# Capstone Project-3
## Cardiovascular Risk Prediction

By
K Vidyasagar

# Points to be discussed:

- Problem statement
- Data Summary
- Data Cleaning
- Univariate Analysis
- Bivariate Analysis
- Resampling the dataset
- Machine Learning Models Training and Testing
- Conclusions

# Problem statement:

**The dataset is from an ongoing cardiovascular study on residents of the town of Framingham, Massachusetts.**
**The classification goal is to predict whether the patient has a 10-year risk of future coronary heart disease (CHD). The dataset provides the patients information. It includes over 4,000 records and 15 attributes. Each attribute is a potential risk factor. There are both demographic, behavioral, and medical risk factors.**

# Data Summary:

**Demographic:**

• Sex: male or female ("M" or "F")

• Age: Age of the patient;(Continuous - Although the recorded ages have been truncated to whole numbers, the concept of age is continuous)

**Behavioural:**

• is_smoking: whether or not the patient is a current smoker ("YES" or "NO")

• Cigs Per Day: the number of cigarettes that the person smoked on average in one day. (Can be considered continuous as one can have any number of cigarettes, even half a cigarette.)

# Data Summary:

**Medical (history):**

- BP Meds: whether or not the patient was on blood pressure medication (Nominal)
- Prevalent Stroke: whether or not the patient had previously had a stroke (Nominal)
- Prevalent Hyp: whether or not the patient was hypertensive (Nominal)
- Diabetes: whether or not the patient had diabetes (Nominal) Medical(current)
- Tot Chol: total cholesterol level (Continuous)
- SysBP: systolic blood pressure (Continuous)
- DiaBP: diastolic blood pressure (Continuous)
- BMI: Body Mass Index (Continuous)

# Data Summary:

**Medical (history):**

- Heart Rate: heart rate (Continuous - In medical research, variables such as heart rate though in fact discrete, yet are considered continuous because of large number of possible values.)
- Glucose: glucose level (Continuous) Predict variable (desired target)
- **Target variable: t**he patient has a 10-year risk of future coronary heart disease (CHD).

# Data Cleaning:

## Missing values in different features:

```
# missing values in each feature
df.isnull().sum().sort_values(ascending=False)
```

```
glucose          304
education         87
BPMeds            44
totChol           38
cigsPerDay        22
BMI               14
heartRate          1
```

```
# Replacing null values with the median
for col in ['glucose','education','BPMeds','totChol','cigsPerDay','BMI','heartRate']:
  df[col] = df[col].fillna(df[col].median())
```

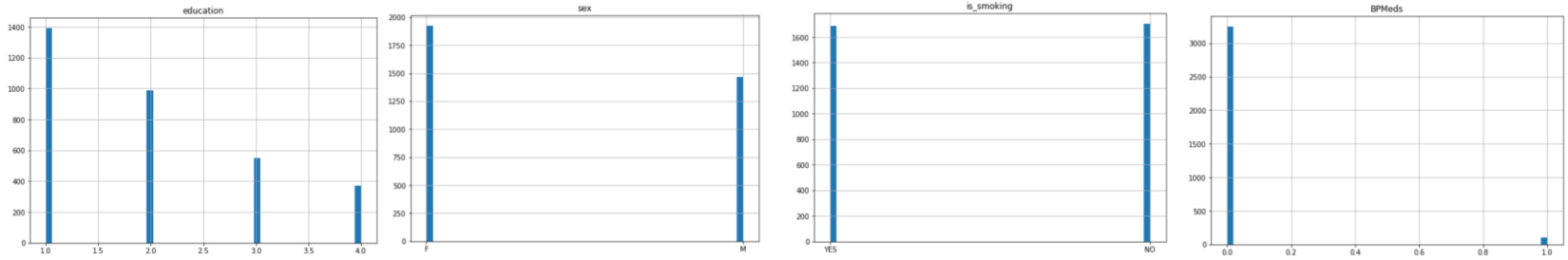
# Univariate Analysis:

## Univariate Analysis:

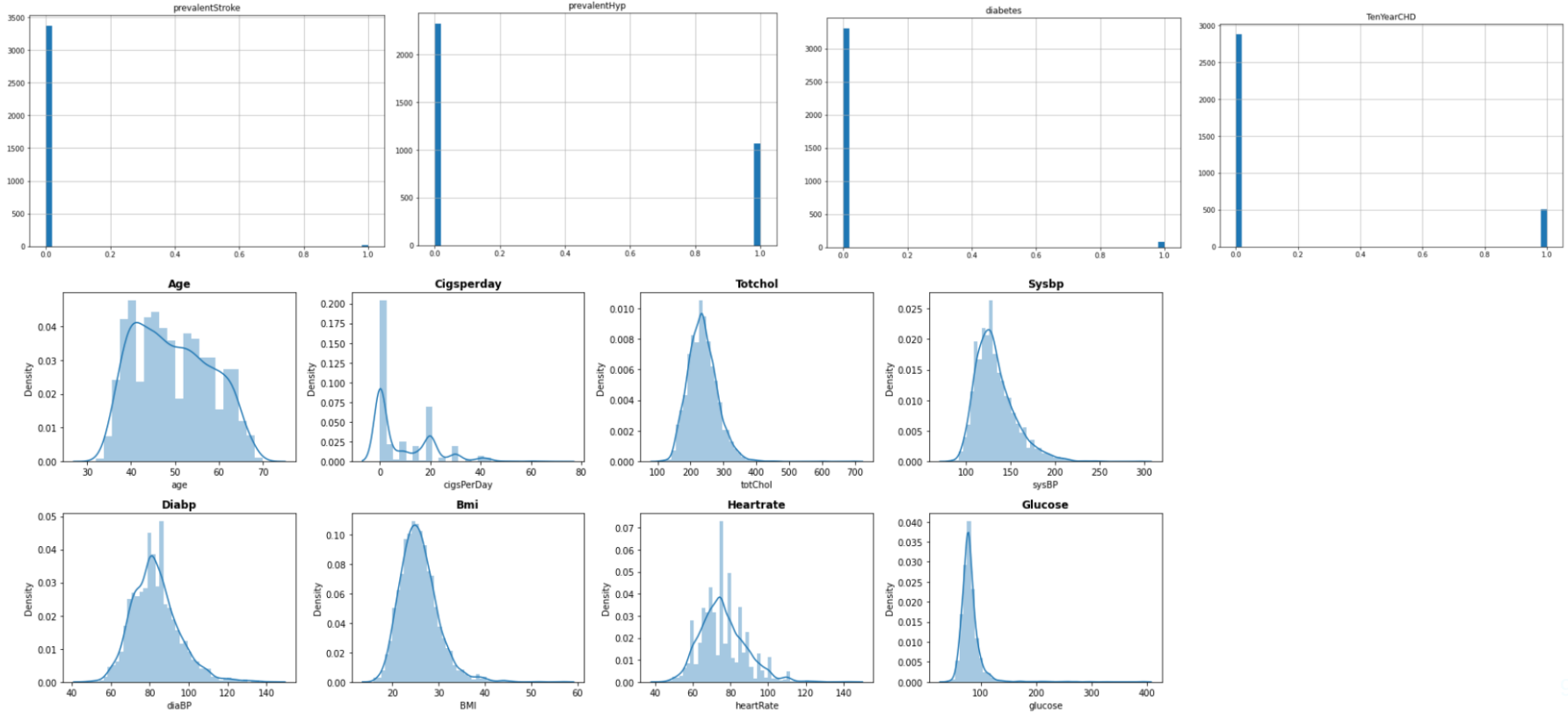
```
# numeric features in our data set
# Lets check the discrete and continuous features
categorical_features = [i for i in df.columns if df[i].nunique()<=4]
numeric_features = [i for i in df.columns if i not in categorical_features]

print(categorical_features)
print(numeric_features)
```

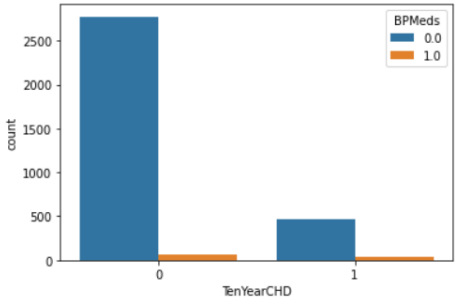
```
['education', 'sex', 'is_smoking', 'BPMeds', 'prevalentStroke', 'prevalentHyp', 'diabetes', 'TenYearCHD']
['age', 'cigsPerDay', 'totChol', 'sysBP', 'diaBP', 'BMI', 'heartRate', 'glucose']
```

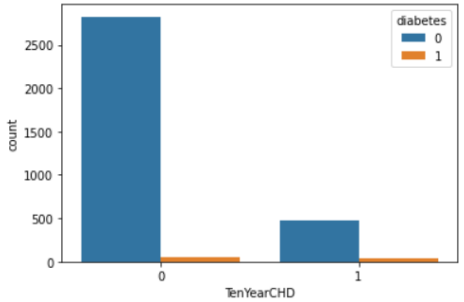


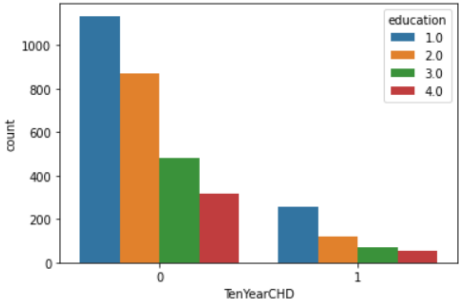
# Univariate Analysis:

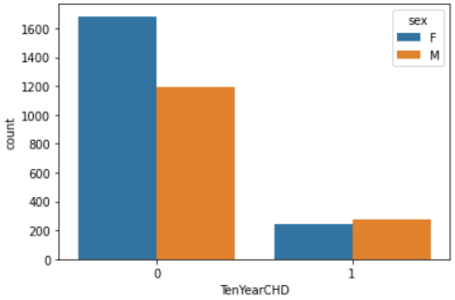


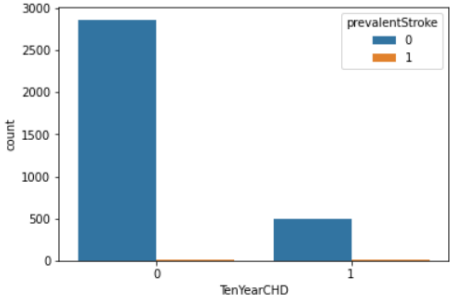
# Bivariate Analysis:

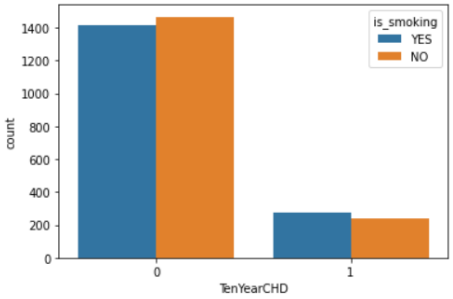


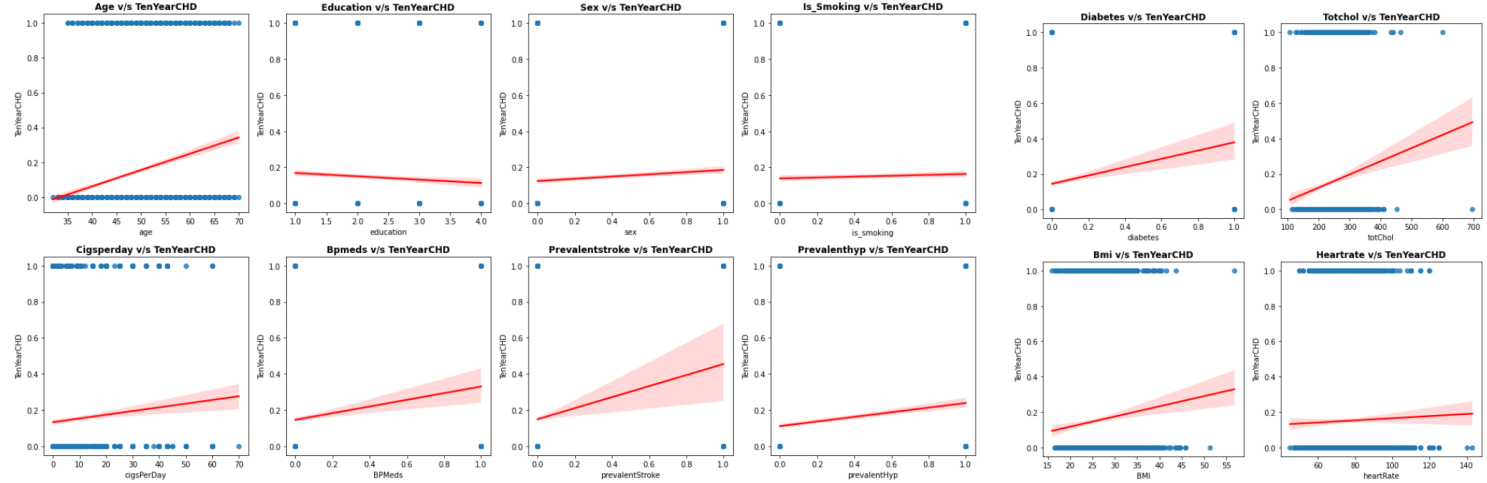



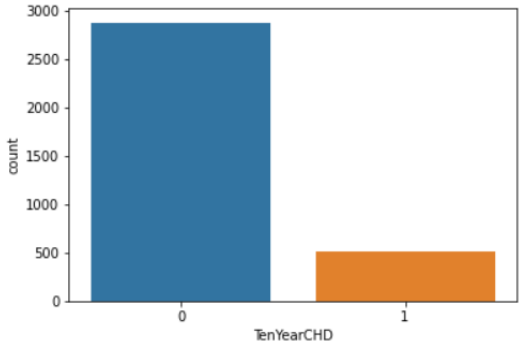



# Bivariate Analysis:



# Resampling the unbalanced dataset:

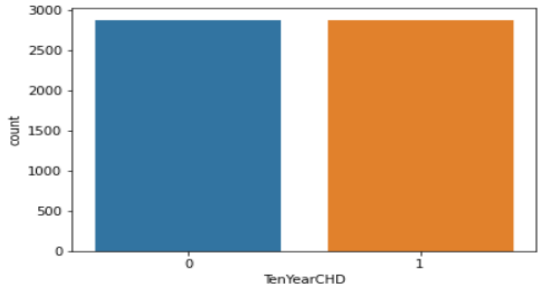



```
from sklearn.utils import resample
#create two different dataframe of majority and minority class
df_majority = df[(df['TenYearCHD']==0)]
df_minority = df[(df['TenYearCHD']==1)]
# upsample minority class
df_minority_upsampled = resample(df_minority,
                                 replace=True,    # sample with replacement
                                 n_samples= 2879, # to match majority class
                                 random_state=40)  # reproducible result

# Combine majority class with upsampled minority class
df_upsampled = pd.concat([df_minority_upsampled, df_majority])
```

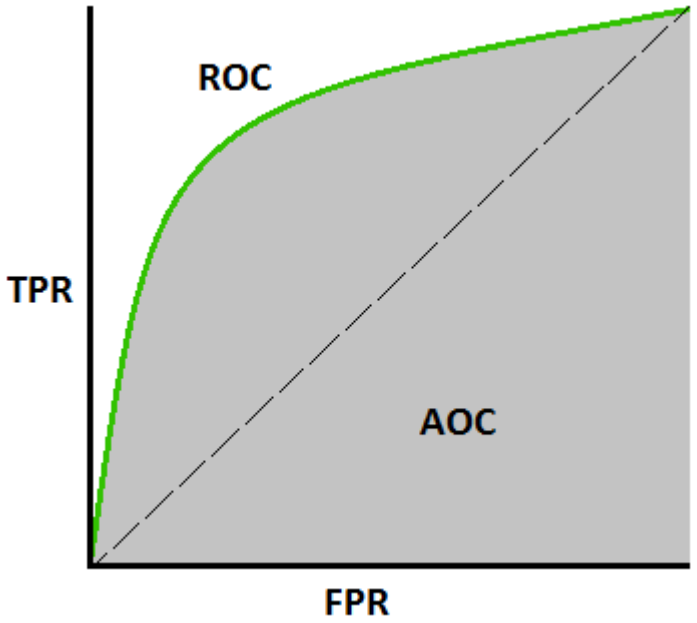
```
sns.countplot(df_upsampled['TenYearCHD'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7feb3a28d2d0>
```

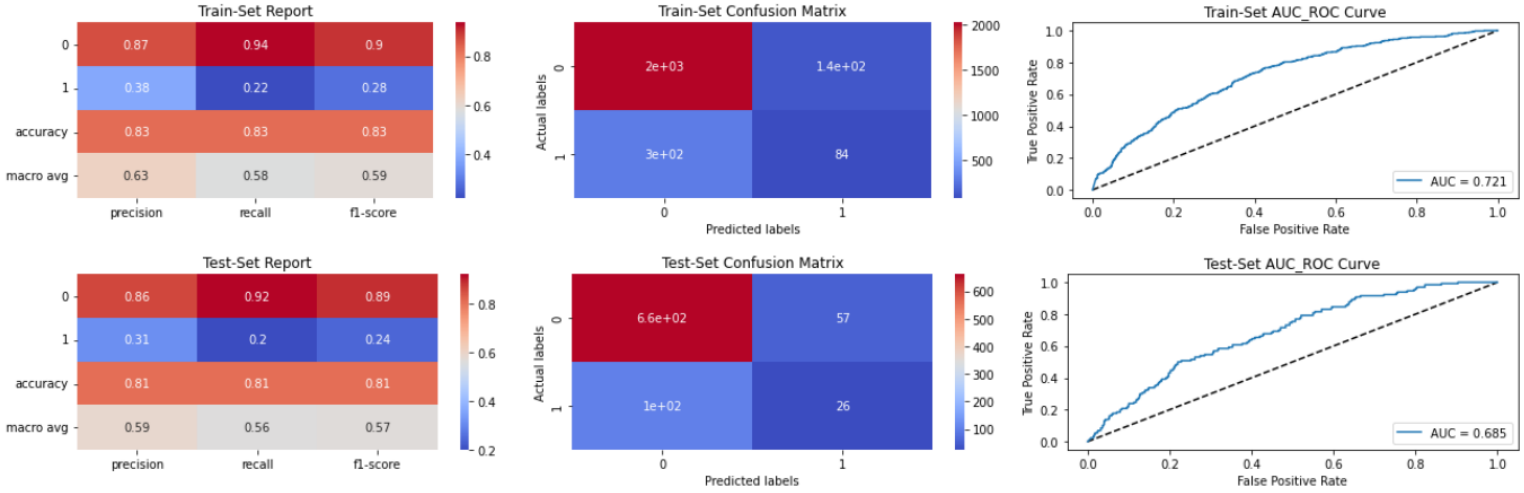


# Machine Learning Models Training and Testing:

| | | Predicted Class: Positive | Predicted Class: Negative | |
|---|---|---|---|---|
| Actual Class | Positive | True Positive (TP) | False Negative (FN) **Type II Error** | **Sensitivity** $\frac{TP}{(TP+FN)}$ |
| | Negative | False Positive (FP) **Type I Error** | True Negative (TN) | **Specificity** $\frac{TN}{(TN+FP)}$ |
| | | **Precision** $\frac{TP}{(TP+FP)}$ | **Negative Predictive Value** $\frac{TN}{(TN+FN)}$ | **Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$ |



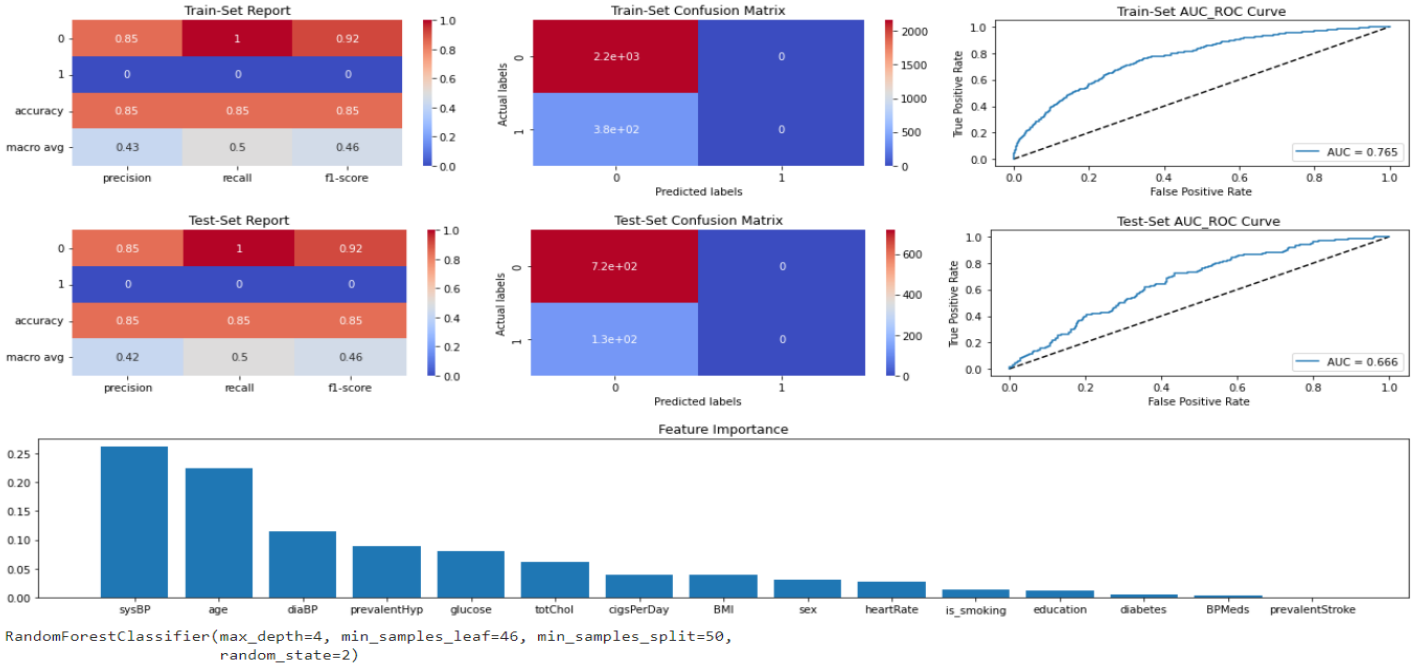
# Machine Learning Models Training and Testing:




<Figure size 1296x216 with 0 Axes>
GaussianNB()

Naive Bayes Classifier on data set without resample

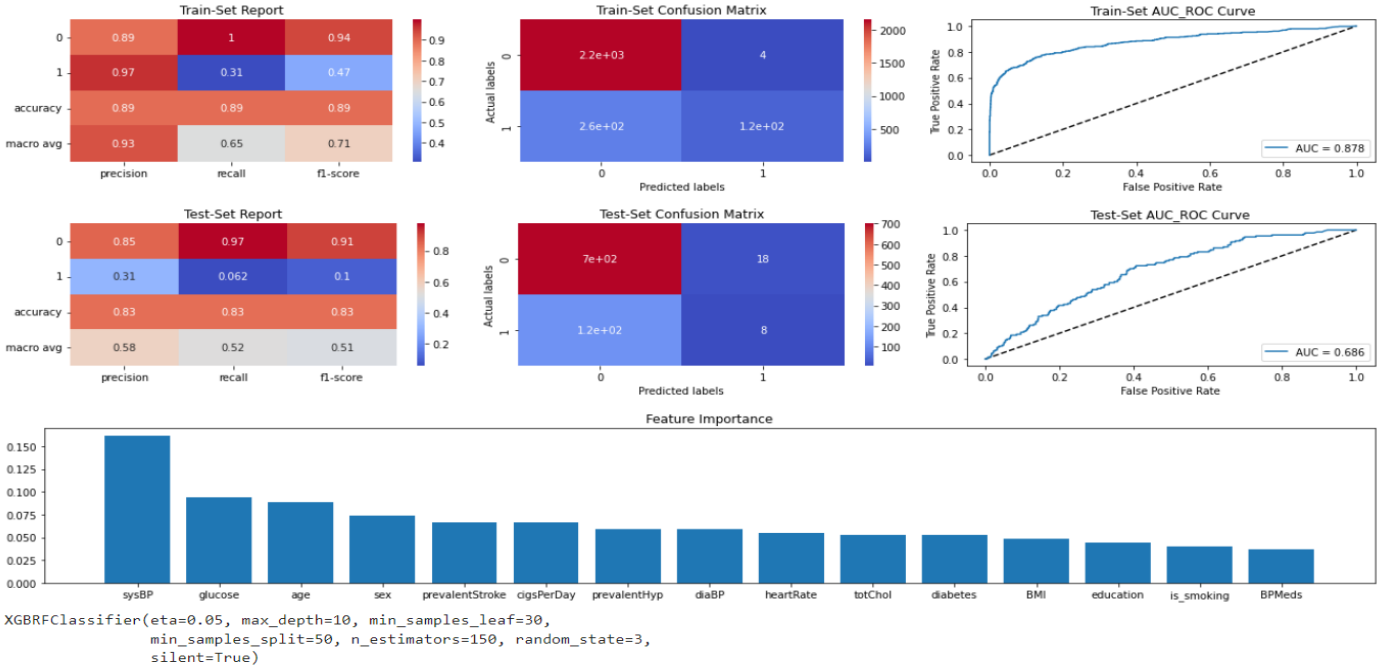
# Machine Learning Models Training and Testing:




```
RandomForestClassifier(max_depth=4, min_samples_leaf=46, min_samples_split=50,
                       random_state=2)
```

Random Forest Classifier on data set without resample

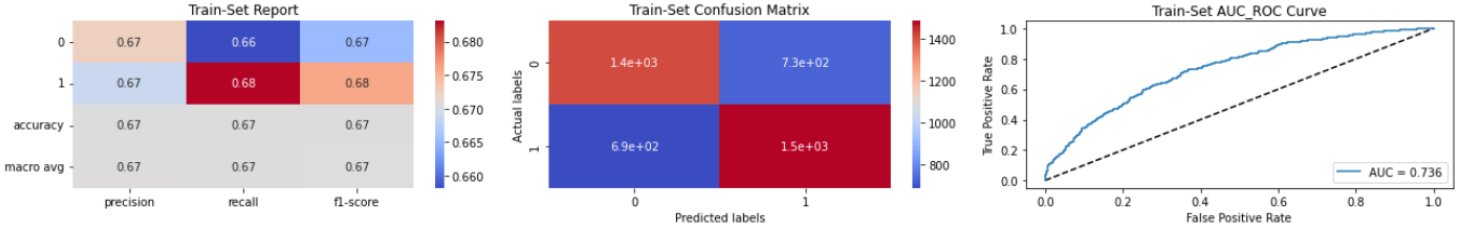
# Machine Learning Models Training and Testing:

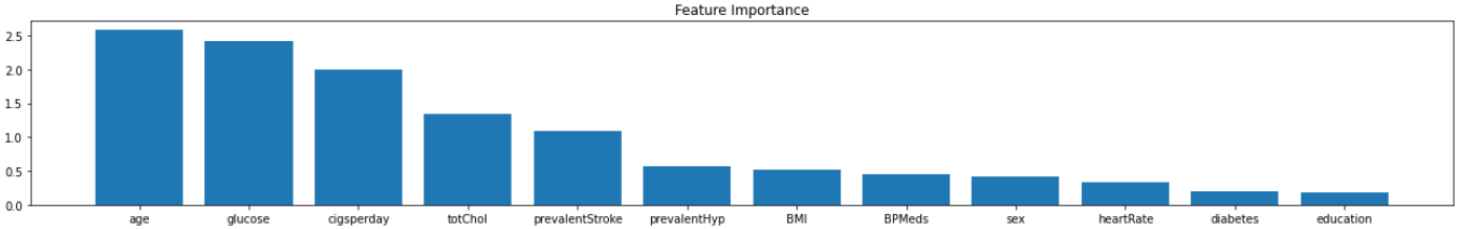



```
XGBRFClassifier(eta=0.05, max_depth=10, min_samples_leaf=30,
                min_samples_split=50, n_estimators=150, random_state=3,
                silent=True)
```

XG Boost Classifier on data set without resample

# Machine Learning Models Training and Testing:

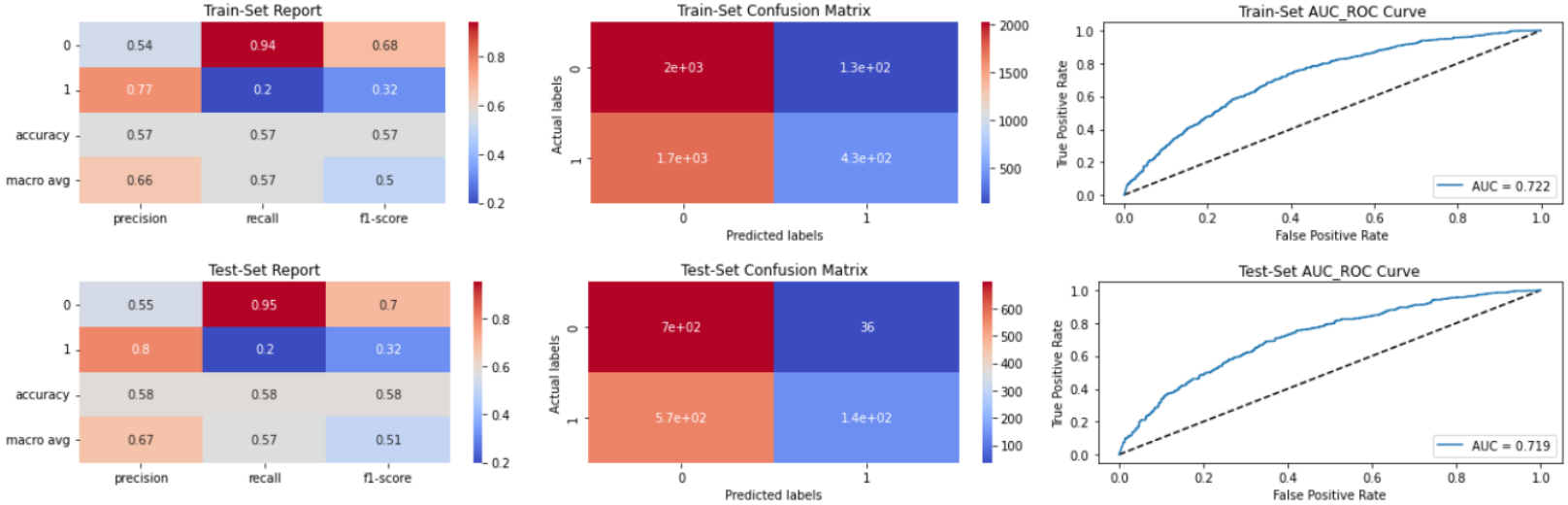



```
LogisticRegression(max_iter=10000)
```

Logistic Regression on data set with resample

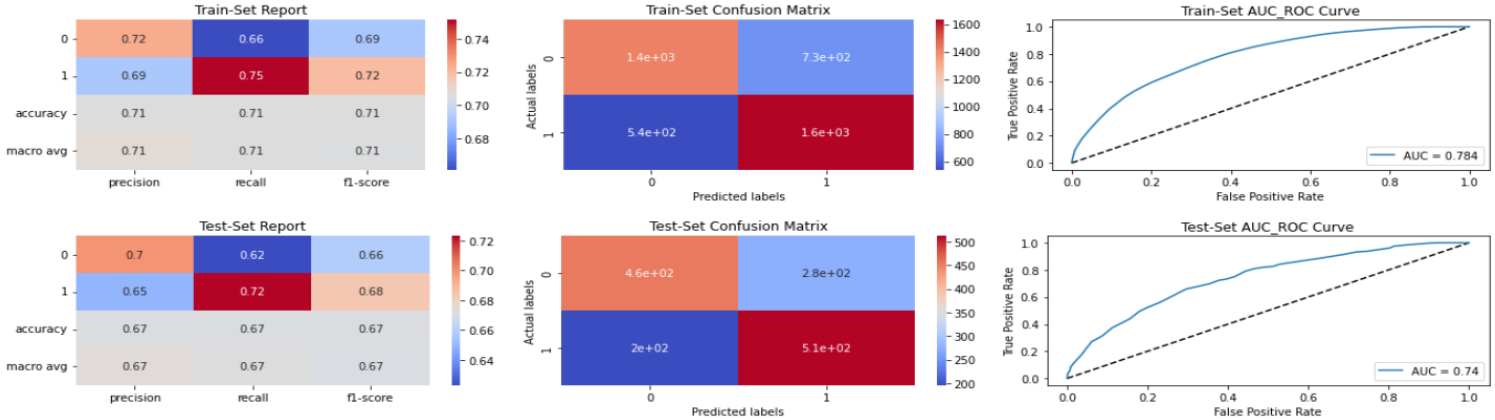
# Machine Learning Models Training and Testing:




<Figure size 1296x216 with 0 Axes>
GaussianNB()

Naive Bayes Classifier on data set with resample

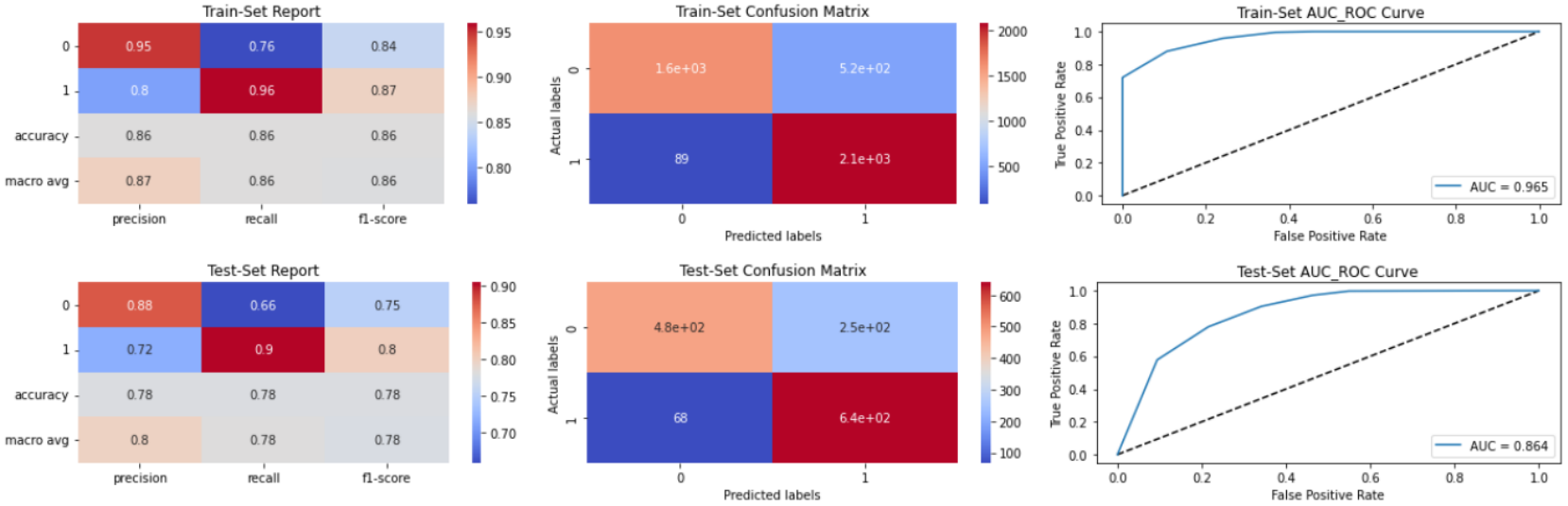
# Machine Learning Models Training and Testing:




```
DecisionTreeClassifier(max_depth=6, min_samples_leaf=60, min_samples_split=50,
                       random_state=10)
```

Decision Tree Classifier on data set with resample

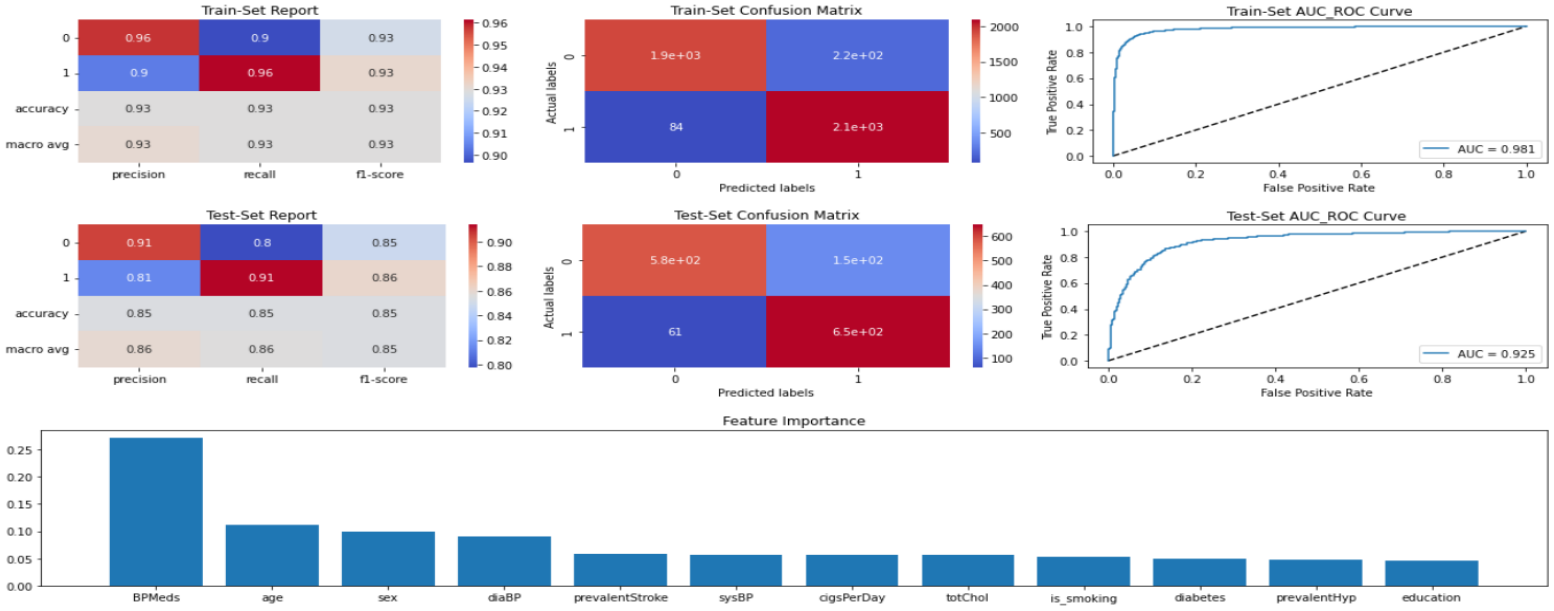
# Machine Learning Models Training and Testing:




```
<Figure size 1296x216 with 0 Axes>
KNeighborsClassifier()
```

KNN Classifier on data set with resample

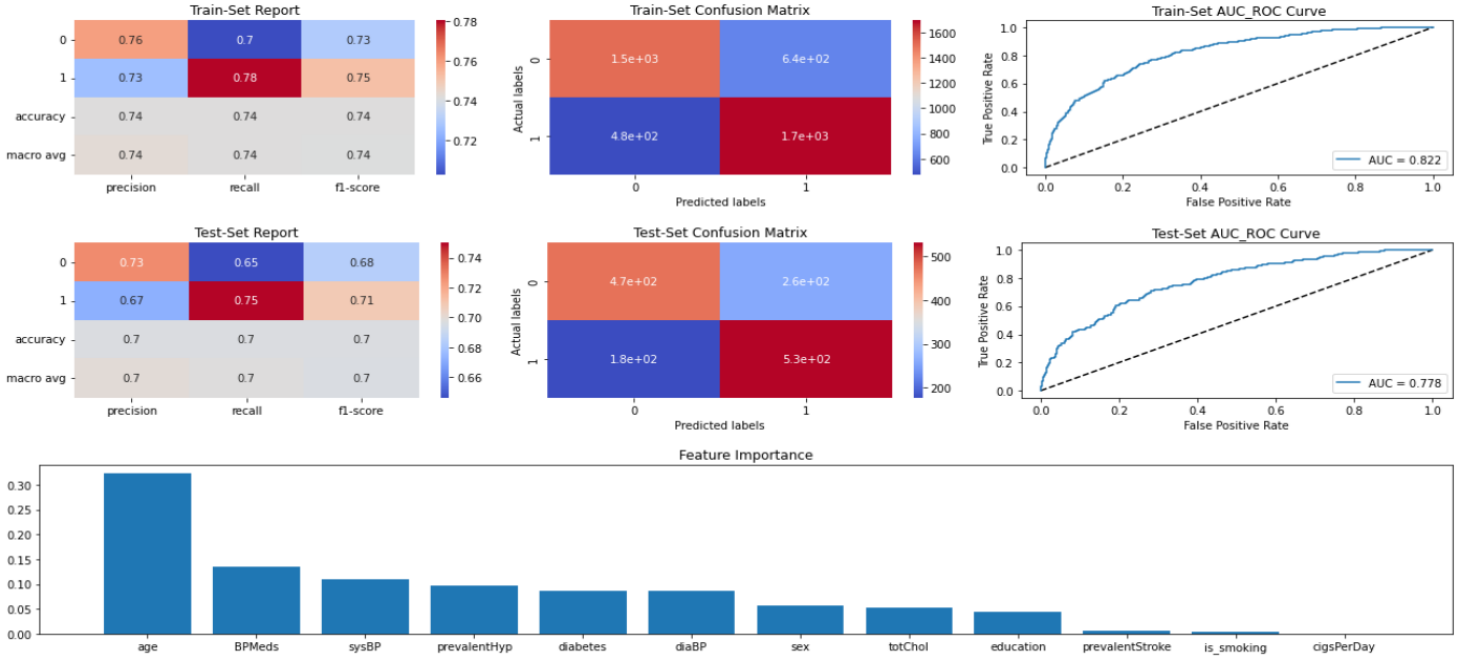
# Machine Learning Models Training and Testing:




```
XGBRFClassifier(eta=0.05, max_depth=10, min_samples_leaf=30,
                min_samples_split=50, n_estimators=150, random_state=11,
                silent=True)
```

XG Boost Classifier on data set with resample

# Machine Learning Models Training and Testing:




```
RandomForestClassifier(max_depth=8, min_samples_leaf=46, min_samples_split=50,
                       n_estimators=150, random_state=12)
```

Random Forest Classifier on data set with resample

# Conclusions:

- Age is ranging from 30 to 70.
- cigsperday is ranging from 0 to 70.
- total cholestral is ranging from 100 to 700.
- sysbp is ranging from 100 to 300.
- diabp is ranging from 40 to 140.
- BMI is ranging from 15 to 55.
- heart rate is ranging from 40 to 140.
- glucose level is from 40 to 400.
- Females are more than number of males in our dataset but number of males prone to heart disease are more compared to females.
- higher education people are less but all education level people having equal share of heart disease prone.
- we have imbalanced dataset. So we built machine learning algorithms in two scenarios.

# Conclusions:

## No Resampling on dataset:

### F1 Score on test dataset :

- Naive Bayes Classifier: 0.57
- Random Forest Classifier: 0.46
- XGBoost Classifier: 0.51

## Resampling dataset case:

### F1 score on test datsset :

- Logistic Regression: 0.66
- Naive Bayes Classifier: 0.51
- Decision Tree Classifier: 0.67
- KNN Classifier: 0.78
- XGBoost Classifier: 0.85
- Random Forest Classifier: 0.7

# Thank You!