The Superior University

Session: 2023-2027

Department of Software Engineering - Superior University

Course : Programming For AI

Course Instructor: Prof Ansar Naseem

Topic: Alzheimer's Prediction

4th Semester BSAI

Fall 2025

Group Members	
Name	**Roll_no**
Fatima Nadeem	076
Muhammad Hassan	101
Muhammad Faraz	091
Muhammad Fasih ud din	094

Introduction:

We have chosen the **Alzheimer's Prediction Dataset (Global)** from Kaggle to create a machine learning model that can help predict Alzheimer's disease. Detecting this disease early is very important because it allows doctors to start treatment sooner, which can slow its progress and improve the patient's life. This dataset includes important details like a person's age, medical history, and test results, making it useful for building a model that can identify people at risk.

Field of Study:

This project belongs to the field of **Healthcare and Artificial Intelligence (AI)**, specifically within **Medical Data Science**. It involves using **Machine Learning (ML) in Healthcare** to analyze patient data and develop an early detection system for Alzheimer's disease. The combination of AI and medical research is becoming increasingly important for improving diagnosis, treatment, and patient care.

Domain Understanding:

This project falls under the domain of **Medical Data Analysis and Predictive Healthcare**. Alzheimer's disease is a neurodegenerative disorder with no known cure, but early detection can help manage symptoms and improve treatment outcomes. The dataset contains valuable patient information, including age, medical history, and cognitive test results, which are critical in determining the risk of Alzheimer's. By understanding these factors and applying machine learning techniques, we can develop a model that supports healthcare professionals in making better diagnostic decisions, ultimately benefiting patients and caregivers.

Data set Link:

https://www.kaggle.com/datasets/ankushpanday1/alzheimers-prediction-dataset-global

Model Training Code:

1*st* Step:

Make the requirement file first and then download the Librarie.

```
import pandas as pd

import numpy as np

import pickle

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.model_selection import train_test_split

from sklearn.svm import SVC

from sklearn.metrics import accuracy_score

import plotly.express as px

from sklearn.ensemble import RandomForestClassifier

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score, classification_report

import joblib

from sklearn.metrics import roc_curve, auc

import matplotlib.pyplot as plt
```

2nd **Step:**

Read the data and analyze the information.df.head is to read the first 5 rows of the dataset.

df=pd.read_csv("alzheimers_prediction_dataset.csv")

df_head = df.head()

df_head

Loading data from csv dataset

```
[2]: df=pd.read_csv("alzheimers_prediction_dataset.csv")

[3]: df_head = df.head()
     df_head
```

[3]:

	Country	Age	Gender	Education Level	BMI	Physical Activity Level	Smoking Status	Alcohol Consumption	Diabetes	Hypertension	...	Dietary Habits	Air Pollution Exposure	Employment Status	Marital Status	Genetic Risk Factor (APOE-ε4 allele)
0	Spain	90	Male	1	33.0	Medium	Never	Occasionally	No	No	...	Healthy	High	Retired	Single	No
1	Argentina	72	Male	7	29.9	Medium	Former	Never	No	No	...	Healthy	Medium	Unemployed	Widowed	No
2	South Africa	86	Female	19	22.9	High	Current	Occasionally	No	Yes	...	Average	Medium	Employed	Single	No
3	China	53	Male	17	31.2	Low	Never	Regularly	Yes	No	...	Healthy	Medium	Retired	Single	No
4	Sweden	58	Female	3	30.0	High	Former	Never	Yes	No	...	Unhealthy	High	Employed	Married	No

5 rows × 25 columns

alyzing data

3*rd* Step:

Checking the information of data .

df_info = df.info()

df_info

[4]:
```
df_info = df.info()
df_info
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 74283 entries, 0 to 74282
Data columns (total 25 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Country                               74283 non-null  object
 1   Age                                   74283 non-null  int64
 2   Gender                                74283 non-null  object
 3   Education Level                       74283 non-null  int64
 4   BMI                                   74283 non-null  float64
 5   Physical Activity Level               74283 non-null  object
 6   Smoking Status                        74283 non-null  object
 7   Alcohol Consumption                   74283 non-null  object
 8   Diabetes                              74283 non-null  object
 9   Hypertension                          74283 non-null  object
 10  Cholesterol Level                     74283 non-null  object
 11  Family History of Alzheimer's         74283 non-null  object
 12  Cognitive Test Score                  74283 non-null  int64
 13  Depression Level                      74283 non-null  object
 14  Sleep Quality                         74283 non-null  object
 15  Dietary Habits                        74283 non-null  object
 16  Air Pollution Exposure                74283 non-null  object
 17  Employment Status                     74283 non-null  object
 18  Marital Status                        74283 non-null  object
 19  Genetic Risk Factor (APOE-ε4 allele)  74283 non-null  object
 20  Social Engagement Level               74283 non-null  object
 21  Income Level                          74283 non-null  object
 22  Stress Levels                         74283 non-null  object
 23  Urban vs Rural Living                 74283 non-null  object
 24  Alzheimer's Diagnosis                 74283 non-null  object
dtypes: float64(1), int64(3), object(21)
memory usage: 14.2+ MB
```

df.describe()

```
df.describe()
```

	Age	Education Level	BMI	Cognitive Test Score
count	74283.000000	74283.000000	74283.000000	74283.000000
mean	71.964703	9.487514	26.780639	64.654241
std	12.980748	5.757020	4.764679	20.153247
min	50.000000	0.000000	18.500000	30.000000
25%	61.000000	4.000000	22.700000	47.000000
50%	72.000000	9.000000	26.800000	65.000000
75%	83.000000	14.000000	30.900000	82.000000
max	94.000000	19.000000	35.000000	99.000000

Count the data present in each columns

df.count()

```
[6]: df.count()
```

```
[6]: Country                                74283
     Age                                    74283
     Gender                                 74283
     Education Level                        74283
     BMI                                    74283
     Physical Activity Level                74283
     Smoking Status                         74283
     Alcohol Consumption                    74283
     Diabetes                               74283
     Hypertension                           74283
     Cholesterol Level                      74283
     Family History of Alzheimer's          74283
     Cognitive Test Score                   74283
     Depression Level                       74283
     Sleep Quality                          74283
     Dietary Habits                         74283
     Air Pollution Exposure                 74283
     Employment Status                      74283
     Marital Status                         74283
     Genetic Risk Factor (APOE-ε4 allele)   74283
     Social Engagement Level                74283
     Income Level                           74283
     Stress Levels                          74283
     Urban vs Rural Living                  74283
     Alzheimer's Diagnosis                  74283
     dtype: int64
```

Checking the number of unique values in a column.

df.count()

```
df.nunique()
```

```
Country                                20
Age                                    45
Gender                                  2
Education Level                        20
BMI                                   166
Physical Activity Level                 3
Smoking Status                          3
Alcohol Consumption                     3
Diabetes                                2
Hypertension                            2
Cholesterol Level                       2
Family History of Alzheimer's          2
Cognitive Test Score                   70
Depression Level                        3
Sleep Quality                           3
Dietary Habits                          3
Air Pollution Exposure                  3
Employment Status                       3
Marital Status                          3
Genetic Risk Factor (APOE-ε4 allele)    2
Social Engagement Level                 3
Income Level                            3
Stress Levels                           3
Urban vs Rural Living                   2
Alzheimer's Diagnosis                   2
dtype: int64
```

Checking the null values a very important step for any data cleaning.

df.isnull().sum()

```
[8]: df.isnull().sum()
```

```
[8]: Country                                  0
     Age                                      0
     Gender                                   0
     Education Level                          0
     BMI                                      0
     Physical Activity Level                  0
     Smoking Status                           0
     Alcohol Consumption                      0
     Diabetes                                 0
     Hypertension                             0
     Cholesterol Level                        0
     Family History of Alzheimer's            0
     Cognitive Test Score                     0
     Depression Level                         0
     Sleep Quality                            0
     Dietary Habits                           0
     Air Pollution Exposure                   0
     Employment Status                        0
     Marital Status                           0
     Genetic Risk Factor (APOE-ε4 allele)     0
     Social Engagement Level                  0
     Income Level                             0
     Stress Levels                            0
     Urban vs Rural Living                    0
     Alzheimer's Diagnosis                    0
     dtype: int64
```

4*th* Step:

Data tranformation.

df['BMI'] = df['BMI'].astype('int64')

```
def dataEncoder(cols):
    for i in cols:
        dataLabelEncoder = LabelEncoder()
        df[i] = dataLabelEncoder.fit_transform(df[i])


columns = ['Country',
 'Age',
 'Gender',
 'Education Level',
 'BMI',
 'Physical Activity Level',
 'Smoking Status',
 'Alcohol Consumption',
 'Diabetes',
 'Hypertension',
 'Cholesterol Level',
 'Family History of Alzheimer's',
 'Cognitive Test Score',
 'Depression Level',
 'Sleep Quality',
 'Dietary Habits',
 'Air Pollution Exposure',
 'Employment Status',
 'Marital Status',
 'Genetic Risk Factor (APOE-ε4 allele)',
```

'Social Engagement Level',
'Income Level',
'Stress Levels',
'Urban vs Rural Living',
'Alzheimer’s Diagnosis']
dataEncoder(columns)

result of this code:
using de.info()

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 74283 entries, 0 to 74282
Data columns (total 25 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Country                               74283 non-null  int64
 1   Age                                   74283 non-null  int64
 2   Gender                                74283 non-null  int64
 3   Education Level                       74283 non-null  int64
 4   BMI                                   74283 non-null  int64
 5   Physical Activity Level               74283 non-null  int64
 6   Smoking Status                        74283 non-null  int64
 7   Alcohol Consumption                   74283 non-null  int64
 8   Diabetes                              74283 non-null  int64
 9   Hypertension                          74283 non-null  int64
 10  Cholesterol Level                     74283 non-null  int64
 11  Family History of Alzheimer's         74283 non-null  int64
 12  Cognitive Test Score                  74283 non-null  int64
 13  Depression Level                      74283 non-null  int64
 14  Sleep Quality                         74283 non-null  int64
 15  Dietary Habits                        74283 non-null  int64
 16  Air Pollution Exposure                74283 non-null  int64
 17  Employment Status                     74283 non-null  int64
 18  Marital Status                        74283 non-null  int64
 19  Genetic Risk Factor (APOE-ε4 allele)  74283 non-null  int64
 20  Social Engagement Level               74283 non-null  int64
 21  Income Level                          74283 non-null  int64
 22  Stress Levels                         74283 non-null  int64
 23  Urban vs Rural Living                 74283 non-null  int64
 24  Alzheimer's Diagnosis                 74283 non-null  int64
dtypes: int64(25)
memory usage: 14.2 MB
```

5*th* Step:

identify the outlier in the dataset. with iqr method.

[16]:
```
Q1=df.quantile(0.25)
Q1
```

[16]:
```
Country                                  4.0
Age                                     11.0
Gender                                   0.0
Education Level                          4.0
BMI                                      4.0
Physical Activity Level                  0.0
Smoking Status                           0.0
Alcohol Consumption                      0.0
Diabetes                                 0.0
Hypertension                             0.0
Cholesterol Level                        0.0
Family History of Alzheimer's           0.0
Cognitive Test Score                    17.0
Depression Level                         0.0
Sleep Quality                            0.0
Dietary Habits                           0.0
Air Pollution Exposure                   0.0
Employment Status                        0.0
Marital Status                           0.0
Genetic Risk Factor (APOE-ε4 allele)     0.0
Social Engagement Level                  0.0
Income Level                             0.0
Stress Levels                            0.0
Urban vs Rural Living                    0.0
Alzheimer's Diagnosis                    0.0
Name: 0.25, dtype: float64
```

7]:
```
Q3=df.quantile(0.75)
Q3
```

7]:
```
Country                                  14.0
Age                                      33.0
Gender                                    1.0
Education Level                          14.0
BMI                                      12.0
Physical Activity Level                   2.0
Smoking Status                            2.0
Alcohol Consumption                       2.0
Diabetes                                  0.0
Hypertension                              1.0
Cholesterol Level                         1.0
Family History of Alzheimer's            1.0
Cognitive Test Score                     52.0
Depression Level                          2.0
Sleep Quality                             2.0
Dietary Habits                            2.0
Air Pollution Exposure                    2.0
Employment Status                         2.0
Marital Status                            2.0
Genetic Risk Factor (APOE-ε4 allele)      0.0
Social Engagement Level                   2.0
Income Level                              2.0
Stress Levels                             2.0
Urban vs Rural Living                     1.0
Alzheimer's Diagnosis                     1.0
Name: 0.75, dtype: float64
```

[18]:
```
IQR=Q3-Q1
IQR
```

[18]:
```
Country                                 10.0
Age                                     22.0
Gender                                   1.0
Education Level                         10.0
BMI                                      8.0
Physical Activity Level                  2.0
Smoking Status                           2.0
Alcohol Consumption                      2.0
Diabetes                                 0.0
Hypertension                             1.0
Cholesterol Level                        1.0
Family History of Alzheimer's           1.0
Cognitive Test Score                    35.0
Depression Level                         2.0
Sleep Quality                            2.0
Dietary Habits                           2.0
Air Pollution Exposure                   2.0
Employment Status                        2.0
Marital Status                           2.0
Genetic Risk Factor (APOE-ε4 allele)     0.0
Social Engagement Level                  2.0
Income Level                             2.0
Stress Levels                            2.0
Urban vs Rural Living                    1.0
Alzheimer's Diagnosis                    1.0
dtype: float64
```

[19]:
```
upper_lim = Q3 + 1.5 * IQR
lower_lim = Q1 - 1.5 * IQR
```

[20]:
```
outliers = ((df < lower_lim) | (df > upper_lim)).sum()
outliers
```

```
outliers = ((df < lower_lim) | (df > upper_lim)).sum()
outliers
```

```
Country                                0
Age                                    0
Gender                                 0
Education Level                        0
BMI                                    0
Physical Activity Level                0
Smoking Status                         0
Alcohol Consumption                    0
Diabetes                           14756
Hypertension                           0
Cholesterol Level                      0
Family History of Alzheimer's         0
Cognitive Test Score                   0
Depression Level                       0
Sleep Quality                          0
Dietary Habits                         0
Air Pollution Exposure                 0
Employment Status                      0
Marital Status                         0
Genetic Risk Factor (APOE-ε4 allele)   14722
Social Engagement Level                0
Income Level                           0
Stress Levels                          0
Urban vs Rural Living                  0
Alzheimer's Diagnosis                  0
dtype: int64
```

handling outliers

6*th* Step:

Handling outliers in data.

```
for col in ['Diabetes', 'Genetic Risk Factor (APOE-ε4 allele)']:
    mode_value = df[col].mode()[0]
    df[col] = df[col].mask((df[col] < lower_lim[col]) | (df[col] > upper_lim[col]), mode_value)
print("Remaining outliers : \n ", ((df < lower_lim) | (df > upper_lim)).sum())
```

```
Remaining outliers :
  Country                              0
Age                                    0
Gender                                 0
Education Level                        0
BMI                                    0
Physical Activity Level                0
Smoking Status                         0
Alcohol Consumption                    0
Diabetes                               0
Hypertension                           0
Cholesterol Level                      0
Family History of Alzheimer's          0
Cognitive Test Score                   0
Depression Level                       0
Sleep Quality                          0
Dietary Habits                         0
Air Pollution Exposure                 0
Employment Status                      0
Marital Status                         0
Genetic Risk Factor (APOE-ε4 allele)   0
Social Engagement Level                0
Income Level                           0
Stress Levels                          0
Urban vs Rural Living                  0
Alzheimer's Diagnosis                  0
dtype: int64
```

```
columns_to_drop = [
    'Country', 'Employment Status', 'Marital Status',
    'Income Level', 'Urban vs Rural Living','BMI'
]
df_cleaned = df.drop(columns=columns_to_drop)
```

```
df_cleaned.columns
```

```
Index(['Age', 'Gender', 'Education Level', 'Physical Activity Level',
       'Smoking Status', 'Alcohol Consumption', 'Diabetes', 'Hypertension',
       'Cholesterol Level', 'Family History of Alzheimer's',
       'Cognitive Test Score', 'Depression Level', 'Sleep Quality',
       'Dietary Habits', 'Air Pollution Exposure',
       'Genetic Risk Factor (APOE-ε4 allele)', 'Social Engagement Level',
       'Stress Levels', 'Alzheimer's Diagnosis'],
      dtype='object')
```

Creating a new csv after feature engineering and reading the data from csv

```
df_cleaned.to_csv("Alzheimer_prediction_cleaned.csv", index = False)
```

7*th* Step:

Superior University lahore

Training and testing data.

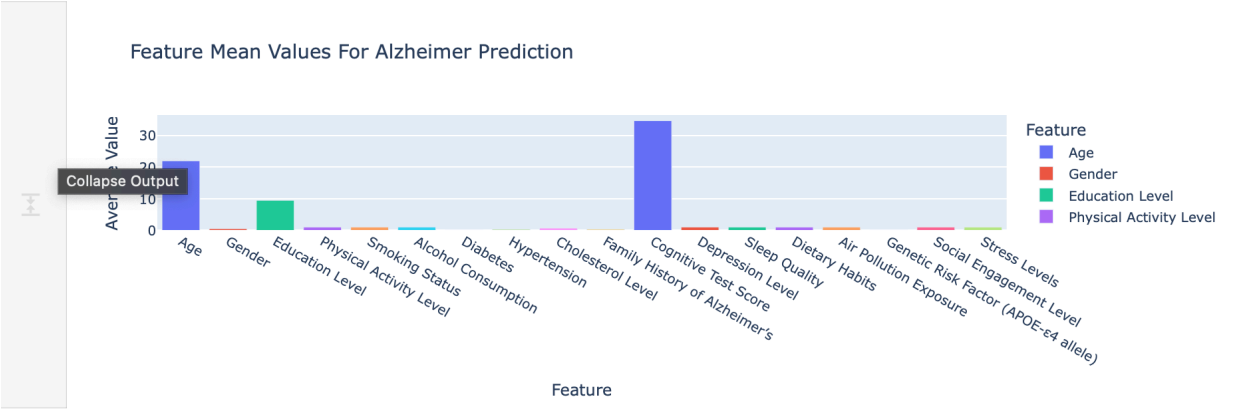
```
[9]: X = df_cleaned.drop(columns=['Alzheimer's Diagnosis'])
     y = df_cleaned['Alzheimer's Diagnosis']
```

```
[0]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

visualize the mean values of different features

```
[9]: import plotly.express as px

feature_means = df_cleaned.drop(columns=['Alzheimer's Diagnosis']).mean().reset_index()
feature_means.columns = ['Feature', 'Mean Value']
fig = px.bar(feature_means, x='Feature', y='Mean Value', title='Feature Mean Values For Alzheimer Prediction',
             labels={'Mean Value': 'Average Value'}, color='Feature')
fig.show()
```

```
feature_means = df_cleaned.drop(columns=['Alzheimer's Diagnosis']).mean().reset_index()
feature_means.columns = ['Feature', 'Mean Value']
fig = px.bar(feature_means, x='Feature', y='Mean Value', title='Feature Mean Values For Alzheimer Prediction',
             labels={'Mean Value': 'Average Value'}, color='Feature')
fig.show()
```




```
[32]: print("Training data shape:", X_train.shape)
      print("Testing data shape:", X_test.shape)
```

```
Training data shape: (59426, 18)
Testing data shape: (14857, 18)
```

[32]:
```
print("Training data shape:", X_train.shape)
print("Testing data shape:", X_test.shape)
```

```
Training data shape: (59426, 18)
Testing data shape: (14857, 18)
```

Moldel training for SVC

[36]:
```
model_svc = SVC(kernel='rbf', C=1.0, gamma='scale')
model_svc.fit(X_train, y_train)
print(model_svc)
```

```
SVC()
```

8*th* Step:

Model Training with svc and random forest, displaying AUC for model Accuracy.

Store the file with the help of pickel

```
7]: pickle.dump(model_svc, open('model_svc.pkl', 'wb'))
```

```
8]: model_svc = pickle.load(open('model_svc.pkl', 'rb'))
```

```
9]: model_predictions = model_svc.predict(X_test)
```

Accuracy of model and report of it

```
3]: from sklearn.metrics import classification_report
accuracy = accuracy_score(y_test, model_predictions)
print(f"Model Accuracy: {accuracy:.2%}")
```

```
Model Accuracy: 71.94%
```

```
2]: print(classification_report(y_test, model_predictions))
```

```
              precision    recall  f1-score   support

           0       0.77      0.75      0.76      8719
           1       0.65      0.68      0.67      6138

    accuracy                           0.72     14857
   macro avg       0.71      0.71      0.71     14857
weighted avg       0.72      0.72      0.72     14857
```

training an other model

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
print(rf_model)
```

```
RandomForestClassifier(random_state=42)
```

```
import joblib
joblib.dump(rf_model, "random_forest_model.pkl")
```

```
['random_forest_model.pkl']
```

```
loaded_model = joblib.load("random_forest_model.pkl")
```

```
new_predictions = loaded_model.predict(X_test)
```

```
print(f"Model Accuracy: {accuracy:.2%}")
```

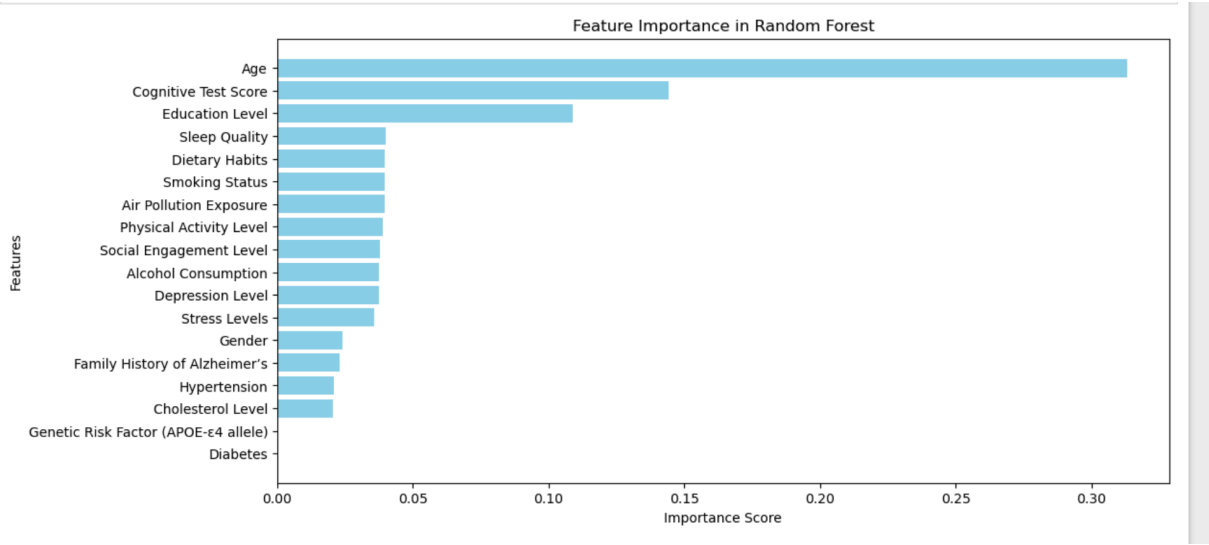
```
Model Accuracy: 71.94%
```

```
print("Classification Report:\n", classification_report(y_test, new_predictions))
```

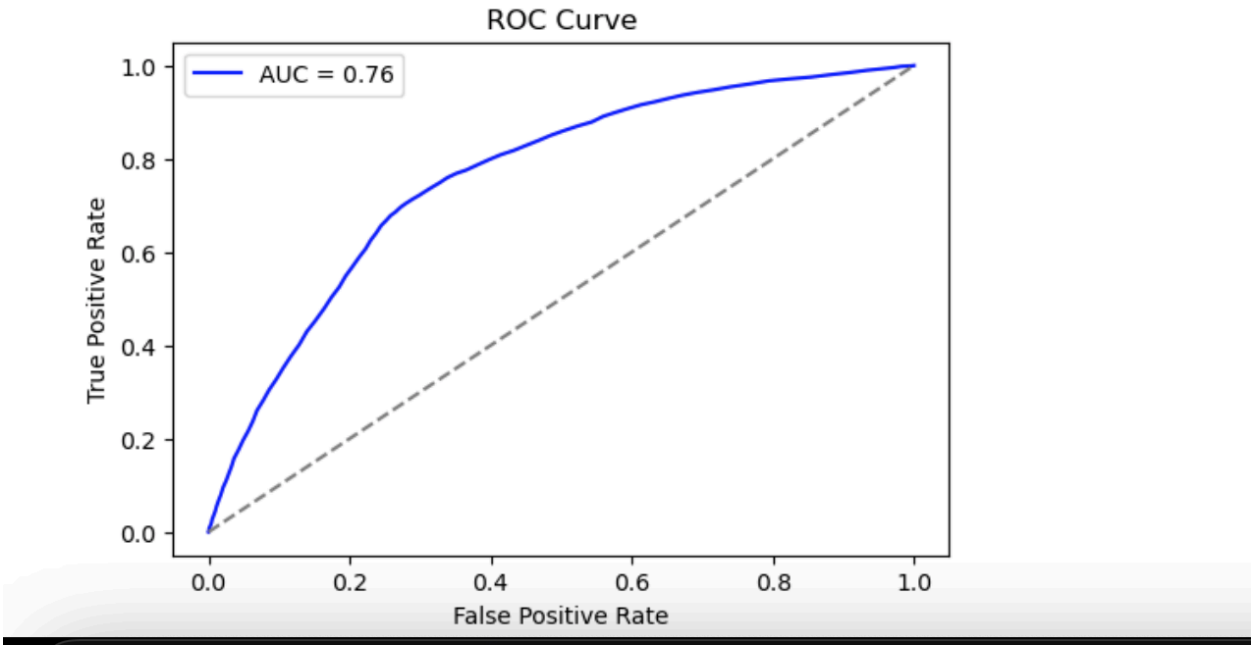
```
Classification Report:
               precision    recall  f1-score   support

           0       0.75      0.76      0.76      8719
           1       0.65      0.64      0.65      6138

    accuracy                           0.71     14857
   macro avg       0.70      0.70      0.70     14857
weighted avg       0.71      0.71      0.71     14857
```

visualizes the feature importance scores

```
import matplotlib.pyplot as plt
import pandas as pd
importances = rf_model.feature_importances_
feature_importance_df = pd.DataFrame({'Feature': X.columns, 'Importance': importances})
feature_importance_df = feature_importance_df.sort_values(by='Importance', ascending=False)
plt.figure(figsize=(12, 6))
plt.barh(feature_importance_df['Feature'], feature_importance_df['Importance'], color='skyblue')
plt.xlabel("Importance Score")
plt.ylabel("Features")
plt.title("Feature Importance in Random Forest")
plt.gca().invert_yaxis()
plt.show()
```


Feature Importance in Random Forest
Age
Cognitive Test Score
Education Level
Sleep Quality
Dietary Habits
Smoking Status
Air Pollution Exposure
Physical Activity Level
Social Engagement Level
Alcohol Consumption
Depression Level
Stress Levels
Gender
Family History of Alzheimer's
Hypertension
Cholesterol Level
Genetic Risk Factor (APOE-ε4 allele)
Diabetes
Features
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Importance Score

```
[62]: from sklearn.metrics import roc_curve, auc
y_probs = rf_model.predict_proba(X_test)[:, 1]
fpr, tpr, _ = roc_curve(y_test, y_probs)
roc_auc = auc(fpr, tpr)
plt.figure(figsize=(6, 4))
plt.plot(fpr, tpr, color='blue', label=f'AUC = {roc_auc:.2f}')
plt.plot([0, 1], [0, 1], color='gray', linestyle='--')  # Baseline
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve")
plt.legend()
plt.show()
```



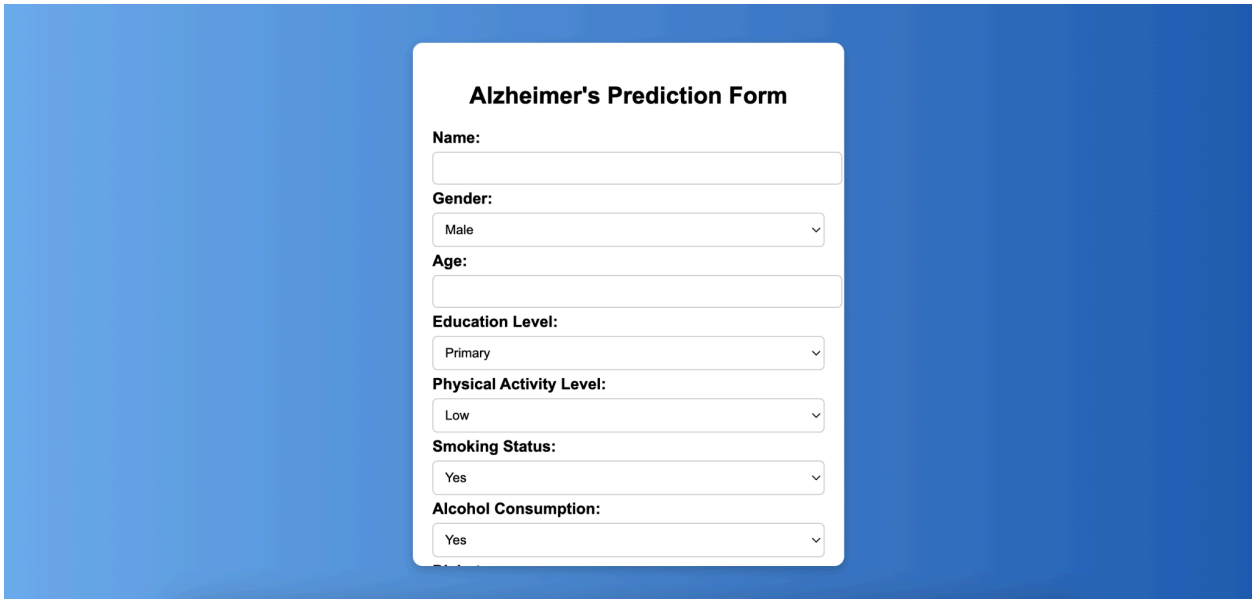
Flask Application:

We have developed a Flask application that predicts Alzheimer's disease using a trained Random Forest model. The application takes input from users based on 18 selected features, which are crucial for the prediction. Once the user provides the necessary details, the model processes the data and generates a comprehensive report on the prediction results. This report is then

made available for download, allowing users to keep a record of their assessment. The application ensures a seamless and efficient experience, making Alzheimer's prediction more accessible and user-friendly.

Flask Display:

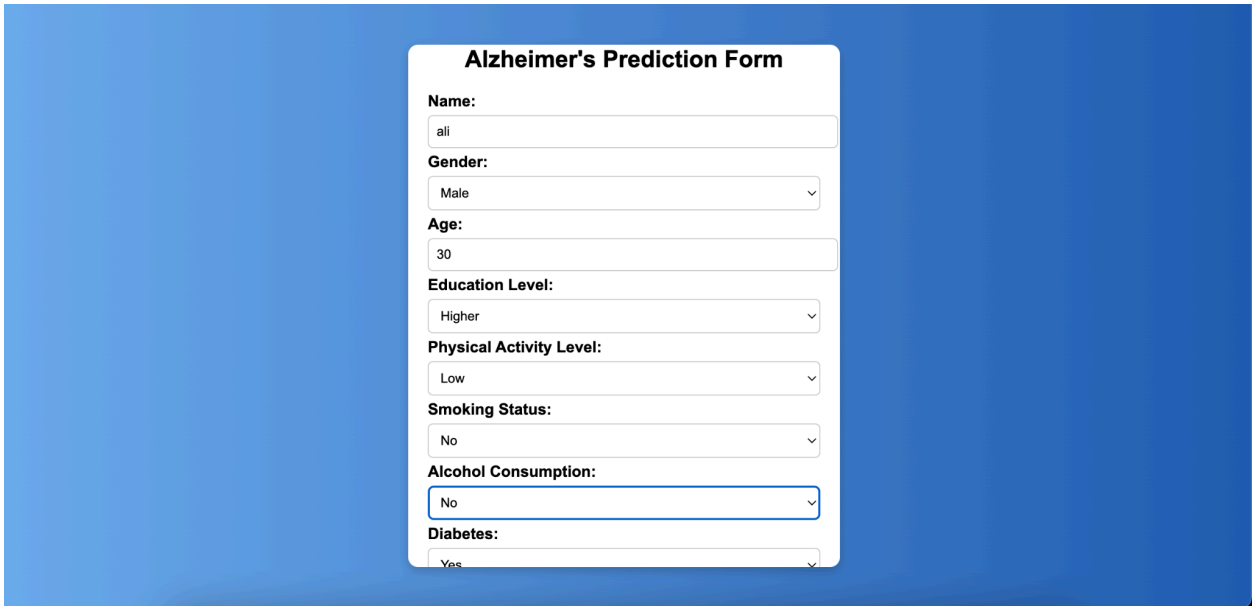
Our Flask application has a clean and centered design where all input fields are neatly placed in the middle of the page. A background color enhances the visual appeal, providing a smooth and user-friendly interface. This layout ensures easy navigation and accessibility, making it convenient for users to enter data and generate predictions seamlessly.



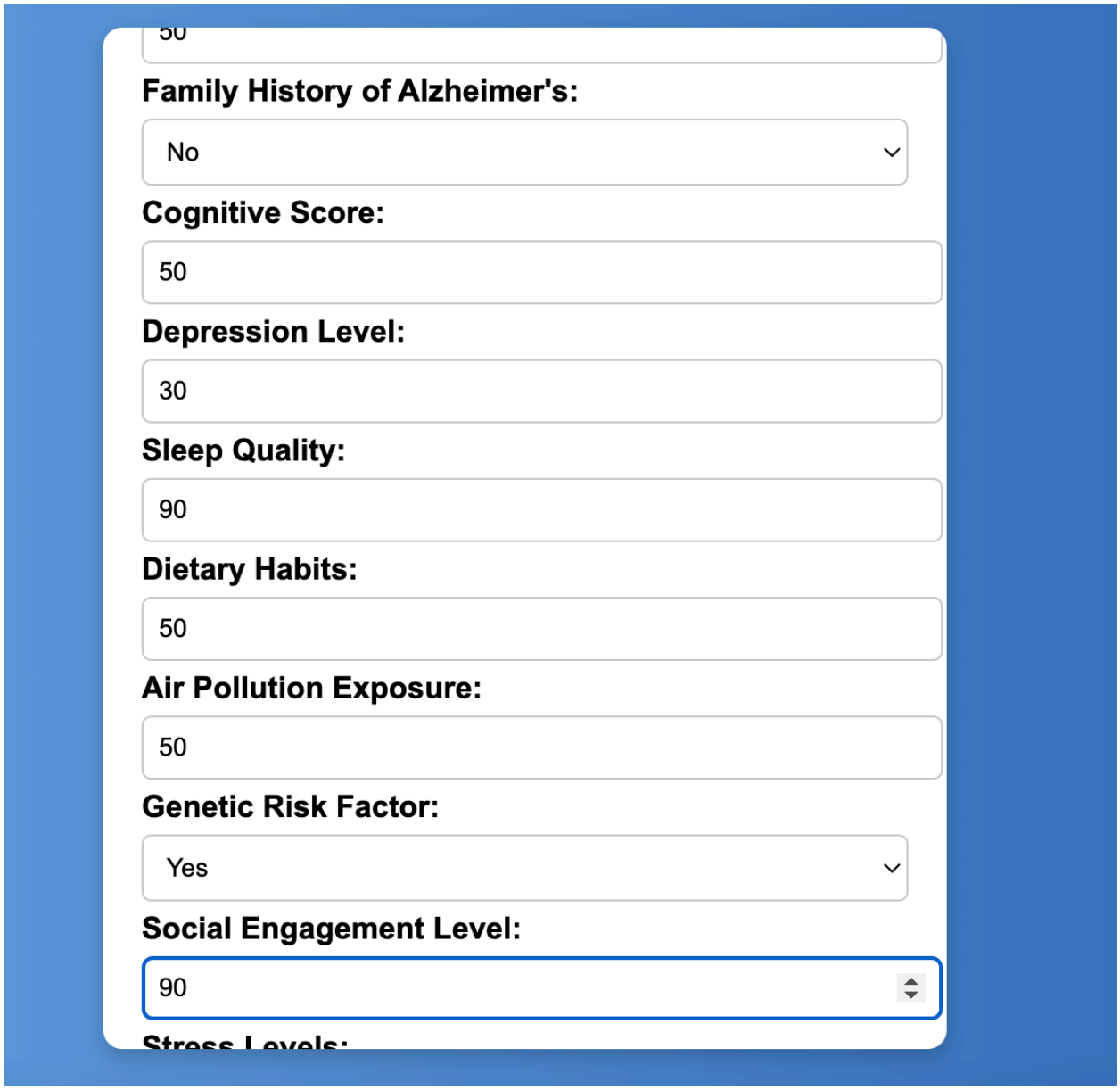
Form for Prediction:

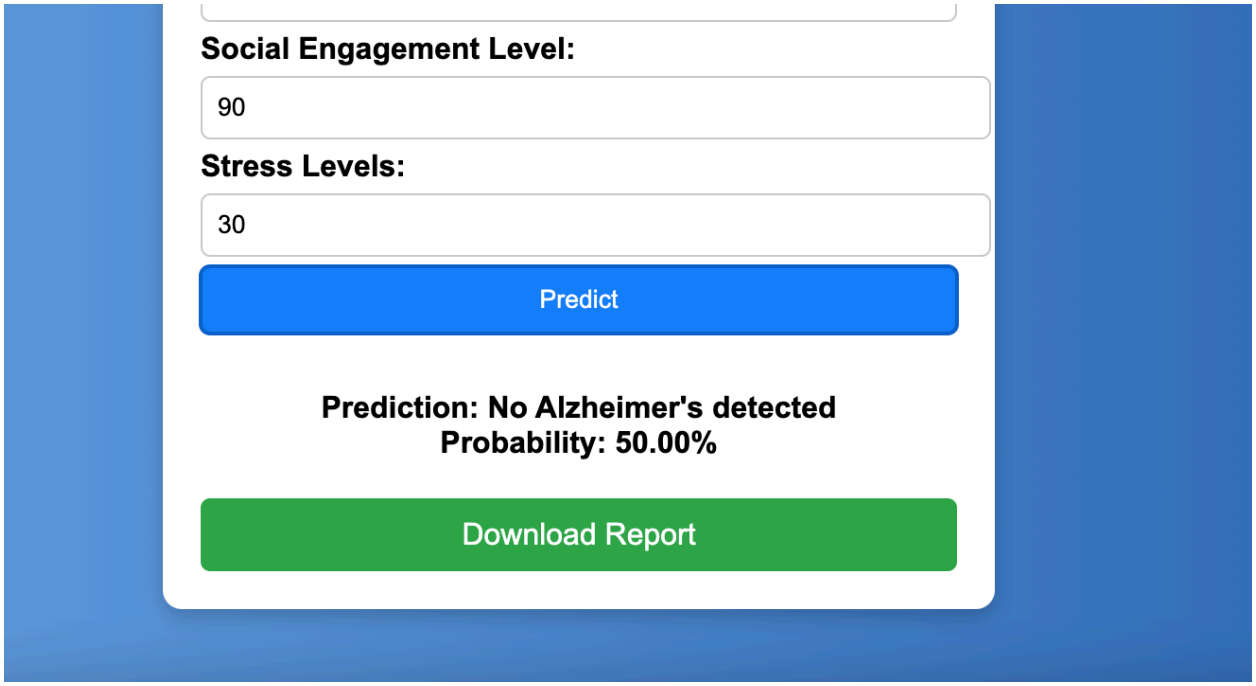
When a user fills out the form in the Flask application, the provided input values are processed by the trained Random Forest model to predict the likelihood of Alzheimer's disease. The form consists of 18 selected features that play a crucial role in determining the outcome. Once the user submits the form, the model analyzes the data and generates a prediction based on the

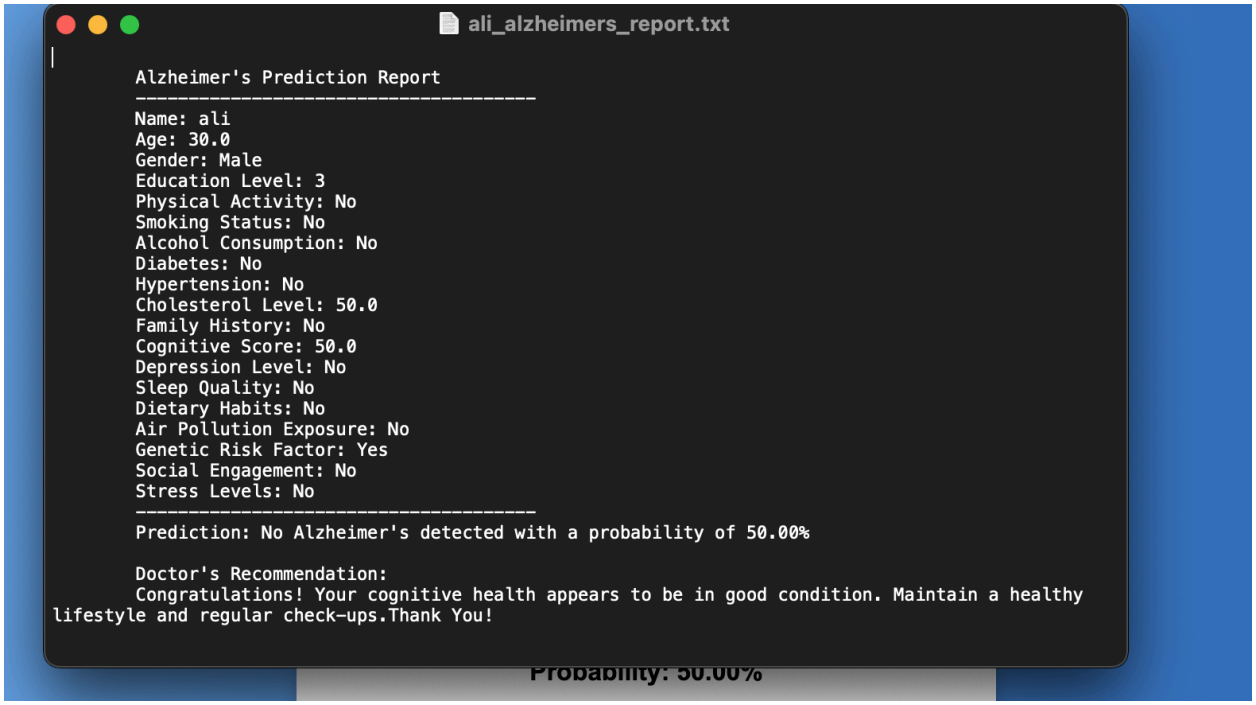
trained patterns. A detailed report is then created, summarizing the results, which the user can download for further reference. This streamlined process ensures accurate and efficient predictions, making the application user-friendly and accessible.



Superior University lahore


50
Family History of Alzheimer's:
No
Cognitive Score:
50
Depression Level:
30
Sleep Quality:
90
Dietary Habits:
50
Air Pollution Exposure:
50
Genetic Risk Factor:
Yes
Social Engagement Level:
90
Stress Levels:





All the requirements are fulfilled of Assignment 1 & 2 .

Superior University lahore