Modeling For Alzheimer's Diagnosis: Predictive Analysis*

STA314 Group28

Yuguang Duan 1007824348 Pingxuan Ren 1007613708
Yiyi Feng 1007763984 Nida Li 1008507411

December 8, 2024

Our team (Group 28) achieved a prediction score of 0.91666 on Kaggle for this competition, ranking 77th.

Table of contents

*Code and data are available at: https://github.com/william03duan/Alzheimer-Disease-Prediction.git

1 Introduction

Alzheimer's disease is a complex condition with no known definitive cause or cure. Its impact on individuals and families is profound, affecting cognitive function and quality of life. As the global population ages, Alzheimer's prevalence is expected to rise, emphasizing the need for early diagnosis and research. Identifying factors contributing to the onset of this disease is crucial for developing effective interventions. This study aims to predict Alzheimer's diagnosis using demographic, medical, and lifestyle factors, and to build a predictive model for early detection.

Our analysis focuses on predicting Alzheimer's diagnosis based on variables such as age, gender, medical history (e.g., hypertension, diabetes), cognitive performance, and lifestyle factors (e.g., smoking, alcohol consumption). We employed a Random Forest model to estimate diagnosis outcomes, incorporating predictors like age, ethnicity, cardiovascular health, memory complaints, and physical activity levels. The goal is to identify which factors most significantly influence the risk of Alzheimer's.

The remainder of this paper is structured as follows: Section 2 describe the research questions addressed in our study. Section 3 provides an overview of the dataset, including data processing and data analysis. Section 4 explains the modeling approach, and the algorithms chosen in this study to assess the model. Section 5 presents the findings, including conclusions drawn for our research questions. In Section 6, we discuss the limitations of our analysis, and what we can do next to improve our model. Codes are placed in the appendix.

2 Problem statement

This study, while constructing a predictive model for Alzheimer's diagnosis, also investigates two research questions:

- Among the 32 predictors included in the model, covering patients' demographic details, lifestyle factors, medical history, clinical measurements, cognitive and functional assessments, and symptoms, which predictors are most critical for accurate diagnosis?
- Within lifestyle factors, including BMI, smoking, alcohol consumption, physical activity, diet quality, and sleep quality, which factors have a statistically significant influence on Alzheimer's diagnosis?

Answering these questions will deepen our understanding of the key contributors to Alzheimer's diagnosis and provide guidance for patients to adopt healthier lifestyles, potentially reducing the risk of developing the disease.

3 Statistical Analysis

3.1 Data Processing

The data preprocessing step mainly focused on data cleaning and transformations. Firstly, the data cleaning procedures include handling missing values (eventually, there are no missing values for all columns), addressing outliers for numerical variables, and checking data types. The column "DoctorInCharge" was dropped since it was irrelevant to our target and research questions. Moreover, according to the correlation heatmap (which will be shown below), for all independent variables, none of the absolute values of the correlation coefficients except the main diagonal reached 0.7 or more, which indicated that none of the two variables were strongly correlated, so all the variables were used for prediction and modeling. For data transformations, standardization or normalization is used for k-nearest-neighbor, logistic regression, and support vector machine classifiers because they were sensitive to distances, and as the magnitudes of data became larger, they significantly contributed to distance calculation which caused incorrect classifications.

3.2 Data Analyzation

To gain a comprehensive understanding of the dataset before modeling, we created various visualizations to quickly grasp its structure. Five types of charts were produced:

- Boxplots and Histograms for quantitative data, illustrating their distributions and highlighting outliers or patterns.
- Pie Charts and Bar Charts for categorical data, providing insights into the proportions and frequencies of each category.
- A Correlation Heatmap for all predictors and diagnose data, showing the relationships and dependencies between variables.

Boxplots are available at here. The boxplot analysis was conducted for each variable. The analysis of 15 variables revealed no outliers.

Histograms are available at here. The histograms highlight distinct patterns in the dataset. Participant ages show a sharp peak at 90, indicating a larger proportion of older individuals. Triglyceride levels cluster around 300, suggesting elevated levels in a significant number of participants. Sleep quality scores are concentrated near 8, reflecting generally good sleep among participants, with fewer reporting poor sleep. MMSE scores exhibit notable variability, with a dip between 10 and 20, pointing to differences in cognitive abilities within the sample. These trends provide meaningful insights and warrant further exploration.

Piecharts are available at here. From the perspective of patient's demographic details, the gender distribution is nearly balanced, with Caucasians significantly outnumbering other ethnic groups, and the majority of patients having a High School education level. From a health standpoint, most individuals are non-smokers and have no history of Alzheimer's in their family, cardiovascular disease, diabetes, depression, head injuries, or hypertension. Similarly, the majority do not report symptoms such as memory impairment, confusion, disorientation, personality changes, difficulty completing tasks, or forgetfulness.

Barcharts are available at here. The bar plot reveals the number of Alzheimer's diagnoses across different categories for each factor. For most factors, there are no significant differences in diagnosis rates among categories. For instance, in the gender factor, males and females have similar diagnosis rates. However, notable differences are observed in certain cases. Caucasians have a slightly higher diagnosis rate compared to other ethnic groups. Among health-related factors, patients with hypertension, memory complaints, behavioral problems, and forgetfulness exhibit significantly higher probabilities of an Alzheimer's diagnosis. Interestingly, patients with a family history of Alzheimer's do not show an elevated diagnosis rate, suggesting that the disease may not have a strong genetic predisposition.

The correlation heatmap is available at here. The correlation matrix shows most variables have weak correlations with diagnosis (between -0.2 and 0.2). However, MMSE, Functional Assessment, and ADL show stronger negative correlations (below -0.2), indicating higher scores are linked to a lower likelihood of Alzheimer's. In contrast, memory complaints and behavioral problems have stronger positive correlations (above 0.2), suggesting that more severe symptoms increase the likelihood of a diagnosis. These findings align with the bar chart analysis, highlighting these variables' predictive value.

4 Model

4.1 Model Justification

In our study, we ultimately decided to use the Random Forest algorithm to predict Alzheimer's disease diagnosis.

During the model training process, we explored many other machine learning algorithms, including k-Nearest Neighbors (kNN), LDA, logistic regression, support vector machine, XGBoost, decision tree, and naive Bayes. However, compared to Random Forest, all models, except for the decision tree and XGBoost models, had significantly lower predictive accuracy. K-NN achieved 0.76, Logistic Regression and LDA reached 0.84, and Support Vector Machine scored 0.82 ('rbf' kernel) and 0.83 ('linear' kernel). Naïve Bayes attained 0.79, while Random Forest got an accuracy of 0.9459. Although the decision tree model achieved an extremely high accuracy of 1.0, it carried a substantial risk of overfitting. Meanwhile, the accuracy of XGBoost was similar to that of the Random Forest model, but when the features in the dataset don't interact much or are weakly correlated, XGBoost may overfit the data and fail to generalize well. In our previous data analysis, the correlation matrix indeed indicated weak associations between the features, so we ultimately chose Random Forest as the predictive model.

We believe that Random Forest performed exceptionally well in our study because of its ability to efficiently handle weakly correlated features and complex variable interactions, which align perfectly with the characteristics of our dataset. By constructing multiple decision trees and aggregating their predictions, Random Forest reduces the risk of overfitting while capturing diverse patterns within the data. Its robustness in managing multicollinearity and irrelevant features enables it to model non-linear relationships and interactions, which is especially critical when the associations between variables are limited. Moreover, Random Forest's ability to rank feature importance aligns with our research objective of identifying key predictive factors for Alzheimer's diagnosis. By striking a balance between predictive accuracy and interpretability, Random Forest emerged as the optimal choice for this study.

4.2 Model Evaluation

After deciding to use the random forest model for prediction, we employed 5-fold cross-validation to tune the mtry parameter, aiming to achieve more accurate predictions for Alzheimer's disease diagnosis.

We constructed random forest models with mtry values of 2, 5, 10, and 13, systematically evaluating their performance to identify the model with the highest predictive accuracy. We found that the random forest model with a single tree exhibited lower complexity but correspondingly lower accuracy. Conversely, models with two or more trees not only achieved perfect F1 scores and RMSE values but also demonstrated 100% accuracy in the confusion matrix, indicating superior overall performance.

The 5-fold cross-validation ultimately selected the random forest model with mtry = 13 as the best-performing model. Given its exceptional performance in the confusion matrix, F1 score, and RMSE values, as shown in Table 1, we finalized our decision to submit the random forest model with 13 trees to Kaggle.

Table 1: Model Evaluation Results

Metric	Value
Accuracy	1.00
Sensitivity	1.00
Specificity	1.00
Precision	1.00
F1 Score	1.00
RMSE	0.00

5 Result and Conclusion

Table 2: Feature Importance for Alzheimer's Diagnosis Model

	Mean Decrease Accuracy	Mean Decrease Gini
PatientID	0.7556	9.6217
Age	-0.0878	8.7842
Gender	-1.8839	1.0174
Ethnicity	-0.9146	2.6374
EducationLevel	-2.1322	3.0536
BMI	0.1764	11.4781
Smoking	-0.7749	0.9211
AlcoholConsumption	-2.2840	10.4492
PhysicalActivity	-0.8622	12.0139
DietQuality	-0.3158	13.7674
SleepQuality	-1.6468	11.8600
FamilyHistoryAlzheimers	-0.1988	1.2512
CardiovascularDisease	-1.9009	0.9469
Diabetes	-0.7044	0.9571
Depression	-1.2811	0.9458
HeadInjury	-1.5377	0.8626
Hypertension	1.8473	1.1975
SystolicBP	-1.2411	8.4176
DiastolicBP	-4.7367	7.6275
CholesterolTotal	-0.3004	10.2351
CholesterolLDL	-1.3531	10.2998
CholesterolHDL	-2.4868	10.2363
CholesterolTriglycerides	0.5814	13.4919
MMSE	112.7732	116.1214
FunctionalAssessment	151.6661	149.4205

	Mean Decrease Accuracy	Mean Decrease Gini
MemoryComplaints	97.3370	75.0427
BehavioralProblems	79.6727	53.5725
ADL	125.3133	135.9569
Confusion	0.5892	0.8026
Disorientation	-2.9606	0.9409
PersonalityChanges	-0.5386	1.0491
DifficultyCompletingTasks	-0.6055	0.7010
Forgetfulness	-0.2538	1.1953
DoctorInCharge	0.0000	0.0000

Table 2 presents the feature importance scores for two metrics, Mean Decrease Accuracy, and Mean Decrease Gini, derived from a random forest model predicting Alzheimer's diagnosis. These scores provide insight into how each feature contributes to the model's predictive performance, which answers our first research question.

Mean Decrease Accuracy measures how much the accuracy of the model decreases when the feature is excluded from the model. A higher value indicates that the feature is more critical to accurate predictions. MMSE (Mini-Mental State Examination), Functional Assessment, Memory Complaints, and Behavioral Problems all show very high values, suggesting these features significantly impact the model's accuracy.

Mean Decrease Gini assesses how much a feature improves the homogeneity of the nodes and leaves in the random forest. A higher value indicates that the feature plays a larger role in reducing impurity and improving the model's decisions. Features like MMSE, Functional Assessment, Memory Complaints, and Behavioral Problems also show strong values here, which further suggests that these variables have strong discriminatory power for classifying Alzheimer's diagnoses.

The impact of Functional Assessment, Memory Complaints, and Behavioral Problems on Alzheimer's disease has been confirmed by numerous studies (Amariglio et al. 2012, Jessen et al. 2014, Pinyopornpanish et al. 2022). However, we have not found any research that establishes a link between MMSE and Alzheimer's disease. Perhaps this could serve as a new perspective for future research into the causes of Alzheimer's disease.

Table 3: Summary of GLM Model Coefficients

	Variable	Estimate	Std. Error	z value	p-value
(Intercept)	(Intercept)	-0.3641	0.3397	-1.0719	0.2838
BMI	BMI	0.0111	0.0075	1.4762	0.1399
Smoking	Smoking	-0.0235	0.1203	-0.1952	0.8452
AlcoholConsumption	AlcoholConsumption	-0.0067	0.0094	-0.7070	0.4795

	Variable	Estimate	Std. Error	z value	p-value
PhysicalActivity	PhysicalActivity	-0.0007	0.0187	-0.0391	0.9688
DietQuality	DietQuality	0.0152	0.0185	0.8209	0.4117
SleepQuality	SleepQuality	-0.0777	0.0309	-2.5102	0.0121

To address the second research question, we conducted a Generalized Linear Model (GLM) analysis on six lifestyle factors: BMI, smoking, alcohol consumption, physical activity, diet quality, and sleep quality. The results of the analysis are presented in Table 3.

Firstly, the null deviance of 1954.4 is slightly larger than the residual deviance of 1944.5, indicating that the health factors explain a small portion of the variability in the training dataset. Secondly, according to our GLM model, the only statistically significant predictor is sleep quality, with a p-value of 0.0121. This means that among the six health-related factors, only sleep quality has a statistically significant impact on the diagnosis of Alzheimer's disease.

The odds ratio is:

$$\text{Odds Ratio} = \mathrm{e}^{\beta_{\text{SleepQuality}}} = 0.925$$

and this means that for every one-unit increase in the sleep quality among the subjects, the odds of being diagnosed with Alzheimer is multiplied by 0.925, or decrease by:

$$\%\text{Change} = 1 \text{ - Odds Ratio} = 1 \text{ - } 0.925 = 7.5\%$$

To validate our findings, we reviewed studies available online regarding the relationship between Alzheimer's disease and sleep quality. We found that the association between sleep quality and both the onset and pathological progression of Alzheimer's disease has been well-established in numerous studies. Research suggests that the underlying mechanism involves sleep disturbances accelerating the accumulation of beta-amyloid and tau proteins, thereby promoting the development and worsening of Alzheimer's disease. A bidirectional relationship has been identified between these factors (Neumann et al. 2017). Moreover, sleep fragmentation and insomnia have been confirmed as risk factors for Alzheimer's disease in many studies (Zhang et al. 2022). Based on our data analysis and supporting evidence from the literature, we conclude that sleep quality is the most critical health factor influencing Alzheimer's disease diagnosis.

6 Discussion

6.1 Limitation

In this study, while we are achieving certain advancements in predicting Alzheimer's disease diagnoses, there are several limitations. For instance, the dataset exhibits biases, with significant disparities in the proportions of different racial and educational groups. This imbalance reduces the generalizability of the study, making it difficult to apply the findings to a broader population. Moreover, the dataset suffers from a limited sample size and a narrow age range, which undermines the precision of the research. The data predominantly represent specific subpopulations, such as the elderly, which may increase the difficulty of replicating the study in future experiments. The dataset lacks demographic information, such as country of origin, preventing us from drawing any region- or country-specific conclusions.

Although various machine learning algorithms were evaluated, and the Random Forest model was ultimately selected, it is still possible that other models, that we did not test on, might better suit this dataset. Furthermore, as shown in Table 2, a small number of variables dominated the feature importance metrics, potentially leading to the underestimation of other predictive factors or misjudging their influence on Alzheimer's disease diagnosis. These limitations highlight areas for improvement in data collection, model selection, and feature analysis for future research.

6.2 Future Study

Future research could significantly benefit from enhancing the diversity and comprehensiveness of the dataset. While the dataset currently includes valuable demographic information such as age, gender, ethnicity, and education level, expanding the range of variables could provide deeper insights into the various factors influencing Alzheimer's Disease. For example, environmental factors, such as the patient's working environment, exposure to noise, pollutants, or other occupational hazards, could play a role in the development and progression of the disease. Understanding how these factors interact with genetic and lifestyle factors could reveal new avenues for prevention or treatment.

Furthermore, the inclusion of clinical data such as blood biomarkers or imaging results, like MRI or PET scans, would add an extra layer of depth to the analysis. These objective measures could help pinpoint specific physiological changes related to Alzheimer's, providing crucial information on the underlying mechanisms of the disease. Adding such data would not only increase the accuracy of predictions but also enhance our understanding of how these biomarkers relate to cognitive decline and other clinical symptoms.

To mitigate the impact of errors arising from an excessive number of variables, dimensionality reduction techniques such as Principal Component Analysis (PCA) should be employed. Retaining only components that explain 95%-99% of the variance would ensure a more concise and robust feature set. Additionally, future studies could investigate the interactions between variables and explore the use of more advanced models or hybrid approaches to further refine predictions and improve model performance, for instance, LightGBM or CatBoost.

Appendix a. Code

All code are available at: https://github.com/william03duan/Alzheimer-Disease-Prediction.git.

6.1 Decision Tree Model (Python)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier

df_train = pd.read_csv('train.csv')
df_train.head()

df_test = pd.read_csv('test.csv')
df_test.head()

df_train.drop(['DoctorInCharge'], axis = 1, inplace = True)

df_test.drop(['DoctorInCharge'], axis = 1, inplace = True)

x_train = df_train.drop(['PatientID'], axis = 1)
y_train = df_train['Diagnosis']
x_test = df_test.drop(['PatientID'], axis = 1)

clf = DecisionTreeClassifier(criterion='gini',
                             splitter='best', random_state = 42)

clf.fit(x_train, y_train)

val_scores = cross_val_score(clf, x_train, y_train, cv = 10)
print(f"Cross-validation scores :{val_scores}")
print(f"Mean accuracy : {val_scores.mean():.2f}")
```

6.2 KNN Model (Python)

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import cross_val_score

df_train = pd.read_csv('train.csv')
df_train.head()

df_train.drop(['DoctorInCharge'],axis = 1, inplace = True)
df_train.head()

numerical_column = []
for col in df_train.drop(['PatientID'], axis = 1).columns:
    if df_train[col].nunique() > 4:
        numerical_column.append(col)
print(f"Numerical columns:", numerical_column)

scaler = StandardScaler()

df_train[numerical_column]=
  scaler.fit_transform(df_train[numerical_column])
df_train

x_train = df_train.drop(['PatientID','Diagnosis'], axis = 1)
y_train = df_train['Diagnosis']

df_test = pd.read_csv('test.csv')
df_test.head()

df_test.drop(['DoctorInCharge'], axis = 1, inplace = True)
df_test

df_test[numerical_column]=
  scaler.fit_transform(df_test[numerical_column])
df_test.head()
```

```
x_test = df_test.drop(['PatientID'], axis = 1)

knn = KNeighborsClassifier()

mean_scores_1 = []
k_value = range(1,40)
for k in k_value:
    knn.n_neighbors = k
    scores = cross_val_score(knn, x_train, y_train,
                             cv = 5, scoring = 'accuracy')
    mean_scores_1.append(scores.mean())
optimal_value_k = k_value[np.argmax(mean_scores_1)]
print(f"optimal k is {optimal_value_k}")

mean_scores_2 = []
k_value = range(1,40)
for k in k_value:
    knn.n_neighbors = k
    scores = cross_val_score(knn, x_train, y_train,
                             cv = 10, scoring = 'accuracy')
    mean_scores_2.append(scores.mean())
optimal_value_k = k_value[np.argmax(mean_scores_2)]
print(f"optimal k is {optimal_value_k}")

mean_scores_3 = []
k_value = range(1,40)
for k in k_value:
    knn.n_neighbors = k
    scores = cross_val_score(knn, x_train, y_train,
                             cv = 15, scoring = 'accuracy')
    mean_scores_3.append(scores.mean())
optimal_value_k = k_value[np.argmax(mean_scores_3)]
print(f"optimal k is {optimal_value_k}")

mean_scores = []
k_value = range(1,40)
for k in k_value:
    knn.n_neighbors = k
    scores = cross_val_score(knn, x_train, y_train,
                             cv = 3, scoring = 'accuracy')
    mean_scores.append(scores.mean())
```

```
optimal_value_k = k_value[np.argmax(mean_scores)]
print(f"optimal k is {optimal_value_k}")

knn_1 = KNeighborsClassifier(n_neighbors = 29)

knn_1.fit(x_train,y_train)

val_scores_3 = cross_val_score(knn_1, x_train, y_train, cv = 10)
print(f"Cross-validation scores_3 :{val_scores_3}")
print(f"Mean accuracy : {val_scores_3.mean():.2f}")

pred = knn_1.predict(x_test)

df_test['Diagnosis'] = pd.Series(pred)
df_test
```

6.3 LDA Model (Python)

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import cross_val_score
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis,
QuadraticDiscriminantAnalysis
from sklearn.preprocessing import StandardScaler

df_train = pd.read_csv('train.csv')
df_train.head()

df_train.drop(['DoctorInCharge'],axis = 1, inplace = True)
df_train.head()

df_test = pd.read_csv('test.csv')
df_test.head()

df_test.drop(['DoctorInCharge'], axis = 1, inplace = True)

numerical_column = []
for col in df_train.drop(['PatientID'], axis = 1).columns:
    if df_train[col].nunique() > 4:
        numerical_column.append(col)
print(f"Numerical columns:", numerical_column)

scaler = StandardScaler()
df_train[numerical_column]=scaler.fit_transform(df_train[numerical_column])
df_train

scaler = StandardScaler()
df_test[numerical_column]=scaler.fit_transform(df_test[numerical_column])
df_test

x_train = df_train.drop(['PatientID', 'Diagnosis'], axis = 1)
y_train = df_train['Diagnosis']

x_test = df_test.drop(['PatientID'], axis = 1)

LDA_classification = LinearDiscriminantAnalysis()

LDA_classification.fit(x_train, y_train)

val_scores = cross_val_score(LDA_classification, x_train, y_train, cv = 10)
print(f"Cross-validation scores_1 :{val_scores}")
print(f"Mean accuracy : {val_scores.mean():.2f}")

LDA_classification.predict(x_test)
```

6.4 Logistic Regression Model (Python)

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import MinMaxScaler

df_train = pd.read_csv('train.csv')
df_train.head()
df_train.drop(['DoctorInCharge'],axis = 1, inplace = True)
df_train.head()

numerical_column = []
for col in df_train.drop(['PatientID'], axis = 1).columns:
    if df_train[col].nunique() > 4:
        numerical_column.append(col)
print(f"Numerical columns:", numerical_column)

# scaler = StandardScaler()
# df_train[numerical_column]=scaler.fit_transform(df_train[numerical_column])
# df_train
min_max_scaler = MinMaxScaler()
df_train[numerical_column] =
  min_max_scaler.fit_transform(df_train[numerical_column])
df_train

x_train = df_train.drop(['PatientID','Diagnosis'], axis = 1)
y_train = df_train['Diagnosis']

df_test = pd.read_csv('test.csv')
df_test

df_test.drop(['DoctorInCharge'], axis = 1, inplace = True)
df_test

# df_test[numerical_column]=scaler.fit_transform(df_test[numerical_column])
df_test[numerical_column] =
  min_max_scaler.fit_transform(df_test[numerical_column])
df_test

x_test = df_test.drop(['PatientID'],axis = 1)

model = LogisticRegression(random_state=50, max_iter=1000)
model.fit(x_train, y_train)

val_scores = cross_val_score(model, x_train, y_train, cv = 10)
print(f"Cross-validation scores_1 :{val_scores}")
print(f"Mean accuracy : {val_scores.mean():.2f}")

pred = model.predict(x_test)
```

```
pred

count = np.count_nonzero(pred == 1)
count

model_1 = LogisticRegression(penalty='l1', solver='saga', max_iter=1000)

model_1.fit(x_train, y_train)

val_scores_1 = cross_val_score(model_1, x_train, y_train, cv = 10)
print(f"Cross-validation scores_1 :{val_scores}")
print(f"Mean accuracy : {val_scores.mean():.2f}")

y_pred = model_1.predict(x_test)
y_pred

count_1 = np.count_nonzero(y_pred == 1)
count_1
```

6.5 Naive Bayes Model (Python)

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import cross_val_score
from sklearn.naive_bayes import GaussianNB

df_train = pd.read_csv('train.csv')
df_train.head()

df_train.drop(['DoctorInCharge'],axis = 1, inplace = True)
df_train.head()

x_train = df_train.drop(['PatientID','Diagnosis'], axis = 1)
y_train = df_train['Diagnosis']

df_test = pd.read_csv('test.csv')
df_test.head()
df_test.drop(['DoctorInCharge'], axis = 1, inplace = True)
df_test

x_test = df_test.drop(['PatientID'], axis = 1)

NB_classifier = GaussianNB()
NB_classifier.fit(x_train, y_train)

val_scores_1 = cross_val_score(NB_classifier, x_train, y_train, cv = 10)
print(f"Cross-validation scores_1 :{val_scores_1}")
print(f"Mean accuracy : {val_scores_1.mean():.2f}")

pred = NB_classifier.predict(x_test)
pred

df_test['Diagnosis'] = pd.Series(pred)
df_test
```

6.6 Support Vector Machine (Python)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC

df_train = pd.read_csv('train.csv')
df_train.head()

df_train.drop(['DoctorInCharge'],axis = 1, inplace = True)

x_train = df_train.drop(['PatientID','Diagnosis'], axis = 1)
y_train = df_train['Diagnosis']

df_test = pd.read_csv('test.csv')
df_test.head()
```

```
x_test = df_test.drop(['PatientID'], axis = 1)

numerical_column = []
for col in df_train.drop(['PatientID'], axis = 1).columns:
    if df_train[col].nunique() > 4:
        numerical_column.append(col)
print(f"Numerical columns:", numerical_column)

scaler = StandardScaler()
x_train[numerical_column]=scaler.fit_transform(df_train[numerical_column])

x_test[numerical_column] = scaler.fit_transform(x_test[numerical_column])

svm_model_1 = SVC(kernel='rbf', C=1.0, gamma='scale', random_state=42)
svm_model_1.fit(x_train, y_train)

val_scores = cross_val_score(svm_model_1, x_train, y_train, cv = 10)
print(f"Cross-validation scores_1 :{val_scores}")
print(f"Mean accuracy : {val_scores.mean():.2f}")

svm_model_2 = SVC(kernel='linear', C=1.0, gamma='scale', random_state=42)
svm_model_2.fit(x_train, y_train)

val_scores = cross_val_score(svm_model_2, x_train, y_train, cv = 10)
print(f"Cross-validation scores_1 :{val_scores}")
print(f"Mean accuracy : {val_scores.mean():.2f}")
```

6.7 XGboost Model (Python)

```
! pip install xgboost

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import xgboost as xgb
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
from xgboost import XGBClassifier
```

```
df_train = pd.read_csv('train.csv')
df_train.head()

df_train.drop(['DoctorInCharge'],axis = 1, inplace = True)

x_train = df_train.drop(['PatientID','Diagnosis'], axis = 1)
y_train = df_train['Diagnosis']

df_test = pd.read_csv('test.csv')
df_test.head()

x_test = df_test.drop(['PatientID','DoctorInCharge'], axis = 1)

params = {'objective':'binary:logistic','max_depth': 4,'alpha': 10,
        'learning_rate': 1.0,'n_estimators':100}

xgb_clf = XGBClassifier(**params)

xgb_clf.fit(x_train, y_train)

val_scores = cross_val_score(xgb_clf, x_train, y_train, cv = 10)
print(f"Cross-validation scores_1 :{val_scores}")
print(f"Mean accuracy : {val_scores.mean():.2f}")

pred = xgb_clf.predict(x_test)
pred

df_test['Diagnosis'] = pd.Series(pred)
df_test

xgb.plot_importance(xgb_clf,max_num_features=20,
                    title='Feature Importances (Weight)', height=0.5)
```

6.8 Random Forest (Rcode)

```
train = read.csv("train.csv")
test = read.csv("test.csv")

# Data Cleaning
```

```
train <- subset(train, select = -DoctorInCharge)
test <- subset(test, select = -DoctorInCharge)

library(randomForest)
set.seed(0)
train$Diagnosis <- as.factor(train$Diagnosis)
# Fit Random Forest model
rf_model <- randomForest(Diagnosis ~ .,
                         data=train, mtry=1, importance=TRUE)
summary(rf_model)
# Predict on the test set
yhat_rf <- predict(rf_model, newdata=test)

# Compute feature importance
importance(rf_model)

# Plot feature importance
varImpPlot(rf_model)

#output the csv file submitted in Kaggle.com
output_df <- data.frame(
  PatientID = test$PatientID,
  Diagnosis = yhat_rf
)

write.csv(output_df, "output.csv", row.names = FALSE)q

# setup
# install.packages("MLmetrics")
# install.packages("Metrics")
# install.packages("caret")
library("MLmetrics")
library("caret")
library("Metrics")

# Model Evaluations (on training data only)
yhat_rf_train <- predict(rf_model, newdata = train)

# Confusion Matrix
conf_matrix <- confusionMatrix(yhat_rf_train, train$Diagnosis)
print(conf_matrix)
```

```
# F1 Score
f1 <- F1_Score(y_true = train$Diagnosis,
               y_pred = yhat_rf_train, positive = "1")
cat("F1 Score (Training):", f1, "\n")

# RMSE
rmse_value <- rmse(as.numeric(train$Diagnosis) - 1,
                   as.numeric(yhat_rf_train) - 1)
cat("RMSE (Training):", rmse_value, "\n")

# Result and analysis

# Outputs a csv file to compare rf(1 tree) and rf(two and more trees)
rf_comparison <- data.frame(
  "Number of Trees" = c("1", "2+"),
  "Accuracy" = c(0.8305, 1.0000),
  "95% CI" = c("(0.8105, 0.8491)", "(0.9976, 1)"),
  "Kappa" = c(0.584, 1.000),
  "Sensitivity" = c(1.0000, 1.0000),
  "Specificity" = c(0.5207, 1.0000),
  "Pos Pred Value" = c(0.7922, 1.0000),
  "Neg Pred Value" = c(1.0000, 1.0000),
  "F1 Score" = c(0.684796, 1.0000),
  "RMSE" = c(0.4117619, 0.0000)
)
write.csv(rf_comparison, "rf_comparison.csv", row.names = FALSE)
rf_comparison

# Set up
set.seed(0)
train_control <- trainControl(
  method = "cv",
  number = 5,
  search = "grid"
)


tune_grid <- expand.grid(
  mtry = c(2, 5, 10, 13)    # Number of trees
)
```

```
# Train the Random Forest model
rf_tuned <- train(
  Diagnosis ~ .,
  data = train,
  method = "rf",
  metric = "Accuracy",
  tuneGrid = tune_grid,
  trControl = train_control,
  ntree = 100
)

# Best model
print(rf_tuned$bestTune)

# Performance data
print(rf_tuned$results)

# Performance Plot
plot(rf_tuned)
```

6.9 Research Question Two Analyzation (Rcode)

```
# Load the libraries
library(dplyr)
library(ggplot2)

# Import train Dataset
train = read.csv("train.csv")

# Fit a Generalized Linear Model
model_one <- glm(Diagnosis ~ BMI + Smoking + AlcoholConsumption
                 + PhysicalActivity + DietQuality + SleepQuality,
                 data = train,
                 family = binomial(link = "logit"))

# GLM Model analysis
summary(model_one)

# Extract variables by significance level (0.05)
significant_vars <- summary(model_one)$coefficients %>%
  as.data.frame() %>%
  filter(`Pr(>|z|)` < 0.05)

# Print significant variables
print(significant_vars)
```

6.10 Exploratory Data Analysis (Python)

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px

df_train = pd.read_csv('train.csv')
df_train.head()

df_train.shape

df_train.tail()

df_train.isnull().sum()

df_train.info()

df_train.shape

df_train.drop(['PatientID', 'DoctorInCharge'], axis = 1, inplace = True)

numerical_columns = []
for col in df_train.columns:
    if df_train[col].nunique() > 4:
        numerical_columns.append(col)
print(f"Numerical columns:", numerical_columns)

print(len(numerical_columns))

df_train[numerical_columns].describe()

for col_1 in numerical_columns:
    plt.figure(figsize=(10, 5))
    sns.boxplot(data=df_train, x=col_1)
    plt.title(f'Boxplot of distribution of {col_1}')
    plt.show()

for col_3 in numerical_columns:
    plt.figure(figsize=(10,10))
    sns.histplot(df_train[col_3], bins=30, kde=True, edgecolor='black', alpha=0.7)
    plt.title(f'Histogram of distribution of {col_3}')
    plt.xlabel('Value', fontsize = 14)
    plt.ylabel('Frequency', fontsize = 14)
    plt.show()

categorical_columns = []
for col in df_train.columns:
    if df_train[col].nunique() <= 4:
        categorical_columns.append(col)
print(f"Categorical columns:", categorical_columns)

print(len(categorical_columns))

alternative_one = categorical_columns[:17]
print(alternative_one)

for col_2 in alternative_one:
    value_counts = df_train[col_2].value_counts()
    plt.figure(figsize = (8,8))
    value_counts.plot.pie(autopct = '%1.1f%%', startangle=90, colormap= 'Set3')
    plt.title(f'{col_2} value counts percentage')
    plt.show()

value_counts = df_train['Diagnosis'].value_counts()
value_counts.plot.pie(autopct = '%1.1f%%', startangle=90, colormap= 'Set3')
plt.title('Diagnosis value counts percentage')
plt.savefig('Diagnosis value counts distribution.png')
plt.show()

for col_5 in alternative_one:
    fig = px.histogram(df_train, x=col_5, color = 'Diagnosis',
                       title=f"Bar Chart counts of {col_5}")
    fig.show()
```

```
plt.figure(figsize = (28,28))
sns.heatmap(df_train.corr(),cmap="coolwarm", annot = True)
plt.title('Correlation heatmap between each variables', size = 25)
plt.show()
```

References

Amariglio, Rebecca E., J. Alex Becker, Jeremy Carmasin, Lauren P. Wadsworth, Natacha Lorius, Caroline Sullivan, Jacqueline E. Maye, et al. 2012. "Subjective Cognitive Complaints and Amyloid Burden in Cognitively Normal Older Individuals." *Neuropsychologia* 50 (12): 2880–86. https://doi.org/10.1016/j.neuropsychologia.2012.08.011.

Jessen, Frank, Rebecca E. Amariglio, Martin van Boxtel, Monique Breteler, Mathieu Ceccaldi, Gaël Chételat, Bruno Dubois, et al. 2014. "A Conceptual Framework for Research on Subjective Cognitive Decline in Preclinical Alzheimer's Disease." *Alzheimer's & Dementia*. https://doi.org/10.1016/j.jalz.2014.01.001.

Neumann, Adam R., Robrecht Raedt, Hendrik W. Steenland, Mathieu Sprengers, Katarzyna Bzymek, Zaneta Navratilova, Lilia Mesina, et al. 2017. "Involvement of Fast-Spiking Cells in Ictal Sequences During Spontaneous Seizures in Rats with Chronic Temporal Lobe Epilepsy." *Brain* 140: 2355–69. https://doi.org/10.1093/brain/awx179.

Pinyopornpanish, Kanokporn, Atiwat Soontornpun, Tinakon Wongpakaran, Nahathai Wongpakaran, Surat Tanprawate, Kanokwan Pinyopornpanish, Angkana Nadsasarn, and Manee Pinyopornpanish. 2022. "Impact of Behavioral and Psychological Symptoms of Alzheimer's Disease on Caregiver Outcomes." *Scientific Reports*. https://doi.org/10.1038/s41598-022-18470-8.

Zhang, Ye, Rong Ren, Linghui Yang, Haipeng Zhang, Yuan Shi, Hamid R. Okhravi, Michael V. Vitiello, Larry D. Sanford, and Xiangdong Tang. 2022. "Sleep in Alzheimer's Disease: A Systematic Review and Meta-Analysis of Polysomnographic Findings." *Translational Psychiatry*. https://doi.org/10.1038/s41398-022-01897-y.