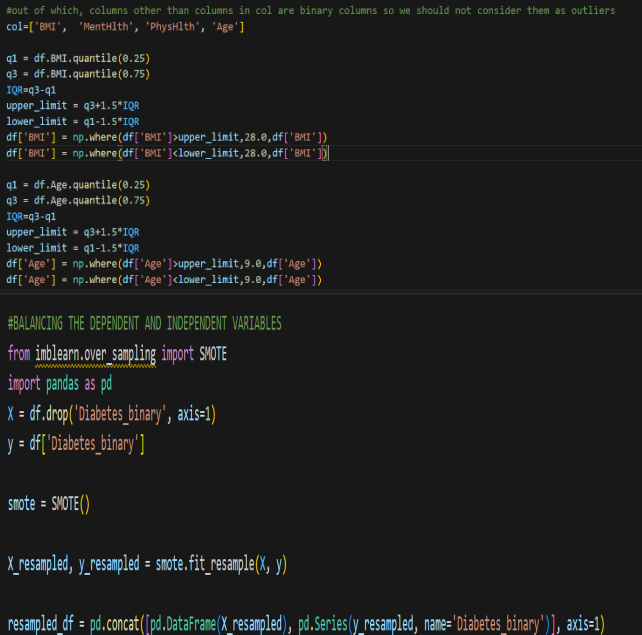


Project Development Phase Model Performance Test

Date	09 November 2023
Project Name	Diabetes Prediction
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Metric s	Regression Model: MAE - , MSE - , RMSE - , R2 score - Classification Model: Confusion Matrix - , Accuray Score- & Classification Report -	 <pre> confusion_matrix(y_test,y_pred_3) array([[5828, 230], [704, 5406]]) print("train accuracy",accuracy_score(y_train,train_pred_3)) print("test accuracy",accuracy_score(y_test,y_pred_3)) train accuracy 0.9957147028345366 test accuracy 0.9180731444798131 </pre>
2.	Tune the Model	Hyperparameter Tuning - Validation Method -	 <pre> #out of which, columns other than columns in col are binary columns so we should not consider them as outliers col=['BMI', 'MenthHth', 'PhysHlth', 'Age'] q1 = df.BMI.quantile(0.25) q3 = df.BMI.quantile(0.75) IQR=q3-q1 upper_limit = q3+1.5*IQR lower_limit = q1-1.5*IQR df['BMI'] = np.where(df['BMI']>upper_limit,28.0,df['BMI']) df['BMI'] = np.where(df['BMI']<lower_limit,28.0,df['BMI']) q1 = df.Age.quantile(0.25) q3 = df.Age.quantile(0.75) IQR=q3-q1 upper_limit = q3+1.5*IQR lower_limit = q1-1.5*IQR df['Age'] = np.where(df['Age']>upper_limit,9.0,df['Age']) df['Age'] = np.where(df['Age']<lower_limit,9.0,df['Age']) #BALANCING THE DEPENDENT AND INDEPENDENT VARIABLES from imblearn.over_sampling import SMOTE import pandas as pd X = df.drop('Diabetes_binary', axis=1) y = df['Diabetes_binary'] smote = SMOTE() X_resampled, y_resampled = smote.fit_resample(X, y) resampled_df = pd.concat([pd.DataFrame(X_resampled), pd.Series(y_resampled, name='Diabetes_binary')], axis=1) </pre>

```

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, roc_auc_score, roc_curve
print("train accuracy", accuracy_score(y_train, train_pred_3))
print("test accuracy", accuracy_score(y_test, y_pred_3))

```

```

train accuracy 0.9957147028345366
test accuracy 0.9180731444798131

```

```

confusion_matrix(y_test, y_pred_3)

```

```

array([[5828, 230],
       [ 704, 5406]])

```

```

#out of which, columns other than columns in col are binary columns so we should not consider them as outliers
col=['BMI', 'MentHlth', 'PhysHlth', 'Age']

```

```

q1 = df.BMI.quantile(0.25)
q3 = df.BMI.quantile(0.75)
IQR=q3-q1
upper_limit = q3+1.5*IQR
lower_limit = q1-1.5*IQR
df['BMI'] = np.where(df['BMI']>upper_limit,28.0,df['BMI'])
df['BMI'] = np.where(df['BMI']<lower_limit,28.0,df['BMI'])

```

```

q1 = df.Age.quantile(0.25)
q3 = df.Age.quantile(0.75)
IQR=q3-q1
upper_limit = q3+1.5*IQR
lower_limit = q1-1.5*IQR
df['Age'] = np.where(df['Age']>upper_limit,9.0,df['Age'])
df['Age'] = np.where(df['Age']<lower_limit,9.0,df['Age'])

```

```

#BALANCING THE DEPENDENT AND INDEPENDENT VARIABLES

```

```

from imblearn.over_sampling import SMOTE
import pandas as pd
X = df.drop('Diabetes_binary', axis=1)
y = df['Diabetes_binary']

```

```

smote = SMOTE()

```

```

X_resampled, y_resampled = smote.fit_resample(X, y)

```

```

resampled_df = pd.concat([pd.DataFrame(X_resampled), pd.Series(y_resampled, name='Diabetes_binary')], axis=1)

```