metrics.

Table 4.5
Nested Cross-Validation Performance (Regression)

Algorithm	R^2 (SD)	RMSE (SD)	MAE (SD)
Elastic Net	0.4071 (0.0421)	4.1409 (0.2598)	3.0994 (0.1586)
LightGBM Regressor	0.4394 (0.0468)	4.0236 (0.2412)	2.9753 (0.1348)

Note. Mean $\pm$ SD computed across outer folds of nested CV using the best hyper-parameters per fold.

Chapter 5

Discussion

5.1 Replication results

The results of the replication were consistent with the findings of the original study (Fukase et al., 2021). Besides some minor issues addressed in the results section, which I would attribute to rounding errors or small differences in the analysis tools, all statistics could be fully replicated.

If my replication of the logistic regression analysis is correct, many participants were excluded from the analysis. As I described in Chapter 2, I suspect that all participants who reported ”I do not know” or ”Non-response” for Household Income were excluded from the regression analysis. These two groups account for 21.64% (586 participants) of the total data. In addition, eight coping strategies from the Brief COPE inventory (Acceptance, Active coping, Emotional support, Positive reframing, Religion, Self-distraction, Substance use and Venting) appear to have been omitted from the regression models without explanation. Since these exclusions are not documented in Fukase et al. (2021), the rationale remains unclear.

The exclusion of around one-fifth of the available data is quite severe. It may have reduced the statistical power and potentially biased the regression coefficients. The values ”I do not know” and ”Non-response” in Household income may be systematically

related to socioeconomic status or other observed or unobserved factors. A more transparent communication and handling of missing data, for example imputation, would have been preferable. This would allow the analysis to retain more data and provide a clearer rationale for the modeling decisions.

Another point of consideration is that the original study relied exclusively on traditional logistic regression. Although simple logistic and linear regression are the standard in the field and provide more easily interpretable results (Ernst & Albers, 2017), my machine learning analysis demonstrates that alternative approaches such as LightGBM may achieve equal or better predictive performance. LightGBM overall demonstrated slightly higher predictive performance, particularly in terms of AUC in classification (about 0.025 higher than logistic regression). These gains were modest and did not fundamentally change the substantive conclusions.

Traditional regression approaches remain valuable in psychological research due to their interpretability and well-established methodological foundations. Although modern machine learning models can capture complex patterns in data, understanding the predictions of these models can be challenging. Recent advances in explainable AI have begun to address these concerns. Tools such as SHapley Additive exPlanations (SHAP) (Lundberg & Lee, 2017) can be used to quantify the contribution of individual features to the prediction of the model. This can offer insights comparable to traditional regression coefficients on complex non-linear models. These techniques allow researchers to understand why a model makes a specific prediction. Rather than viewing traditional statistical approaches and modern machine learning as competing paradigms, these methods can be seen as complementary tools. Classical regression models can benefit from machine learning infrastructure, including nested CV and automated HPO, to reduce overfitting and provide more robust estimates of generalization performance.

5.2 Limitations

Although I was able to reproduce the findings of Fukase et al. (2021), I have to acknowledge several limitations in my own analysis.

First, I implemented my analysis using different software tools than those reported in

the original study. As I described in Chapter 2, this might have led to some minor differences in the results. Even when the same statistical models are applied, small variations in the software can lead to differences in the results due to different defaults, optimization procedures, or rounding behavior.

Second, the handling of missing income responses remains ambiguous. Because the original analysis procedure was not well documented, I assumed that the authors had excluded participants with missing or non-response values, since this approach was the only way I could reproduce the reported regression results. Although this exclusion appears plausible, it cannot be confirmed with certainty. A more rigorous approach would have been to use multiple imputation or sensitivity analyses to examine whether the results are valid under different assumptions about missing data. As the ambiguity could not be resolved from the data or the paper alone, I proceeded with this assumption to complete the analysis.

Third, my analysis is constrained by the fact that I did not collect data myself. My conclusions are limited by the data made available by Fukase (2020). Without access to the original survey instruments and a complete record of preprocessing decisions, I cannot verify all aspects regarding data handling. Future replications would benefit from greater transparency and the publication of analysis code along with the datasets, which would facilitate more complete reproducibility.

Beyond these methodological points, there are also several broader limitations. My replication concentrated on reproducing the reported statistical analyses, but I did not re-evaluate other aspects of the original study, such as survey design, recruitment, or measurement validity. Thus, my replication is concerned with the reproducibility of the reported results rather than with the study as a whole.

As described in Chapter 1, the distribution of probable depression and not probable depression in the dataset was skewed. 18.35% of participants were classified in the probable depression group, compared to only 2.0% of depression prevalence reported by Demiya et al. (2022). This high rate might indicate sampling bias or measurement sensitivity. The sample was also unbalanced, with more than four times as many participants in the not probable depression group as in the probable depression group (82.75% vs

18.35%). This imbalance might affect the stability of statistical estimates and limit the generalizability of the findings.

The machine learning results come with certain methodological considerations. Procedures such as nested CV and HPO are not entirely deterministic. If random states or search spaces change, these methods might yield different outcomes. However, when combined with explainable AI techniques such as SHAP (Lundberg & Lee, 2017), machine learning models can provide both predictive accuracy and interpretability of individual predictors. This combination allows machine learning approaches to address similar research questions as traditional regression models while potentially capturing more complex patterns in the data. Rather than being limited to prediction tasks, modern machine learning paired with explainability tools can serve as a full methodological alternative to traditional statistical approaches.

Finally, both traditional regression and machine learning models only explained a moderate proportion of the variance in depressive symptoms (about 40%). This indicates that some important influential factors are missing in the dataset and that additional predictors are needed to explain more of the variance in the outcome.

These limitations do not undermine my replication, but highlight the importance of transparent reporting, careful handling of missing data, and continued methodological development. They also yield promising directions for future research that combine the interpretability of traditional statistical methods and the predictive advantages of modern machine learning approaches.

Chapter 6

Conclusion

The purpose of this thesis was to replicate the analysis reported by Fukase et al. (2021) on depression during the COVID-19 pandemic in Japan and to extend their work with additional machine learning analyses. My replication confirmed the robustness of the original results. All key statistics were successfully reproduced, with only minor discrepancies attributable to rounding errors or software differences. This strengthens confidence in the validity of the original findings and demonstrates that the reported results are reproducible when the same data and statistical methods are applied.

At the same time, my replication highlighted several issues of transparency and methodological choice. Most notably, it appears that more than one-fifth of the sample was excluded from the logistic regression analysis due to missing Household income data, a decision not explicitly documented in the original paper. Such exclusions reduce statistical power and may bias parameter estimates, underscoring the importance of clearly reporting how missing data are handled. In addition, reliance on traditional regression models without further discussion of predictive performance or alternative modeling strategies reflects common practice in empirical psychological research but also leaves room for methodological improvements.

The additional machine learning analysis illustrates how modern approaches can complement traditional methods. LightGBM models achieved modestly higher predictive

performance than the logistic and linear regression. However, both traditional regression and machine learning models explained only a moderate proportion of variance in depressive symptoms (around 40%), indicating that important predictors might be missing from the dataset. Importantly, these analyses also demonstrate that it can be beneficial to apply machine learning infrastructure, such as nested CV and HPO, to classical models such as logistic and linear regression. Doing so can improve robustness and provide more reliable estimates of generalization performance while preserving interpretability.

Overall, this thesis contributes both to the reliability of psychological research and to its methodological development. By confirming the reproducibility of Fukase et al. (2021), it strengthens confidence in their findings, while also drawing attention to the need for more transparent reporting of data exclusions and preprocessing decisions. At the same time, it shows that integrating modern machine learning practices into traditional analysis frameworks can improve the robustness of results without sacrificing interpretability. Future research should therefore combine the strengths of traditional statistical methods with modern machine learning techniques, while also ensuring transparent reporting and comprehensive handling of missing data.

Appendix A

Appendix

Additional Tables

Table A.1
Replication of Additional Table 1: Categorical Variables

Demographic	All, n (%)	Not depressed, n (%)	Depressed, n (%)	χ^2
Male	1354 (50.00)	1073 (48.53)	281 (56.54)	
Female	1354 (50.00)	1138 (51.47)	216 (43.46)	
Employment status				55.79***
Full-time worker	956 (35.3)	821 (37.13)	135 (27.16)	
No regular employment	876 (32.35)	705 (31.89)	171 (34.41)	
Homemaker	392 (14.48)	341 (15.42)	51 (10.26)	
Not working	484 (17.87)	344 (15.56)	140 (28.17)	
Residential areas				7.29
Hokkaido	193 (7.13)	150 (6.78)	43 (8.65)	
Ibaraki	66 (2.44)	53 (2.40)	13 (2.62)	
Saitama	277 (10.23)	234 (10.58)	43 (8.65)	
Chiba	220 (8.12)	176 (7.96)	44 (8.85)	
Tokyo	507 (18.72)	417 (18.86)	90 (18.11)	
Kanagawa	325 (12.00)	272 (12.30)	53 (10.66)	
Ishikawa	42 (1.55)	33 (1.49)	9 (1.81)	
Gifu	58 (2.14)	47 (2.13)	11 (2.21)	
Aichi	267 (9.86)	218 (9.86)	49 (9.86)	
Kyoto	91 (3.36)	72 (3.26)	19 (3.82)	
Osaka	328 (12.11)	270 (12.21)	58 (11.67)	
Hyogo	205 (7.57)	169 (7.64)	36 (7.24)	
Fukuoka	129 (4.76)	100 (4.52)	29 (5.84)	
Underlying disease				4.95*
Without	2389 (88.22)	1965 (88.87)	424 (85.31)	
With	319 (11.78)	246 (11.13)	73 (14.69)	
Marital status				129.06***
Single	1216 (44.90)	879 (39.76)	337 (67.81)	
Married	1492 (55.10)	1332 (60.24)	160 (32.19)	
Household income				44.84***
$<$ 2 million JPY	299 (11.04)	214 (9.68)	85 (17.10)	
2 - 8 million JPY	1434 (52.95)	1182 (53.46)	252 (50.70)	
$>$ 8 million JPY	389 (14.36)	352 (15.92)	37 (7.44)	
I do not know	297 (10.97)	230 (10.40)	67 (13.48)	
Non-response	289 (10.67)	233 (10.54)	56 (11.27)	
Economically impacted				29.68***
Without impact	1471 (54.32)	1254 (56.72)	217 (43.66)	
Negative impact	1160 (42.84)	893 (40.39)	267 (53.72)	
Positive impact	77 (2.84)	64 (2.89)	13 (2.62)	

Note. Absolute frequencies of categorical variables with relative frequencies in parentheses. χ^2 values are reported for each categorical variable. $^{*}p < .05$. $^{**}p < .01$. $^{***}p < .001$.

Table A.2
Replication of Additional Table 1: Categorical Variables

Demographic	All, M (SD)	Not depressed, M (SD)	Depressed, M (SD)	t
Age	49.16 (16.32)	50.91 (16.21)	41.41 (14.48)	12.91***
PHQ-9	5.30 (5.41)	3.19 (2.76)	14.66 (4.22)	57.83***
State anger	15.18 (5.38)	14.16 (4.37)	19.71 (6.89)	17.21***
Anger control	19.10 (4.03)	19.06 (4.06)	19.27 (3.91)	1.05
Acceptance	5.97 (1.32)	5.99 (1.31)	5.91 (1.36)	1.14
Active coping	5.26 (1.30)	5.30 (1.29)	5.06 (1.35)	3.74***
Behavioural disengagement	3.88 (1.31)	3.72 (1.22)	4.60 (1.46)	12.58***
Denial	3.15 (1.24)	3.10 (1.18)	3.36 (1.45)	3.74***
Emotional support	4.07 (1.45)	4.03 (1.41)	4.25 (1.60)	2.74**
Humour	3.68 (1.36)	3.67 (1.31)	3.73 (1.55)	0.77
Instrumental support	4.06 (1.48)	4.04 (1.44)	4.17 (1.65)	1.60
Planning	5.07 (1.39)	5.10 (1.37)	4.92 (1.49)	2.42*
Positive reframing	4.75 (1.42)	4.76 (1.40)	4.68 (1.51)	1.02
Religion	3.25 (1.38)	3.16 (1.31)	3.64 (1.59)	6.26***
Self blame	3.52 (1.40)	3.31 (1.25)	4.43 (1.66)	14.19***
Self distraction	4.70 (1.37)	4.63 (1.34)	4.99 (1.43)	5.25***
Substance use	3.20 (1.54)	3.08 (1.44)	3.72 (1.85)	7.21***
Venting	4.06 (1.35)	3.99 (1.29)	4.37 (1.54)	5.21***

Note. Mean of continuous variables with standard deviation in parentheses. t-values are reported for each continuous variable. $^{*}p < .05$. $^{**}p < .01$. $^{***}p < .001$.

Table A.3
Replication of Additional Table 1: Age groups

Demographic	All, n (%)	Not depressed, n (%)	Depressed, n (%)	χ^2
Age group				146.64***
20s	454 (16.77)	311 (14.07)	143 (28.77)	
30s	454 (16.77)	343 (15.51)	111 (22.33)	
40s	454 (16.77)	358 (16.19)	96 (19.32)	
50s	454 (16.77)	367 (16.60)	87 (17.51)	
60s	454 (16.77)	423 (19.13)	31 (6.24)	
70s	438 (16.17)	409 (18.50)	29 (5.84)	

Note. Absolute frequencies of age groups with relative frequencies in parentheses. χ^2 values are reported for each age group. $^{*}p < .05$. $^{**}p < .01$. $^{***}p < .001$.

Software Environment

The Python environment used in the analysis is documented in the following. For clarity, the main packages are presented in Table A.4, while the complete requirements.txt is also provided for reproducibility in Listing A.1.

Table A.4
Core Python packages used in the analysis

Package	Version
pandas	2.3.2
numpy	2.3.2
statsmodels	0.14.5
scipy	1.16.1
scikit-learn	1.7.1
lightgbm	4.6.0
optuna	4.5.0
matplotlib	3.10.5
seaborn	0.13.2
openpyxl	3.1.5

Note. Versions correspond to the Python 3.12.3 environment used in this thesis.

The complete environment specification in the form of the `requirements.txt` is given below:

Listing A.1: requirements.txt

```
alembic==1.16.4
colorlog==6.9.0
contourpy==1.3.3
cycler==0.12.1
et_xmlfile==2.0.0
fonttools==4.59.1
greenlet==3.2.4
Jinja2==3.1.6
joblib==1.5.1
kiwisolver==1.4.9
lightgbm==4.6.0
Mako==1.3.10
MarkupSafe==3.0.2
matplotlib==3.10.5
numpy==2.3.2
```

```
openpyxl==3.1.5
optuna==4.5.0
optuna-integration==4.5.0
packaging==25.0
pandas==2.3.2
patsy==1.0.1
pillow==11.3.0
pyparsing==3.2.3
python-dateutil==2.9.0.post0
pytz==2025.2
PyYAML==6.0.2
scikit-learn==1.7.1
scipy==1.16.1
seaborn==0.13.2
six==1.17.0
SQLAlchemy==2.0.43
statsmodels==0.14.5
threadpoolctl==3.6.0
tqdm==4.67.1
typing_extensions==4.15.0
tzdata==2025.2
```

Bibliography

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/3292500.3330701

Altman, D. G., & Royston, P. (2006). The cost of dichotomising continuous variables. *BMJ (Clinical research ed.)*, *332*(7549), 1080. https://doi.org/10.1136/bmj.332.7549.1080

Box, G. E., & Tidwell, P. W. (1962). Transformation of the independent variables. *Technometrics : a journal of statistics for the physical, chemical, and engineering sciences*, *4*(4), 531–550. https://doi.org/10.1080/00401706.1962.10490038

Bradshaw, T. J., Huemann, Z., Hu, J., & Rahmim, A. (2023). A guide to cross-validation for artificial intelligence in medical imaging. *Radiology: Artificial Intelligence*, *5*(4), e220232. https://doi.org/10.1148/ryai.220232

Breiman, L. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, *16*(3), 199–231. https://doi.org/10.1214/ss/1009213726

Breiman, L., Friedman, J., Olshen, R. A., & Stone, C. J. (2017). *Classification and regression trees*. Chapman and Hall/CRC.

Carver, C. S. (1997). You want to measure coping but your protocol'too long: Consider the brief cope. *International journal of behavioral medicine*, *4*(1), 92–100. https://doi.org/10.1207/s15327558ijbm0401_6

Chand, S. P., Arif, H., & Kutlenios, R. M. (2023). Depression (nursing). In *StatPearls [internet]*. StatPearls Publishing.

Cho, Y., Mishiro, I., Fujimoto, S., & Nakajima, T. (2021). Impact of depression onset and treatment on the trend of annual medical costs in japan: An exploratory, descriptive analysis of employer-based health insurance claims data. *Advances in Therapy*, 1–14. https://doi.org/10.1007/s12325-021-01963-9

Choi, E. P. H., Hui, B. P. H., & Wan, E. Y. F. (2020). Depression and anxiety in hong kong during COVID-19. *International journal of environmental research and public health*, *17*(10), 3740. https://doi.org/10.3390/ijerph17103740

Christodoulou, E., Ma, J., Collins, G. S., Steyerberg, E. W., Verbakel, J. Y., & Van Calster, B. (2019). A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. *Journal of clinical epidemiology*, *110*, 12–22. https://doi.org/10.1016/j.jclinepi.2019.02.004

Collaboration, O. S. (2015). Estimating the reproducibility of psychological science. *Science*, *349*(6251), aac4716. https://doi.org/10.1126/science.aac4716

Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *20*(2), 215–232. https://doi.org/10.1111/j.2517-6161.1958.tb00292.x

Cuijpers, P., Silverstein, M., & Gladstone, T. (2025). Preventing depression: Challenges and innovations. *Journal of consulting and clinical psychology*, *93*(4), 191. https://doi.org/10.1037/ccp0000951

Daly, M., & Robinson, E. (2022). Depression and anxiety during COVID-19. *The Lancet*, *399*(10324), 518. https://doi.org/10.1016/S0140-6736(22)00187-8

Demiya, S., Takeno, S., Kim, S. W., Lee, W. S., & Sakai, Y. (2022). EPH14 prevalence of depression in japan and the US populations before and during the COVID-19 pandemic: A retrospective observational study using real-world data. *Value in Health*, *25*(12), S193. https://doi.org/10.1016/j.jval.2022.09.936

Ernst, A. F., & Albers, C. J. (2017). Regression assumptions in clinical psychology research practice—a systematic review of common misconceptions. *PeerJ*, *5*, e3323. https://doi.org/10.7717/peerj.3323

Farahzadi, M. (2017). Depression; let's talk. *Journal of Community Health Research*, *6*(2), 74–76.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of statistics*, 1189–1232.

Fukase, Y. (2020). Depression & coping during COVID-19 second waves in Japan. *Inter-university Consortium for PoliticalSocial Research [distributor]*. https://doi.org/10.3886/E121361V1

Fukase, Y., Ichikura, K., Murase, H., & Tagaya, H. (2021). Depression, risk factors, and coping strategies in the context of social dislocations resulting from the second wave of COVID-19 in Japan. *BMC psychiatry*, *21*, 1–9. https://doi.org/10.1186/s12888-021-03047-y

Gazoni, E., & Clark, C. (2024). *Openpyxl* (Version 3.1.5). https://pypi.org/project/openpyxl/

Grant, S., Wendt, K. E., Leadbeater, B. J., Supplee, L. H., Mayo-Wilson, E., Gardner, F., & Bradshaw, C. P. (2022). Transparent, Open, and Reproducible Prevention Science. *Prevention Science*, *23*(5), 701–722. https://doi.org/10.1007/s11121-022-01336-w

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hawton, K., i Comabella, C. C., Haw, C., & Saunders, K. (2013). Risk factors for suicide in individuals with depression: A systematic review. *Journal of affective disorders*, *147*(1–3), 17–28. https://doi.org/10.1016/j.jad.2013.01.004

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, *21*(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

Hengartner, M. P. (2018). Raising awareness for the replication crisis in clinical psychology by focusing on inconsistencies in psychotherapy research: How much can we rely on published findings from efficacy trials? *Frontiers in Psychology*, *9*, 256. https://doi.org/10.3389/fpsyg.2018.00256

Hori, K., Yamada, Y., Namba, H., Kimura, M., Fujita, H., & Date, H. (2025). Economic deterioration and social factors affecting mental health during the COVID-19 pandemic in japan: Web-based cross-sectional survey. *JMIR Formative Research*, *9*, e65204. https://doi.org/10.2196/65204

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

IBM Corp. (2013). *IBM SPSS statistics for windows*. manual. Armonk, NY.

Ioannidis, J. P. (2005). Why most published research findings are false. *PLoS medicine*, *2*(8), e124. https://doi.org/10.1371/journal.pmed.1004085

Ishikawa, H., Kawakami, N., Kessler, R. C., Collaborators, W. M. H. J. S., et al. (2016). Lifetime and 12-month prevalence, severity and unmet need for treatment of common mental disorders in Japan: Results from the final dataset of World Mental Health Japan Survey. *Epidemiology and psychiatric sciences*, *25*(3), 217–229. https://doi.org/10.1017/S2045796015000566

Kawakami, N. (2007). Epidemiology of depressive disorders in Japan and the world. *Nihon rinsho. Japanese journal of clinical medicine*, *65*(9), 1578–1584.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, *30*.

Korbmacher, M., Azevedo, F., Pennington, C. R., Hartmann, H., Pownall, M., Schmidt, K., Elsherif, M., Breznau, N., Robertson, O., Kalandadze, T., et al. (2023). The replication crisis has led to positive structural, procedural, and community changes. *Communications Psychology*, *1*(1), 3. https://doi.org/10.1038/s44271-023-00003-2

Kroenke, K., Spitzer, R. L., & Williams, J. B. (2001). The PHQ-9: Validity of a brief depression severity measure. *Journal of general internal medicine*, *16*(9), 606–613. https://doi.org/10.1046/j.1525-1497.2001.016009606.x

Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2005). *Applied linear statistical models*. McGraw-hill.

Lépine, J.-P., & Briley, M. (2011). The increasing burden of depression. *Neuropsychiatric disease and treatment*, *7*, 3–7. https://doi.org/10.2147/NDT.S19617

Lovasi, G. S., Underhill, L. J., Jack, D., Richards, C., Weiss, C., & Rundle, A. (2012). At odds: Concerns raised by using odds ratios for continuous or common dichotomous outcomes in research on physical activity and obesity. *The open epidemiology journal*, *5*, 13. https://doi.org/10.2174/1874297101205010013

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S.

Vishwanathan & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

McKinney, W. (2010). Data Structures for Statistical Computing in Python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 56–61). https://doi.org/10.25080/Majora-92bf1922-00a

Nagib, N., Horita, R., Miwa, T., Adachi, M., Tajirika, S., Imamura, N., Ortiz, M. R., & Yamamoto, M. (2023). Impact of COVID-19 on the mental health of Japanese university students (years II-IV). *Psychiatry research*, *325*, 115244. https://doi.org/10.1016/j.psychres.2023.115244

Nosek, B. A., Hardwicke, T. E., Moshontz, H., Allard, A., Corker, K. S., Dreber, A., Fidler, F., Hilgard, J., Kline Struhl, M., Nuijten, M. B., Rohrer, J. M., Romero, F., Scheel, A. M., Scherer, L. D., Schönbrodt, F. D., & Vazire, S. (2022). Replicability, robustness, and reproducibility in psychological science. *Annual Review of Psychology*, *73*, 719–748. https://doi.org/10.1146/annurev-psych-020821-114157

O'brien, R. M. (2007). A caution regarding rules of thumb for variance inflation factors. *Quality & quantity*, *41*(5), 673–690. https://doi.org/10.1007/s11135-006-9018-6

Olaya, B., Perez-Moreno, M., Bueno-Notivol, J., Gracia-Garcia, P., Lasheras, I., & Santabarbara, J. (2021). Prevalence of depression among healthcare workers during the COVID-19 outbreak: A systematic review and meta-analysis. *Journal of clinical medicine*, *10*(15), 3406. https://doi.org/10.3390/jcm10153406

Otsuka, Y., Sasaki, T., Iwasaki, K., & Mori, I. (2009). Working hours, coping skills, and psychological health in Japanese daytime workers. *Industrial health*, *47*(1), 22–32. https://doi.org/10.2486/indhealth.47.22

Pearson, K. (1900). X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *50*(302), 157–175. https://doi.org/10.1080/14786440009463897

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau,

D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Racine, N., McArthur, B. A., Cooke, J. E., Eirich, R., Zhu, J., & Madigan, S. (2021). Global prevalence of depressive and anxiety symptoms in children and adolescents during COVID-19: A meta-analysis. *JAMA pediatrics*, *175*(11), 1142–1150. https://doi.org/10.1001/jamapediatrics.2021.2482

Raschka, S. (2018). *Model evaluation, model selection, and algorithm selection in machine learning*. https://doi.org/10.48550/arxiv.1811.12808

Robinson, E., Sutin, A. R., Daly, M., & Jones, A. (2022). A systematic review and meta-analysis of longitudinal cohort studies comparing mental health before versus during the COVID-19 pandemic in 2020. *Journal of affective disorders*, *296*, 567–576. https://doi.org/10.1016/j.jad.2021.09.098

Santomauro, D. F., Herrera, A. M. M., Shadid, J., Zheng, P., Ashbaugh, C., Pigott, D. M., Abbafati, C., Adolph, C., Amlag, J. O., Aravkin, A. Y., et al. (2021). Global prevalence and burden of depressive and anxiety disorders in 204 countries and territories in 2020 due to the COVID-19 pandemic. *The Lancet*, *398*(10312), 1700–1712. https://doi.org/10.1016/S0140-6736(21)02143-7

Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with python. *9th Python in Science Conference*. https://doi.org/10.25080/Majora-92bf1922-011

Shi, Y., Ke, G., Soukhavong, D., Lamb, J., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y., Titov, N., & Cortes, D. (2025). *Lightgbm: Light gradient boosting machine*. manual. https://github.com/Microsoft/LightGBM

Shmueli, G. (2010). To explain or to predict? *Statistical Science*, *25*(3), 289–310. https://doi.org/10.1214/10-STS330

Spielberger, C. D. (1988). Manual for the state-trait anger expression inventory. *Odessa, FL: Psychological Assessment Resources*.

Stoltzfus, J. C. (2011). Logistic regression: A brief primer. *Academic emergency medicine*, *18*(10), 1099–1104. https://doi.org/10.1111/j.1553-2712.2011.01185.x

Student. (1908). The probable error of a mean. *Biometrika*, 1–25. https://doi.org/10.2307/2331554

Suzuki, T., & Haruki, Y. (1994). The relationship between anger and circulatory disease. *Japanese Journal of Health Psychology*. https://doi.org/10.11560/jahp.7.1_1

Tackett, J. L., Brandes, C. M., King, K. M., & Markon, K. E. (2019). Psychology's replication crisis and clinical psychological science. *Annual review of clinical psychology*, *15*(1), 579–604. https://doi.org/10.1146/annurev-clinpsy-050718-095710

team, T. pandas development. (2025, June). *Pandas-dev/pandas: Pandas* (Version v2.3.0). https://doi.org/10.5281/zenodo.15597513

Tibshiranit, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, *58*(1), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Ueda, M., Nordström, R., & Matsubayashi, T. (2022). Suicide and mental health during the COVID-19 pandemic in Japan. *Journal of Public Health*, *44*(3), 541–548. https://doi.org/10.1093/pubmed/fdab113

Van Rossum, G., Drake, F. L., et al. (1995). *Python reference manual* (Vol. 111). Centrum voor Wiskunde en Informatica Amsterdam.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Waskom, M. L. (2021). Seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/joss.03021

Yamamoto, T., Uchiumi, C., Suzuki, N., Yoshimoto, J., & Murillo-Rodriguez, E. (2020). The psychological impact of 'mild lockdown'in Japan during the COVID-19 pandemic: A nationwide survey under a declared state of emergency. *International journal of environmental research and public health*, *17*(24), 9382. https://doi.org/10.3390/ijerph17249382

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295–316. https://doi.org/10.1016/j.neucom.2020.07.061

Yarkoni, T., & Westfall, J. (2017a). Choosing prediction over explanation in psychology: Lessons from machine learning. *Perspectives on Psychological Science*, *12*(6), 1100–1122. https://doi.org/10.1177/1745691617693393

Yarkoni, T., & Westfall, J. (2017b). Choosing prediction over explanation in psychology: Lessons from machine learning. *Perspectives on Psychological Science*, *12*(6), 1114. https://doi.org/10.1177/1745691617693393

Zou, T., Hui ind Hastie. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *67*(2), 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x

Acronyms

AUC Area Under The Receiver Operating Characteristic Curve 7, 26, 40, 42

Brief COPE Brief Coping Orientation to Problems Experienced 5, 9, 11, 14, 41

CART Classification And Regression Trees 29

CI Confidence Interval 37

CV cross-validation 21–24, 39, 42, 44, 46

GBDT Gradient Boosted Decision Trees 7, 15, 22, 31, 32

HPO hyper parameter optimization 21, 23, 39, 42, 44, 46

L1 Lasso 26, 29

L2 Ridge 26, 29

LightGBM Light Gradient Boosting Machine 7, 14, 31, 32, 42, 45

MAE Mean Absolute Error 40

MDD Major Depressive Disorder 2, 3

ML Machine Learning 8, 21–23, 25, 26, 29, 32

MLE Maximum Likelihood Estimation 13, 26

MSE Mean Squared Error 20, 32

OLS Ordinary Least Squares 27, 29

OR Odds Ratio 13

ORs Odds Ratios 6

PHQ-9 Patient Health Questionnaire-9 5, 9–12, 14, 15, 19, 28, 49

RMSE Root Mean Squared Error 7, 40

SHAP SHapley Additive exPlanations 42, 44

STAXI State–Trait Anger Expression Inventory 5, 9, 11

VIF Variance Inflation Factor 19, 28, 36, 37

Source Code

A.1 Replication – Descriptive

Listing A.2: replication/descriptive/dataframe_creator.py

```
import pandas as pd
from scipy import stats

from constants.column_names import (
    ALL,
    DEMOGRAPHIC,
    DEPRESSION,
    NOT_DEPRESSION,
)


def get_significance_stars(pvalue: float, statistic: float) -> str:
    """Get significance stars."""
    if pvalue < 0.001:
        return f"{round(abs(statistic), 2)}***"
    elif pvalue < 0.01:
        return f"{round(abs(statistic), 2)}**"
    elif pvalue < 0.05:
        return f"{round(abs(statistic), 2)}*"
    else:
        return f"{round(abs(statistic), 2)}"


def create_df_for_continuous_column(column_name: str, df: pd.DataFrame) -> pd.
    DataFrame:
    """Compute interval statistics for a given column."""
    df_depression = df[df.Depression]
    df_not_depression = df[~df.Depression]

    # Compute Levene and T-tests
    levene_statistic, levene_pvalue = stats.levene(
        df_depression[column_name], df_not_depression[column_name]
    )
```

```
33      ttest_statistic, ttest_pvalue = stats.ttest_ind(
34          df_depression[column_name],
35          df_not_depression[column_name],
36          equal_var=levene_pvalue >= 0.05,
37      )
38
39      df_output: pd.DataFrame = pd.DataFrame.from_dict(
40          {
41              DEMOGRAPHIC: f"{column_name}, Mean (SD)",
42              ALL: f"{round(df[column_name].mean(), 2)} ({round(df[column_name].
                    std(), 2)})",
43              NOT_DEPRESSION: f"{round(df_not_depression[column_name].mean(), 2)}
                    ({round(df_not_depression[column_name].std(), 2)})",
44              DEPRESSION: f"{round(df_depression[column_name].mean(), 2)} ({round(
                    df_depression[column_name].std(), 2)})",
45              "t": [get_significance_stars(ttest_pvalue, ttest_statistic)],
46          }
47      )
48
49      return df_output
50
51
52  def create_df_for_categorical_column(
53      column_name: str,
54      dictionary: dict[int, str],
55      df: pd.DataFrame,
56  ) -> pd.DataFrame:
57      """Compute and append categorical statistics for a given column."""
58      df_depression = df[df.Depression]
59      df_not_depression = df[~df.Depression]
60
61      # Compute chi-square test
62      contingency_table = [
63          [
64              len(df_not_depression[df_not_depression[column_name] == key]),
65              len(df_depression[df_depression[column_name] == key]),
66          ]
67          for key in dictionary.keys()
68      ]
69      chisq, p, _, _ = stats.chi2_contingency(contingency_table, correction=False)
70
71      # Create results
72      df_output: pd.DataFrame = pd.DataFrame.from_dict(
73          {
74              DEMOGRAPHIC: rf"{column_name}, N (\%)",
75              ALL: "-",
76              NOT_DEPRESSION: "-",
77              DEPRESSION: "-",
78              "Chi_square": [get_significance_stars(p, chisq)],
79          }
80      )
81
```

```
82     for key, value in dictionary.items():
83         df_temp: pd.DataFrame = pd.DataFrame.from_dict(
84             {
85                 DEMOGRAPHIC: [f"{value}"],
86                 ALL: [
87                     f"{len(df[df[column_name] == key])} ({round(len(df[df[
                           column_name] == key]) / len(df) * 100, 2)})"
88                 ],
89                 NOT_DEPRESSION: [
90                     f"{len(df_not_depression[df_not_depression[column_name] ==
                           key])} ({round(len(df_not_depression[df_not_depression[
                           column_name] == key]) / len(df_not_depression) * 100, 2)
                           })"
91                 ],
92                 DEPRESSION: [
93                     f"{len(df_depression[df_depression[column_name] == key])} ({
                           round(len(df_depression[df_depression[column_name] == key
                           ]) / len(df_depression) * 100, 2)})"
94                 ],
95                 "Chi_square": ["-"],
96             },
97         )
98         df_output: pd.DataFrame = pd.concat([df_output, df_temp])
99
100    return df_output
```

Listing A.3: replication/descriptive/replication_descriptive_analysis.py

```
1  # Import necessary libraries
2  import pandas as pd
3  import scipy.stats as stats
4
5  from constants.column_mappings import (
6      ECONOMIC_INFLUENCE_DICT,
7      EMPLOYMENT_STATUS_DICT,
8      HOUSEHOLD_INCOME_DICT,
9      MARRIED_DICT,
10     RESIDENTIAL_AREA_DICT,
11     SEX_DICT,
12     UNDERLYING_DISEASE_DICT,
13 )
14 from constants.column_names import (
15     AGE,
16     ALL,
17     CATEGORICAL_COLUMNS,
18     CHI_SQUARE,
19     CONTINUOUS_COLUMNS,
20     DEMOGRAPHIC,
21     DEPRESSION,
22     DESCRIPTIVE_COLUMNS,
23     NOT_DEPRESSION,
24 )
```

```
25 from constants.constants import (
26     OUTPUT_FOLDER,
27     REPLICATION_FOLDER,
28     TABLE_S1_FOLDER,
29 )
30 from dataframe_creator import (
31     create_df_for_categorical_column,
32     create_df_for_continuous_column,
33     get_significance_stars,
34 )
35 from lib.output_writer import output_csv, output_latex
36 from lib.get_preprocessed_dataframe import get_preprocessed_dataframe
37
38
39 dictionaries = [
40     ECONOMIC_INFLUENCE_DICT,
41     EMPLOYMENT_STATUS_DICT,
42     HOUSEHOLD_INCOME_DICT,
43     MARRIED_DICT,
44     RESIDENTIAL_AREA_DICT,
45     UNDERLYING_DISEASE_DICT,
46     SEX_DICT,
47 ]
48
49
50 OUTPUT_PATH: str = f"{OUTPUT_FOLDER}/{REPLICATION_FOLDER}/{TABLE_S1_FOLDER}"
51
52 age_bins: list[int] = [20, 30, 40, 50, 60, 70, 80]
53
54
55 def main() -> None:
56     """
57     Main function to create and save descriptive analysis tables."""
58
59     preprocessed_dataframe: pd.DataFrame = get_preprocessed_dataframe()
60
61     # Compute descriptive statistics for categorical columns
62     df_categorical_output: pd.DataFrame = pd.DataFrame()
63
64     for column_name, dictionary in zip(CATEGORICAL_COLUMNS, dictionaries):
65         df_categorical_column: pd.DataFrame = create_df_for_categorical_column(
66             column_name, dictionary, preprocessed_dataframe
67         )
68         df_categorical_output = pd.concat(
69             [df_categorical_output, df_categorical_column],
70             ignore_index=True,
71             sort=False,
72         )
73
74     output_csv(df_categorical_output, f"{OUTPUT_PATH}/categorical_columns")
75     df_categorical_output.rename(columns={"Chi_square": r"$\chi"}, inplace=True)
76     output_latex(df_categorical_output, f"{OUTPUT_PATH}/categorical_columns")
```

```
77
78      # Compute descriptive statistics for continuous columns
79      df_continuous_output: pd.DataFrame = pd.DataFrame()
80
81      for column_name in CONTINUOUS_COLUMNS:
82          df_continuous_column: pd.DataFrame = create_df_for_continuous_column(
83              column_name, preprocessed_dataframe
84          )
85          df_continuous_output: pd.DataFrame = pd.concat(
86              [df_continuous_output, df_continuous_column], ignore_index=True,
                    sort=False
87          )
88
89      output_csv(df_continuous_output, f"{OUTPUT_PATH}/continuous_columns")
90      output_latex(df_continuous_output, f"{OUTPUT_PATH}/continuous_columns")
91
92      # Compute descriptive statistics for age groups
93      age_bin_labels: list[str] = []
94      age_bin_dataframe: pd.DataFrame = preprocessed_dataframe[[AGE, DEPRESSION]].
            copy()
95
96      for i in range(len(age_bins) - 1):
97          age_bin_label = f"{age_bins[i]}s"
98          age_bin_labels.append(age_bin_label)
99          age_bin_dataframe[age_bin_label] = (
100             (age_bin_dataframe[AGE] >= age_bins[i])
101             & (age_bin_dataframe[AGE] < age_bins[i + 1])
102         ).astype(int)
103
104     age_bins_contingency_table = [
105         [
106             len(
107                 age_bin_dataframe[
108                     (age_bin_dataframe[age_bin_label] == 1)
109                     & (age_bin_dataframe[DEPRESSION] == 0)
110                 ]
111             ),
112             len(
113                 age_bin_dataframe[
114                     (age_bin_dataframe[age_bin_label] == 1)
115                     & (age_bin_dataframe[DEPRESSION] == 1)
116                 ]
117             ),
118         ]
119         for age_bin_label in age_bin_labels
120     ]
121     chisq, p, _, _ = stats.chi2_contingency(
122         age_bins_contingency_table, correction=False
123     )
124
125     df_age_group: pd.DataFrame = pd.DataFrame.from_dict(
126         {
```

```
127             DEMOGRAPHIC: r"Age group, N, (\%), Chi-square",
128             ALL: "-",
129             NOT_DEPRESSION: "-",
130             DEPRESSION: "-",
131             CHI_SQUARE: [get_significance_stars(p, chisq)],
132         }
133     )
134
135     all_age_groups_not_depression: int = len(
136         age_bin_dataframe[age_bin_dataframe[DEPRESSION] == 0]
137     )
138     all_age_groups_depression: int = len(
139         age_bin_dataframe[age_bin_dataframe[DEPRESSION] == 1]
140     )
141     for i, age_bin_label in enumerate(age_bin_labels):
142         all: int = len(age_bin_dataframe[age_bin_dataframe[age_bin_label] == 1])
143         not_depression: int = len(
144             age_bin_dataframe[
145                 (age_bin_dataframe[age_bin_label] == 1)
146                 & (age_bin_dataframe[DEPRESSION] == 0)
147             ]
148         )
149         depression: int = len(
150             age_bin_dataframe[
151                 (age_bin_dataframe[age_bin_label] == 1)
152                 & (age_bin_dataframe[DEPRESSION] == 1)
153             ]
154         )
155         df_temp: pd.DataFrame = pd.DataFrame.from_dict(
156             {
157                 DEMOGRAPHIC: [f"{age_bin_label}"],
158                 ALL: [f"{all} ({round(all / len(age_bin_dataframe) * 100, 2)})"],
159                 NOT_DEPRESSION: [
160                     f"{not_depression} ({round(not_depression /
                            all_age_groups_not_depression * 100, 2)})"
161                 ],
162                 DEPRESSION: [
163                     f"{depression} ({round(depression / all_age_groups_depression
                            * 100, 2)})"
164                 ],
165                 CHI_SQUARE: ["-"],
166             }
167         )
168         df_age_group = pd.concat([df_age_group, df_temp], ignore_index=True)
169
170     output_csv(df_age_group, f"{OUTPUT_PATH}/age_groups")
171     df_age_group.rename(columns={"Chi_square": r"$\chi$"}, inplace=True)
172     output_latex(df_age_group, f"{OUTPUT_PATH}/age_groups")
173
174
175 if __name__ == "__main__":
176     main()
```

A.2 Replication – Logistic Regression

Listing A.4: replication/logistic_regression/replication_logistic_regression.py

```
# Import packages
import os
from sklearn.metrics import balanced_accuracy_score, roc_auc_score
import numpy as np
import openpyxl
import pandas as pd
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor
from constants.constants import (
    OUTPUT_FOLDER,
    REPLICATION_FOLDER,
    TABLE_S2_FOLDER,
)
from lib.output_writer import output_csv, output_latex
from lib.prepare_data import prepare_data

OUTPUT_PATH: str = (
    f"{OUTPUT_FOLDER}/{REPLICATION_FOLDER}/{TABLE_S2_FOLDER}/
        logistic_regression_results"
)


def compute_logistic_regression() -> None:
    X, y = prepare_data(
        classification=True, drop_household_income_non_response_participants=
            True
    )
    X: pd.DataFrame = sm.add_constant(X) # type: ignore
    print("Columns in X:", X.columns.tolist())

    model = sm.Logit(y, X).fit()

    summary = model.summary2().tables[1]

    coefficients = summary["Coef."]
    standard_errors = summary["Std.Err."]
    wald_stats = ((coefficients / standard_errors) ** 2).round(
        2
    ) # Wald = (Coef / SE)^2
    odds_ratios = np.exp(coefficients).round(2) # OR = exp(Coefficients)
    conf_int = np.exp(summary[["[0.025", "0.975]"]]).round(
        2
    ) # Transform CI to OR scale and round

    # Compute Variance Inflation Factor (VIF) for predictors
    vif_predictors = [
        round(variance_inflation_factor(X.values, i), 2) for i in range(X.shape
            [1])
    ]
```

```
47
48      # Box-Tidwell test for linearity (continuous variables only)
49      continuous_vars = [
50          "Age",
51          "state_anger_score",
52          "anger_control_score",
53          "behavioural_disengagement_score",
54          "denial_score",
55          "humour_score",
56          "instrumental_support_score",
57          "planning_score",
58          "self_blame_score",
59      ]
60
61      box_tidwell_p = []
62      for col in X.columns:
63          if col in continuous_vars and col in X.columns:
64              # Create interaction term with log(X) for Box-Tidwell test
65              X_bt = X.copy()
66              log_var = np.log(X[col] + 0.001) # Add small constant to avoid log
                    (0)
67              X_bt[f"{col}_log"] = X[col] * log_var
68
69              try:
70                  model_bt = sm.Logit(y, X_bt).fit(disp=0)
71                  bt_p = model_bt.pvalues[f"{col}_log"]
72                  box_tidwell_p.append(f"{bt_p:.4f}")
73              except:
74                  box_tidwell_p.append("N/A")
75          else:
76              box_tidwell_p.append("N/A")
77
78      # Cook's distance (influence measure)
79      influence = model.get_influence()
80      cooks_d = influence.cooks_distance[0]
81      max_cooks_d = np.max(cooks_d)
82
83      # Count influential observations (Cook's D > 4/n)
84      threshold = 4 / len(y)
85      influential_obs = np.sum(cooks_d > threshold)
86
87      y_pred_prob = model.predict(X)
88      y_pred_label = (y_pred_prob >= 0.5).astype(int)
89      balanced_acc = balanced_accuracy_score(y, y_pred_label)
90      auc = roc_auc_score(y, y_pred_prob)
91      print(f"Logistic regression balanced accuracy (in-sample): {balanced_acc:.4f
            }")
92      print(f"Logistic regression AUC (in-sample): {auc:.4f}")
93      print(f"Max Cook's distance: {max_cooks_d:.4f}")
94      print(f"Influential observations (Cook's D > {threshold:.4f}): {
            influential_obs}")
95
```

```
96      # Create the final dataframe with the required columns, including VIF (
            intercept excluded)
97      final_df = pd.DataFrame(
98          {
99              "Predictor variable": X.columns,
100             "Beta": coefficients.round(2),
101             "SE": standard_errors.round(2),
102             "Wald": wald_stats,
103             "OR": odds_ratios,
104             "95% CI": [f"[{ci[0]}, {ci[1]}]" for ci in conf_int.values],
105             "P-value": summary["P>|z|"].apply(lambda x: f"{x:.4f}"),
106             "VIF": vif_predictors,
107             "Box-Tidwell p": box_tidwell_p,
108         }
109     )
110     significance_stars = summary["P>|z|"].apply(
111         lambda p: "***" if p < 0.001 else "**" if p < 0.01 else "*" if p < 0.05
                else ""
112     )
113     final_df["95% CI"] = [
114         f"{ci[0]:.2f} - {ci[1]:.2f}{star}"
115         for ci, star in zip(conf_int.values, significance_stars)
116     ]
117
118     os.makedirs(os.path.dirname(OUTPUT_PATH), exist_ok=True)
119     latex_mapping = {
120         "CI 95%": r"CI_{95\%}",
121     }
122     output_csv(df=final_df, output_path=OUTPUT_PATH)
123     with open(f"{OUTPUT_PATH}.csv", "a") as f:
124         f.write(f"\nBalanced Accuracy (in-sample),,,,,,{balanced_acc:.4f}\n")
125         f.write(f"AUC (in-sample),,,,,,{auc:.4f}\n")
126         f.write(f"Max Cook's distance,,,,,,{max_cooks_d:.4f}\n")
127         f.write(f"Influential observations,,,,,,{influential_obs}\n")
128
129     final_df.rename(columns=latex_mapping, inplace=True)
130     output_latex(
131         df=final_df,
132         output_path=OUTPUT_PATH,
133     )
134     with open(f"{OUTPUT_PATH}.tex", "a") as f:
135         f.write(f"% Balanced Accuracy (in-sample): {balanced_acc:.4f}\n")
136         f.write(f"% AUC (in-sample): {auc:.4f}\n")
137         f.write(f"% Max Cook's distance: {max_cooks_d:.4f}\n")
138         f.write(f"% Influential observations: {influential_obs}\n")
139
140
141 if __name__ == "__main__":
142     compute_logistic_regression()
```

Listing A.5: replication/logistic_regression/pre_analysis.py

```
from typing import List
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor
from statsmodels.stats.multitest import multipletests

from matplotlib import pyplot as plt
import numpy as np
import seaborn as sns
import pandas as pd

from constants.column_names import (
    CONTINUOUS_COLUMNS,
)
from constants.constants import (
    OUTPUT_FOLDER,
    PRE_ANALYSIS_FOLDER,
    REPLICATION_FOLDER,
)
from lib.output_writer import output_csv, output_latex
from lib.prepare_data import prepare_data


def pre_analysis() -> None:
    X, y = prepare_data(
        classification=True, drop_household_income_non_response_participants=
            True
    )

    correlation_matrix: pd.DataFrame = X.corr()
    save_correlation_matrix_as_heatmap(X, correlation_matrix)

    X_full: pd.DataFrame = sm.add_constant(X) # type: ignore

    # Variance Inflation Factor (VIF)
    vif_df = pd.DataFrame(
        {
            "Predictor": list(X_full.columns),
            "VIF": [
                round(variance_inflation_factor(X_full.values, i), 2)
                for i in range(X_full.shape[1])
            ],
        }
    )

    # Cook's Distance using GLM with binomial family
    model_full = sm.GLM(y, X_full, family=sm.families.Binomial()).fit()
    influence_full = model_full.get_influence()
    cooks_d_full = influence_full.cooks_distance[0]

    # Box-Tidwell only for continuous predictors (with all other predictors as
        controls)
    box_tidwell_results = []
```

```
    for predictor in X.columns:
        if predictor not in CONTINUOUS_COLUMNS:
            continue

        predictor_log = f"{predictor}_log"

        if (X[predictor] == 0).any():
            print(f"{predictor} contains zero values.")

        X_bt = X_full.copy()
        X_bt[predictor_log] = X[predictor] * np.log(X[predictor] + 1e-6)

        model_bt = sm.Logit(y, X_bt).fit(disp=False)
        log_coef = model_bt.params[predictor_log]
        log_pval = model_bt.pvalues[predictor_log]

        box_tidwell_results.append(
            {
                "Predictor": predictor,
                "Predictor x Log Interaction Coefficient": log_coef,
                "P value": log_pval,
            }
        )

    box_tidwell_df = pd.DataFrame(box_tidwell_results)

    if len(box_tidwell_df) > 0:
        valid_pvals = box_tidwell_df["P value"].dropna()
        if len(valid_pvals) > 0:
            valid_indices = box_tidwell_df["P value"].notna()
            rejected, pvals_corrected, alpha_sidak, alpha_bonf = multipletests(
                valid_pvals, alpha=0.05, method="fdr_bh"
            )
            box_tidwell_df.loc[valid_indices, "P value (FDR corrected)"] = (
                pvals_corrected
            )
        else:
            box_tidwell_df["P value (FDR corrected)"] = np.nan
    else:
        box_tidwell_df["P value (FDR corrected)"] = np.nan
    assumption_measures_df = pd.DataFrame(
        {
            "Predictor": vif_df["Predictor"],
            "VIF": vif_df["VIF"],
            "Max Cook's D": np.max(cooks_d_full),
        }
    )
    assumption_measures_df = assumption_measures_df.merge(
        box_tidwell_df[
            [
                "Predictor",
                "Predictor x Log Interaction Coefficient",
```

```
103                 "P value",
104                 "P value (FDR corrected)",
105             ]
106         ],
107         on="Predictor",
108         how="left",
109     )
110
111     output_csv(
112         df=assumption_measures_df,
113         output_path=f"{OUTPUT_FOLDER}/{REPLICATION_FOLDER}/{PRE_ANALYSIS_FOLDER
                }/assumption_measures_logistic_regression",
114     )
115     output_latex(
116         df=assumption_measures_df,
117         output_path=f"{OUTPUT_FOLDER}/{REPLICATION_FOLDER}/{PRE_ANALYSIS_FOLDER
                }/assumption_measures_logistic_regression",
118     )
119
120
121 def save_correlation_matrix_as_heatmap(X, correlation_matrix):
122     plt.figure(figsize=(10, 8))
123     sns.heatmap(
124         correlation_matrix,
125         annot=True,
126         fmt=".2f",
127         cmap="coolwarm",
128         xticklabels=list(X.columns),
129         yticklabels=list(X.columns),
130         cbar=True,
131         square=True,
132         linewidths=0.5,
133         annot_kws={"size": 8},
134     )
135     plt.title("Correlation Matrix of Predictors")
136     plt.xticks(rotation=45, ha="right")
137     plt.yticks(rotation=0)
138     plt.tight_layout()
139     plt.savefig(
140         f"{OUTPUT_FOLDER}/{REPLICATION_FOLDER}/{PRE_ANALYSIS_FOLDER}/
                correlation_matrix_predictors.png",
141         bbox_inches="tight",
142         pad_inches=0,
143     )
144     plt.close()
145
146
147 if __name__ == "__main__":
148     pre_analysis()
```

A.3 Constants

Listing A.6: constants/column_names.py

```
# Column names
from typing import List

from constants.column_mappings import (
    ECONOMIC_INFLUENCE_DICT,
    EMPLOYMENT_STATUS_DICT,
    HOUSEHOLD_INCOME_DICT,
    MARRIED_DICT,
    UNDERLYING_DISEASE_DICT,
)


ALL: str = "All"
AGE: str = "Age"
ANGER_CONTROL_SCORE = "anger_control_score"
CHI_SQUARE: str = "Chi_square"
DEMOGRAPHIC: str = "Demographic"
DEPRESSION: str = "Depression"
ECONOMIC_INFLUENCE: str = "Economic influence"
EMPLOYMENT_STATUS: str = "Employment status"
HOUSEHOLD_INCOME: str = "Household income"
MARRIED: str = "Married"
NOT_DEPRESSION: str = "Not Depression"
RESIDENTIAL_AREAS: str = "Residential areas"
PHQ9_SCORE: str = "PHQ9_score"
SEX: str = "Sex"
STATE_ANGER_SCORE = "state_anger_score"
UNDERLYING_DISEASE: str = "Underlying disease"

# BRIEF COPE column names
SELF_DISTRACTION_SCORE: str = "self_distraction_score"
ACTIVE_COPING_SCORE: str = "active_coping_score"
DENIAL_SCORE: str = "denial_score"
SUBSTANCE_USE_SCORE: str = "substance_use_score"
EMOTIONAL_SUPPORT_SCORE: str = "emotional_support_score"
INSTRUMENTAL_SUPPORT_SCORE: str = "instrumental_support_score"
BEHAVIOURAL_DISENGAGEMENT_SCORE: str = "behavioural_disengagement_score"
VENTING_SCORE: str = "venting_score"
POSITIVE_REFRAMING_SCORE: str = "positive_reframing_score"
PLANNING_SCORE: str = "planning_score"
HUMOUR_SCORE: str = "humour_score"
ACCEPTANCE_SCORE: str = "acceptance_score"
RELIGION_SCORE: str = "religion_score"
SELF_BLAME_SCORE: str = "self_blame_score"

CATEGORICAL_COLUMNS: List[str] = [
    ECONOMIC_INFLUENCE,
    EMPLOYMENT_STATUS,
    HOUSEHOLD_INCOME,
```

```
50     MARRIED,
51     RESIDENTIAL_AREAS,
52     SEX,
53     UNDERLYING_DISEASE,
54 ]
55
56 I_DO_NOT_KNOW_VALUE: int = 4
57 NON_RESPONSE_VALUE: int = 5
58
59 REFERENCE_IGNORED_COLUMNS: List[str] = [
60     "Without impact", # Economic influence reference category
61     "Full-time worker", # Employment status reference category
62     "< 2 million JPY", # Household income reference category
63     "Single", # Married reference category
64     "Without", # Underlying disease reference category
65     "I do not know", # Household income non-response
66     "Non-response", # Household income non-response
67 ]
68
69 # Residential areas and Sex are not needed for analysis, filter out reference
       columns
70 CATEGORICAL_COLUMNS_DUMMIES: List[str] = (
71     [v for v in ECONOMIC_INFLUENCE_DICT.values() if v not in
           REFERENCE_IGNORED_COLUMNS]
72     + [v for v in EMPLOYMENT_STATUS_DICT.values() if v not in
           REFERENCE_IGNORED_COLUMNS]
73     + [v for v in HOUSEHOLD_INCOME_DICT.values() if v not in
           REFERENCE_IGNORED_COLUMNS]
74     + [v for v in MARRIED_DICT.values() if v not in REFERENCE_IGNORED_COLUMNS]
75     + [
76         v
77         for v in UNDERLYING_DISEASE_DICT.values()
78         if v not in REFERENCE_IGNORED_COLUMNS
79     ]
80 )
81
82
83 CONTINUOUS_COLUMNS: List[str] = [
84     AGE,
85     PHQ9_SCORE,
86     STATE_ANGER_SCORE,
87     ANGER_CONTROL_SCORE,
88     ACCEPTANCE_SCORE,
89     ACTIVE_COPING_SCORE,
90     BEHAVIOURAL_DISENGAGEMENT_SCORE,
91     DENIAL_SCORE,
92     EMOTIONAL_SUPPORT_SCORE,
93     HUMOUR_SCORE,
94     INSTRUMENTAL_SUPPORT_SCORE,
95     PLANNING_SCORE,
96     POSITIVE_REFRAMING_SCORE,
97     RELIGION_SCORE,
```

```
    SELF_BLAME_SCORE,
    SELF_DISTRACTION_SCORE,
    SUBSTANCE_USE_SCORE,
    VENTING_SCORE,
]

ANALYSIS_COLUMNS: List[str] = CATEGORICAL_COLUMNS_DUMMIES + CONTINUOUS_COLUMNS
DESCRIPTIVE_COLUMNS: List[str] = CATEGORICAL_COLUMNS + CONTINUOUS_COLUMNS

LOGISTIC_REGRESSION_IGNORED_COPING_STRATEGY_COLUMNS: List[str] = [
    ACCEPTANCE_SCORE,
    ACTIVE_COPING_SCORE,
    EMOTIONAL_SUPPORT_SCORE,
    POSITIVE_REFRAMING_SCORE,
    RELIGION_SCORE,
    SELF_DISTRACTION_SCORE,
    SUBSTANCE_USE_SCORE,
    VENTING_SCORE,
]


LOGISTIC_REGRESSION_PREDICTOR_COLUMNS: List[str] = CATEGORICAL_COLUMNS_DUMMIES
    + [
    column
    for column in CONTINUOUS_COLUMNS
    if column not in LOGISTIC_REGRESSION_IGNORED_COPING_STRATEGY_COLUMNS
    and column != PHQ9_SCORE
]
```

Listing A.7: constants/column_mappings.py

```
SEX_DICT: dict[int, str] = {0: "Male", 1: "Female"}
EMPLOYMENT_STATUS_DICT: dict[int, str] = {
    0: "Full-time worker",
    1: "Homemaker",
    2: "Not working",
    3: "No regular employment",
}
RESIDENTIAL_AREA_DICT: dict[int, str] = {
    0: "Tokyo",
    1: "Hokkaido",
    8: "Ibaraki",
    11: "Saitama",
    12: "Chiba",
    14: "Kanagawa",
    17: "Ishikawa",
    21: "Gifu",
    23: "Aichi",
    26: "Kyoto",
    27: "Osaka",
    28: "Hyogo",
    40: "Fukuoka",
```

```
}
UNDERLYING_DISEASE_DICT: dict[int, str] = {0: "Without", 1: "With"}
MARRIED_DICT: dict[int, str] = {1: "Single", 2: "Married"}
HOUSEHOLD_INCOME_DICT: dict[int, str] = {
    1: "< 2 million JPY",
    2: "2 - 8 million JPY",
    3: "> 8 million JPY",
    4: "I do not know",
    5: "Non-response",
}
ECONOMIC_INFLUENCE_DICT: dict[int, str] = {
    0: "Without impact",
    1: "Negative impact",
    3: "Positive impact",
}

# https://novopsych.com.au/assessments/formulation/brief-cope/
COPING_STRATEGY_NAME_INDEX_TUPLE: dict[str, tuple[int, int]] = {
    "self_distraction_score": (1, 19),
    "active_coping_score": (2, 7),
    "denial_score": (3, 8),
    "substance_use_score": (4, 11),
    "emotional_support_score": (5, 15),
    "instrumental_support_score": (10, 23),
    "behavioural_disengagement_score": (6, 16),
    "venting_score": (9, 21),
    "positive_reframing_score": (12, 17),
    "planning_score": (14, 25),
    "humour_score": (18, 28),
    "acceptance_score": (20, 24),
    "religion_score": (22, 27),
    "self_blame_score": (13, 26),
}
```

Listing A.8: constants/constants.py

```
import os

# File names
DATA_FILE_NAME: str = "raw_data.xlsx"

# Folder names
PROJECT_ROOT = os.path.abspath(os.path.join(os.path.dirname(__file__), "..", "
    .."))
DATA_FOLDER = os.path.join(PROJECT_ROOT, "resources/original_paper/")
SOURCE_FOLDER: str = "src"
OUTPUT_FOLDER: str = "output"
MACHINE_LEARNING_FOLDER: str = "machine_learning"
REPLICATION_FOLDER: str = "replication"
TABLE_S1_FOLDER: str = "table1"
TABLE_S2_FOLDER: str = "table2"
PRE_ANALYSIS_FOLDER: str = "pre_analysis"
```

```
16
17 # Variable names
18 PHQ_CUTOFF: int = 10
19
20 RAW_DATA_PATH = os.path.join(DATA_FOLDER, DATA_FILE_NAME)
```

A.4 Library

Listing A.9: lib/Logger.py

```
1 class Logger:
2     def __init__(self, log_file: str):
3         self.log_file = log_file
4         with open(self.log_file, "w") as f:
5             f.write("Logger initialized.\n")
6
7     def log(self, message: str):
8         with open(self.log_file, "a") as f:
9             f.write(f"{message}\n")
```

Listing A.10: lib/get_dummy_dataframe.py

```
1 import pandas as pd
2
3 from constants.column_mappings import (
4     ECONOMIC_INFLUENCE_DICT,
5     EMPLOYMENT_STATUS_DICT,
6     HOUSEHOLD_INCOME_DICT,
7     MARRIED_DICT,
8     UNDERLYING_DISEASE_DICT,
9 )
10 from constants.column_names import (
11     ECONOMIC_INFLUENCE,
12     EMPLOYMENT_STATUS,
13     HOUSEHOLD_INCOME,
14     MARRIED,
15     REFERENCE_IGNORED_COLUMNS,
16     SEX,
17     UNDERLYING_DISEASE,
18 )
19
20
21 def get_dummy_dataframe(
22     df: pd.DataFrame,
23 ) -> pd.DataFrame:
24     df_copy: pd.DataFrame = df.copy()
25
26     # Replace categorical keys with values
27     df_copy.replace(
28         {
```

```
29              ECONOMIC_INFLUENCE: ECONOMIC_INFLUENCE_DICT,
30              EMPLOYMENT_STATUS: EMPLOYMENT_STATUS_DICT,
31              HOUSEHOLD_INCOME: HOUSEHOLD_INCOME_DICT,
32              MARRIED: MARRIED_DICT,
33              UNDERLYING_DISEASE: UNDERLYING_DISEASE_DICT,
34          },
35          inplace=True,
36      )
37
38      # Dummy all categorical variables
39      economic_influence: pd.DataFrame = pd.get_dummies(df_copy[ECONOMIC_INFLUENCE
            ])
40      employment_status: pd.DataFrame = pd.get_dummies(df_copy[EMPLOYMENT_STATUS])
41      household_income: pd.DataFrame = pd.get_dummies(df_copy[HOUSEHOLD_INCOME])
42      married: pd.DataFrame = pd.get_dummies(df_copy[MARRIED])
43      underlying_disease: pd.DataFrame = pd.get_dummies(df_copy[UNDERLYING_DISEASE
            ])
44
45      # Drop the original categorical columns
46      df_copy.drop(
47          columns=[
48              ECONOMIC_INFLUENCE,
49              EMPLOYMENT_STATUS,
50              HOUSEHOLD_INCOME,
51              MARRIED,
52              UNDERLYING_DISEASE,
53          ],
54          inplace=True,
55      )
56
57      # Add dummies to dataframe
58      df_dummies: pd.DataFrame = pd.concat(
59          [
60              df_copy,
61              underlying_disease,
62              married,
63              employment_status,
64              household_income,
65              economic_influence,
66          ],
67          axis=1,
68      )
69
70      # Drop reference categories and ignored columns
71      df_dummies.drop(columns=REFERENCE_IGNORED_COLUMNS, inplace=True, errors="
            ignore")
72
73      return df_dummies
```

Listing A.11: lib/output_writer.py

```
1 import pandas as pd
```

```

def output_csv(df: pd.DataFrame, output_path: str) -> None:
    """Output a DataFrame to a csv file."""
    df.to_csv(f"{output_path}.csv", index=False)
    print(f"Table saved to {output_path}.csv")


def output_latex(df: pd.DataFrame, output_path: str) -> None:
    """Output a DataFrame to a latex file."""
    df.to_latex(f"{output_path}.tex", index=False, float_format="%.2f")
    print(f"Table saved to {output_path}.tex")
```

Listing A.12: lib/prepare_data.py

```
from typing import Tuple

import pandas as pd
from constants.column_names import (
    DEPRESSION,
    HOUSEHOLD_INCOME,
    I_DO_NOT_KNOW_VALUE,
    LOGISTIC_REGRESSION_PREDICTOR_COLUMNS,
    NON_RESPONSE_VALUE,
)
from lib.get_dummy_dataframe import get_dummy_dataframe
from lib.get_preprocessed_dataframe import get_preprocessed_dataframe


def prepare_data(
    classification: bool, drop_household_income_non_response_participants: bool
        = False
) -> Tuple[pd.DataFrame, pd.Series]:
    df_preprocessed = get_preprocessed_dataframe(
        binarize_depression_score=classification
    )

    if drop_household_income_non_response_participants:
        df_preprocessed = df_preprocessed[
            df_preprocessed[HOUSEHOLD_INCOME] != NON_RESPONSE_VALUE
        ]
        df_preprocessed = df_preprocessed[
            df_preprocessed[HOUSEHOLD_INCOME] != I_DO_NOT_KNOW_VALUE
        ]

    df_dummies = get_dummy_dataframe(df_preprocessed)

    X = df_dummies[LOGISTIC_REGRESSION_PREDICTOR_COLUMNS].copy()
    for col in X.columns:
        if X[col].dtype == "bool":
            X[col] = X[col].astype("int")
    y = df_dummies[DEPRESSION]
```

```
37
38      return X, y
```

Listing A.13: lib/get_preprocessed_dataframe.py

```
1  from typing import List
2  import pandas as pd
3
4  from constants.column_mappings import COPING_STRATEGY_NAME_INDEX_TUPLE
5  from constants.column_names import (
6      ANGER_CONTROL_SCORE,
7      DEPRESSION,
8      PHQ9_SCORE,
9      STATE_ANGER_SCORE,
10 )
11 from constants.constants import DATA_FILE_NAME, DATA_FOLDER, PHQ_CUTOFF,
       SOURCE_FOLDER
12
13
14 def get_preprocessed_dataframe(binarize_depression_score: bool = True) -> pd.
       DataFrame:
15     df: pd.DataFrame = pd.read_excel(f"{DATA_FOLDER}/{DATA_FILE_NAME}")
16
17     df.columns = pd.Index(map(lambda x: x.splitlines()[0], df.columns))
18
19     phq_columns: List[str] = [f"PHQ{i}" for i in range(1, 10)]
20     state_anger_columns: List[str] = [f"State Anger{i}" for i in range(1, 11)]
21     anger_control_columns: List[str] = [f"Anger Control{i}" for i in range(1, 8)
           ]
22
23     df[PHQ9_SCORE] = df[phq_columns].sum(axis=1)
24     df[STATE_ANGER_SCORE] = df[state_anger_columns].sum(axis=1)
25     df[ANGER_CONTROL_SCORE] = df[anger_control_columns].sum(axis=1)
26
27     # Create score columns for coping strategies
28     for coping_strategy_name, index_tuple in COPING_STRATEGY_NAME_INDEX_TUPLE.
           items():
29         df[coping_strategy_name] = (
30             df.loc[
31                 :,
32                 f"BriefCOPE{
33             index_tuple[0]}",
34             ]
35             + df.loc[:, f"BriefCOPE{index_tuple[1]}"]
36         )
37
38     # Drop the individual variables related to Brief Cope, State Anger, and PHQ9
           , but keep the score columns
39     drop_columns = [
40         column
41         for column in df.columns
42         if (
```

```
43              "phq" in column.lower()
44              or "state anger" in column.lower()
45              or "anger control" in column.lower()
46              or "briefcope" in column.lower()
47          )
48          and "score" not in column.lower()
49      ]
50      df.drop(columns=drop_columns, inplace=True)
51
52      # Determine if participant is considered depressive
53      df[DEPRESSION] = (
54          df[PHQ9_SCORE] >= PHQ_CUTOFF if binarize_depression_score else df[
                PHQ9_SCORE]
55      )
56
57      # Check for missing values
58      if df.isnull().sum().sum() != 0:
59          raise ValueError("Input error: Missing values")
60      # Check if all participants have given consent
61      if not df["Participant consent"].eq(1).all():
62          raise ValueError("Input error: All participants need to consent")
63      # Drop irrelevant columns
64      irrelevant_columns = ["Sampleid", "Answerdate", "Participant consent"]
65      df.drop(columns=irrelevant_columns, inplace=True)
66
67      return df
```

A.5 Machine Learning

Listing A.14: machine_learning/optuna_regression.py

```
1  import numpy as np
2  import pandas as pd
3  from sklearn.model_selection import KFold, StratifiedKFold
4  from sklearn.linear_model import LogisticRegression, ElasticNet
5  from sklearn.metrics import (
6      balanced_accuracy_score,
7      mean_absolute_error,
8      mean_squared_error,
9  )
10 from sklearn.preprocessing import StandardScaler
11 from constants.constants import MACHINE_LEARNING_FOLDER, OUTPUT_FOLDER
12 from lib.Logger import Logger
13 from lib.prepare_data import prepare_data
14 import optuna
15 from sklearn.metrics import roc_auc_score, confusion_matrix
16
17 CV_OUTER = 10
18 RANDOM_STATE = 42
19
20
```

```
21 def nested_cv_logistic_regression(X: pd.DataFrame, y: pd.Series):
22     logger: Logger = Logger(
23         f"{OUTPUT_FOLDER}/{MACHINE_LEARNING_FOLDER}/
               optuna_logistic_regression_results.txt"
24     )
25
26     outer_cv = StratifiedKFold(
27         n_splits=CV_OUTER, shuffle=True, random_state=RANDOM_STATE
28     )
29
30     scores = []
31     aucs = []
32     best_params = []
33
34     logger.log("Optuna Nested CV Logistic Regression (Classifier)\n")
35
36     for fold, (train_idx, test_idx) in enumerate(outer_cv.split(X, y)):
37         X_train, X_test = X.iloc[train_idx], X.iloc[test_idx]
38         y_train, y_test = y.iloc[train_idx], y.iloc[test_idx]
39
40         scaler = StandardScaler()
41         X_train_scaled = pd.DataFrame(
42             scaler.fit_transform(X_train), columns=X_train.columns, index=
                   X_train.index
43         )
44         X_test_scaled = pd.DataFrame(
45             scaler.transform(X_test), columns=X_test.columns, index=X_test.index
46         )
47
48         def objective(trial):
49             C = trial.suggest_float("C", 1e-3, 10, log=True)
50
51             penalty = trial.suggest_categorical("penalty", ["l1", "l2"])
52
53             solver = "liblinear" if penalty == "l1" else "lbfgs"
54
55             model = LogisticRegression(
56                 C=C, penalty=penalty, solver=solver, max_iter=10000
57             )
58
59             model.fit(X_train_scaled, y_train)
60
61             y_pred = model.predict(X_test_scaled)
62
63             return balanced_accuracy_score(y_test, y_pred)
64
65         study = optuna.create_study(direction="maximize")
66
67         study.optimize(objective, n_trials=20, show_progress_bar=False) # type:
               ignore
68
69         best_score = study.best_value
```

```
70          best_param = study.best_params
71          best_param["solver"] = "liblinear" if best_param["penalty"] == "l1" else
                "lbfgs"
72
73          # Refit with best params to get probabilities and confusion matrix
74          model = LogisticRegression(
75              C=best_param["C"],
76              penalty=best_param["penalty"],
77              solver=best_param["solver"],
78              max_iter=10000,
79          )
80          model.fit(X_train_scaled, y_train)
81          y_pred = model.predict(X_test_scaled)
82          y_proba = model.predict_proba(X_test_scaled)[:, 1]
83          auc = roc_auc_score(y_test, y_proba)
84          cm = confusion_matrix(y_test, y_pred)
85
86          scores.append(best_score)
87          aucs.append(auc)
88          best_params.append(best_param)
89
90          logger.log(
91              f"Fold {fold+1}: Balanced Accuracy = {best_score:.4f}, AUC = {auc:.4
                    f}, Best Params = {best_param}\n"
92          )
93          logger.log(f"Fold {fold+1} Confusion Matrix:\n{cm}\n")
94
95      mean = np.mean(scores)
96      std = np.std(scores)
97      mean_auc = np.mean(aucs)
98      std_auc = np.std(aucs)
99
100     logger.log(f"\nNested CV Mean Balanced Accuracy: {mean:.4f} ±{std:.4f}\n")
101     logger.log(f"Nested CV Mean AUC: {mean_auc:.4f} ±{std_auc:.4f}\n")
102     logger.log(f"Best params per fold: {best_params}\n")
103
104     return scores, aucs, best_params, mean, mean_auc
105
106
107 def nested_cv_linear_regression(X: pd.DataFrame, y: pd.Series):
108     logger: Logger = Logger(
109         f"{OUTPUT_FOLDER}/{MACHINE_LEARNING_FOLDER}/
                optuna_elasticnet_regression_results.txt"
110     )
111
112     outer_cv = KFold(n_splits=CV_OUTER, shuffle=True, random_state=RANDOM_STATE)
113
114     scores = []
115     r2_scores = []
116     mae_scores = []
117     best_params = []
118
```

```
119     logger.log("Optuna Nested CV ElasticNet Regression (Regressor)\n")
120
121     for fold, (train_idx, test_idx) in enumerate(outer_cv.split(X, y)):
122         X_train, X_test = X.iloc[train_idx], X.iloc[test_idx]
123         y_train, y_test = y.iloc[train_idx], y.iloc[test_idx]
124
125         # Standardize features within each fold
126         scaler = StandardScaler()
127         X_train_scaled = pd.DataFrame(
128             scaler.fit_transform(X_train), columns=X_train.columns, index=
                    X_train.index
129         )
130         X_test_scaled = pd.DataFrame(
131             scaler.transform(X_test), columns=X_test.columns, index=X_test.index
132         )
133
134         def objective(trial):
135             alpha = trial.suggest_float("alpha", 1e-4, 10, log=True)
136             l1_ratio = trial.suggest_float("l1_ratio", 0.0, 1.0)
137             model = ElasticNet(alpha=alpha, l1_ratio=l1_ratio, max_iter=10000)
138
139             model.fit(X_train_scaled, y_train)
140
141             y_pred = model.predict(X_test_scaled)
142
143             return -np.sqrt(mean_squared_error(y_test, y_pred))
144
145         study = optuna.create_study(direction="maximize")
146         study.optimize(objective, n_trials=20, show_progress_bar=False)
147
148         best_score = study.best_value
149         best_param = study.best_params
150
151         # Refit with best params to get R^2 and MAE
152         model = ElasticNet(
153             alpha=best_param["alpha"], l1_ratio=best_param["l1_ratio"], max_iter
                    =10000
154         )
155         model.fit(X_train_scaled, y_train)
156         y_pred = model.predict(X_test_scaled)
157         r2 = model.score(X_test_scaled, y_test)
158         mae = mean_absolute_error(y_test, y_pred)
159
160         scores.append(best_score)
161         r2_scores.append(r2)
162         mae_scores.append(mae)
163         best_params.append(best_param)
164
165         logger.log(
166             f"Fold {fold+1}: neg_RMSE = {best_score:.4f}, Best Params = {
                    best_param}\n"
167         )
```

```
168
169     mean_rmse = np.mean(scores)
170     std_rmse = np.std(scores)
171     mean_r2 = np.mean(r2_scores)
172     std_r2 = np.std(r2_scores)
173     mean_mae = np.mean(mae_scores)
174     std_mae = np.std(mae_scores)
175
176     logger.log(f"\nNested CV Mean neg_RMSE: {mean_rmse:.4f} ±{std_rmse:.4f}\n")
177     logger.log(f"Nested CV Mean R²: {mean_r2:.4f} ±{std_r2:.4f}\n")
178     logger.log(f"Nested CV Mean MAE: {mean_mae:.4f} ±{std_mae:.4f}\n")
179     logger.log(f"Best params per fold: {best_params}\n")
180
181     return scores, r2_scores, mae_scores, best_params, mean_rmse, mean_r2,
            mean_mae
182
183
184 X_class, y_class = prepare_data(classification=True)
185 nested_cv_logistic_regression(X_class, y_class)
186 X_reg, y_reg = prepare_data(classification=False)
187 nested_cv_linear_regression(X_reg, y_reg)
```

Listing A.15: machine_learning/simple_linear_regression.py

```
 1 import numpy as np
 2 import pandas as pd
 3 from sklearn.metrics import mean_squared_error
 4 from sklearn.preprocessing import StandardScaler
 5 import statsmodels.api as sm
 6 import matplotlib.pyplot as plt
 7
 8 from lib.output_writer import output_csv, output_latex
 9 from constants.constants import (
10     MACHINE_LEARNING_FOLDER,
11     OUTPUT_FOLDER,
12 )
13 from lib.prepare_data import prepare_data
14
15
16 def main():
17     X, y = prepare_data(classification=False)
18
19     # Convert boolean columns to int before scaling
20     X = X.astype({col: "int" for col in X.select_dtypes(include=["bool"]).
           columns})
21
22     # Standardize features (excluding constant)
23     scaler = StandardScaler()
24     X_scaled = pd.DataFrame(
25         scaler.fit_transform(X),
26         columns=X.columns,
27         index=X.index
```

```
28      )
29
30      # Add constant after scaling
31      X: pd.DataFrame = sm.add_constant(X_scaled) # type: ignore
32
33      model = sm.OLS(y, X).fit()
34
35      coefficients = model.params
36      standard_errors = model.bse
37      conf_int = model.conf_int()
38      wald_stats = ((coefficients / standard_errors) ** 2).round(
39          2
40      ) # Wald = (Coef / SE)^2
41      p_values = model.pvalues
42
43      significance_stars = p_values.apply(
44          lambda p: "***" if p < 0.001 else "**" if p < 0.01 else "*" if p < 0.05
                else ""
45      )
46
47      y_pred = model.predict(X)
48      rmse = np.sqrt(mean_squared_error(y, y_pred))
49
50      # Create the final dataframe with the required columns
51      final_df = pd.DataFrame(
52          {
53              "Predictor variable": X.columns,
54              "Beta": coefficients.round(2),
55              "SE": standard_errors.round(2),
56              "Wald": wald_stats,
57              "95% CI": [
58                  f"{ci[0]:.2f} - {ci[1]:.2f}{star}"
59                  for ci, star in zip(conf_int.values, significance_stars)
60              ],
61              "p-value": p_values.apply(lambda x: f"{x:.4f}"),
62          }
63      )
64
65      output_csv(
66          final_df,
67          f"{OUTPUT_FOLDER}/{MACHINE_LEARNING_FOLDER}/linear_regression_results",
68      )
69      output_latex(
70          final_df,
71          f"{OUTPUT_FOLDER}/{MACHINE_LEARNING_FOLDER}/linear_regression_results",
72      )
73
74      plt.barh(final_df["Predictor variable"], final_df["Beta"], color="blue")
75      plt.title("Linear Regression Coefficients")
76      plt.xlabel("Coefficient Value")
77      plt.show()
78
```

```
79     print(f"R-squared: {model.rsquared:.4f}")
80     print(f"RMSE: {rmse:.4f}")
81
82
83 if __name__ == "__main__":
84     main()
```

Listing A.16: machine_learning/optuna_lightgbm.py

```
 1 import optuna.integration.lightgbm as lgb_optuna
 2 import lightgbm as lgb
 3 import pandas as pd
 4 import numpy as np
 5 from sklearn.model_selection import StratifiedKFold, KFold
 6 from sklearn.preprocessing import StandardScaler
 7 from constants.constants import MACHINE_LEARNING_FOLDER, OUTPUT_FOLDER
 8 from lib.prepare_data import prepare_data
 9 from sklearn.metrics import (
10     balanced_accuracy_score,
11     mean_absolute_error,
12     r2_score,
13     roc_auc_score,
14 )
15
16 N_OUTER = 10
17 N_INNER = 10
18 RANDOM_STATE = 42
19
20
21 def nested_cv_optuna_lightgbm(X: pd.DataFrame, y: pd.Series, is_classifier:
       bool):
22     if is_classifier:
23         outer_cv = StratifiedKFold(
24             n_splits=N_OUTER, shuffle=True, random_state=RANDOM_STATE
25         )
26     else:
27         outer_cv = KFold(n_splits=N_OUTER, shuffle=True, random_state=
               RANDOM_STATE)
28
29     scores = []
30     r2_scores = []
31     mae_scores = []
32     aucs = []
33     best_params = []
34
35     output_file: str = (
36         f"{OUTPUT_FOLDER}/{MACHINE_LEARNING_FOLDER}/optuna_lightgbm_{'classifier
               ' if is_classifier else 'regressor'}_results.txt"
37     )
38
39     with open(output_file, "w") as f:
40         f.write(
```

```
41          f"Optuna LightGBM Nested CV ({'Classifier' if is_classifier else '
                Regressor'})\n\n"
42      )
43
44  for fold, (train_idx, test_idx) in enumerate(outer_cv.split(X, y)):
45      X_train, X_test = X.iloc[train_idx], X.iloc[test_idx]
46      y_train, y_test = y.iloc[train_idx], y.iloc[test_idx]
47
48      scaler = StandardScaler()
49      X_train_scaled = scaler.fit_transform(X_train)
50      X_test_scaled = scaler.transform(X_test)
51
52      dtrain = lgb.Dataset(X_train_scaled, label=y_train)
53      dtest = lgb.Dataset(X_test_scaled, label=y_test)
54
55      params = {
56          "objective": "binary" if is_classifier else "regression",
57          "metric": "binary_logloss" if is_classifier else "rmse",
58          "verbosity": -1,
59          "boosting_type": "gbdt",
60          "random_state": RANDOM_STATE,
61          "early_stopping_rounds": 100,
62      }
63
64      tuner = lgb_optuna.train(
65          params,
66          dtrain,
67          valid_sets=[dtest],
68          optuna_seed=RANDOM_STATE,
69      )
70
71      best_param = tuner.params
72      best_params.append(best_param)
73
74      y_pred = tuner.predict(X_test_scaled)
75      if is_classifier:
76          y_pred_label = (y_pred >= 0.5).astype(int)
77          balanced_acc = balanced_accuracy_score(y_test, y_pred_label)
78          auc = roc_auc_score(y_test, y_pred)
79          test_score = balanced_acc
80          aucs.append(auc)
81          with open(output_file, "a") as f:
82              f.write(
83                  f"Fold {fold+1}: Balanced Accuracy = {balanced_acc:.4f}, AUC
                        = {auc:.4f}, Best Params = {best_param}\n"
84              )
85      else:
86          test_score = -np.sqrt(np.mean((y_test - y_pred) ** 2)) # negative
                RMSE
87          r2 = r2_score(y_test, y_pred)
88          mae = mean_absolute_error(y_test, y_pred)
89          r2_scores.append(r2)
```

```
90              mae_scores.append(mae)
91              with open(output_file, "a") as f:
92                  f.write(
93                      f"Fold {fold+1}: neg_RMSE = {test_score:.4f}, R^2 = {r2:.4f},
                            MAE = {mae:.4f}, Best Params = {best_param}\n"
94                  )
95          scores.append(test_score)
96
97      mean = np.mean(scores)
98      std = np.std(scores)
99      with open(output_file, "a") as f:
100         if is_classifier:
101             mean_auc = np.mean(aucs)
102             std_auc = np.std(aucs)
103             f.write(f"\nNested CV Mean Balanced Accuracy: {mean:.4f} ±{std:.4f}\
                    n")
104             f.write(f"Nested CV Mean AUC: {mean_auc:.4f} ±{std_auc:.4f}\n")
105             return scores, aucs, best_params, mean, mean_auc
106         else:
107             mean_r2 = np.mean(r2_scores)
108             std_r2 = np.std(r2_scores)
109             mean_mae = np.mean(mae_scores)
110             std_mae = np.std(mae_scores)
111             f.write(f"\nNested CV Mean neg_RMSE: {mean:.4f} ±{std:.4f}\n")
112             f.write(f"Nested CV Mean R^2: {mean_r2:.4f} ±{std_r2:.4f}\n")
113             f.write(f"Nested CV Mean MAE: {mean_mae:.4f} ±{std_mae:.4f}\n")
114             f.write(f"Best parameters (per fold):\n{best_params}\n\n")
115             return scores, r2_scores, mae_scores, best_params, mean, mean_r2,
                    mean_mae
116
117
118 X_class, y_class = prepare_data(classification=True)
119 nested_cv_optuna_lightgbm(X_class, y_class, is_classifier=True)
120 X_reg, y_reg = prepare_data(classification=False)
121 nested_cv_optuna_lightgbm(X_reg, y_reg, is_classifier=False)
```