HR - ANALYTICS AND EMPLOYEE ATTRITION

- RAKESH M V

PROBLEM STATEMENT

- The objective of this project is to predict the attrition rate for each employee, to find out who’s more likely to leave the organization.
- The key to success in an organisation is the ability to attract and retain top talents. It is vital for the Human Resource (HR) Department to identify the factors that keep employees and those which prompt them to leave.
- It will help organisations to find ways to prevent attrition or to plan in advance the hiring of new candidate.
- Attrition proves to be a costly and time consuming problem for the organization and it also leads to loss of productivity.
- The scope of the project extends to companies in all industries.

DATA SOURCES:

For this project, an HR dataset named '**IBM HR Analytics Employee Attrition& Performance**', has been picked from kaggle datasets.

The data contains records of 18749 employees.

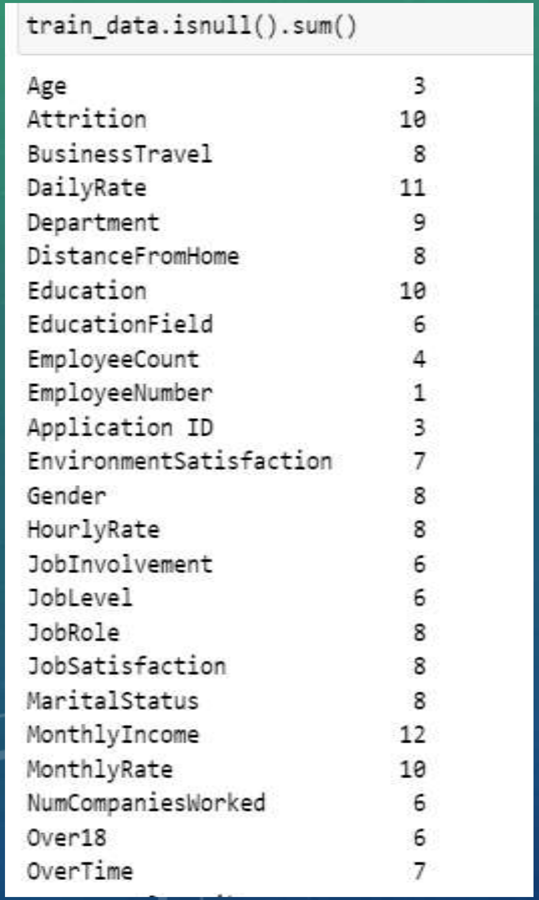
It has information about employee's current employment status, the total number of companies worked for in the past, Total number of years at the current company and the current roles, Their education level, distance from home, monthly income, etc.

ANALYTICS APPROACH

- Check for missing values in the data, and if any, will process the data accordingly.
- Understand how the features are related with our target variable – attrition
- Convert target variable into numeric form
- Apply feature selection and feature engineering to make it model ready
- Apply various algorithms to check which one is the most suitable
- Draw out recommendations based on our analysis.

TOOLS AND TECHNOLOGY

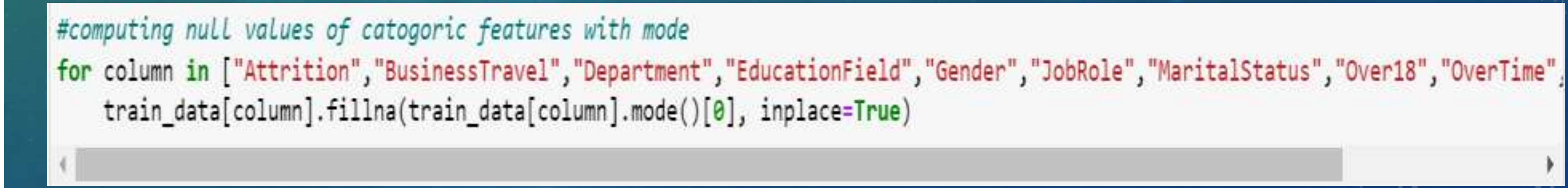
- We have selected Python3 as our analytics tool.
- Python includes many packages such as Pandas, NumPy, Matplotlib, Seaborne etc.
- Algorithms such as Logistic Regression, Random Forest, Decision Tree, Support Vector Machine, KNN ,XGBoost and Gridserch have been used for prediction.



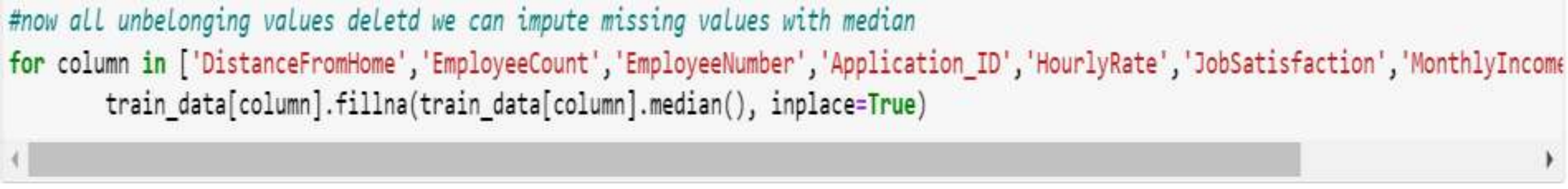
```
train_data.isnull().sum()

Age                          3
Attrition                   10
BusinessTravel               8
DailyRate                   11
Department                   9
DistanceFromHome             8
Education                   10
EducationField               6
EmployeeCount                4
EmployeeNumber               1
Application ID               3
EnvironmentSatisfaction      7
Gender                       8
HourlyRate                   8
JobInvolvement               6
JobLevel                     6
JobRole                      8
JobSatisfaction              8
MaritalStatus                8
MonthlyIncome               12
MonthlyRate                 10
NumCompaniesWorked           6
Over18                       6
OverTime                     7
```

❑ HANDLING MISSING VALUES

- 284 missing values present in the Train dataset.
- To resolve this problem of missing values treat with MEADIAN and MODE



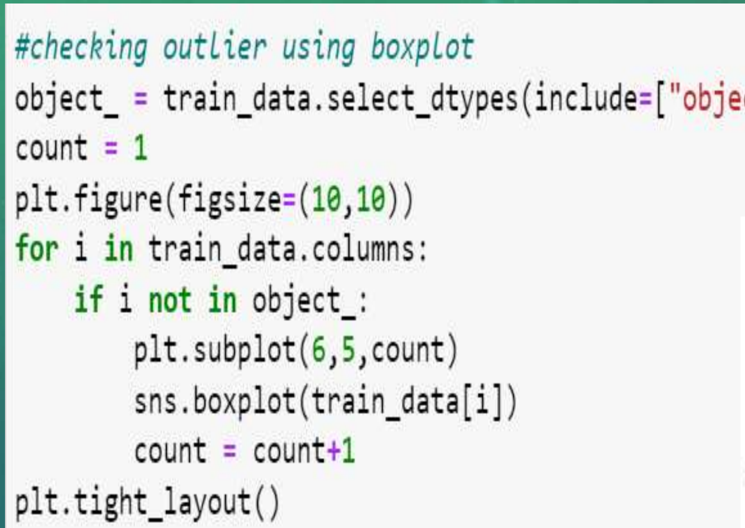
```
#computing null values of catogoric features with mode
for column in ["Attrition","BusinessTravel","Department","EducationField","Gender","JobRole","MaritalStatus","Over18","OverTime",
    train_data[column].fillna(train_data[column].mode()[0], inplace=True)
```



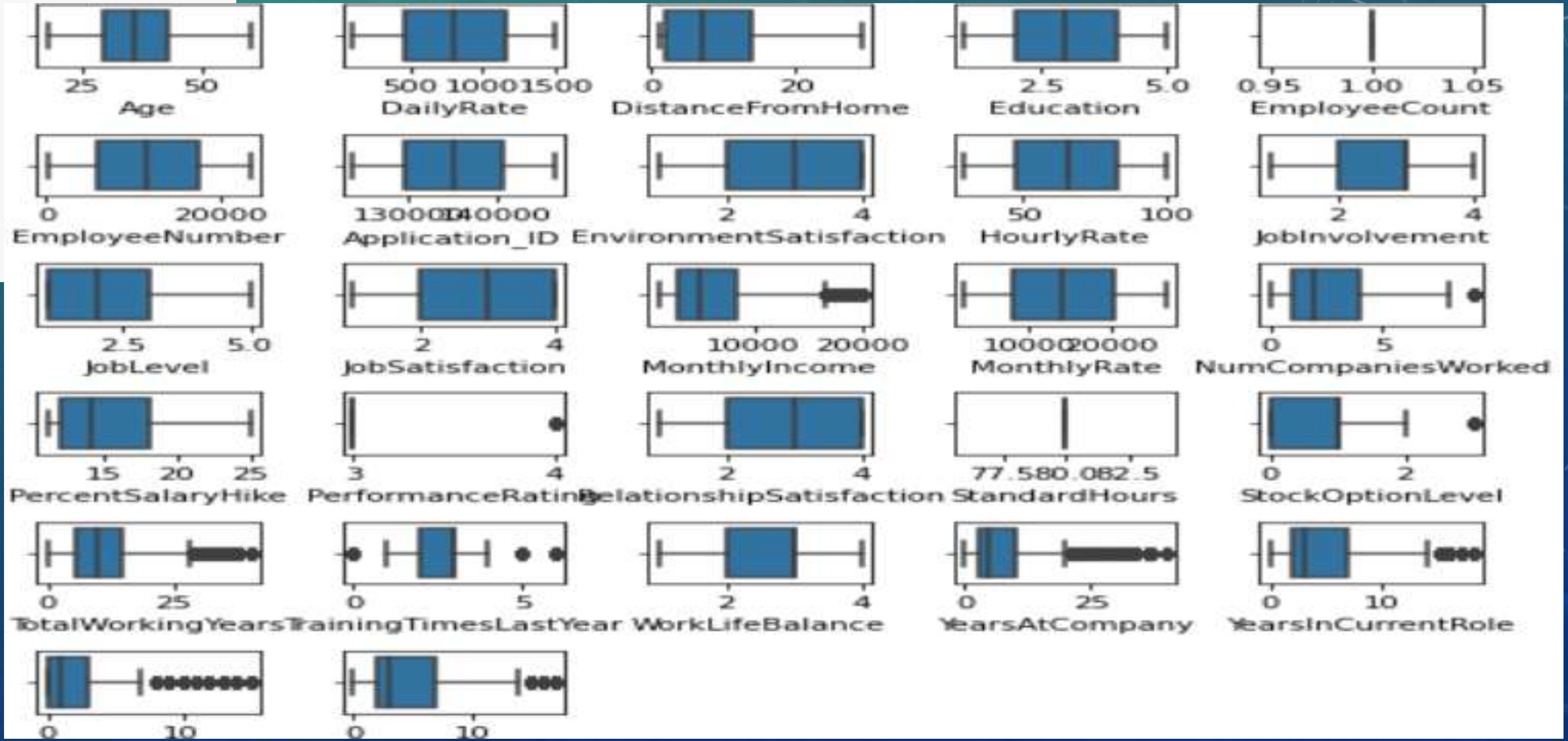
```
#now all unbelonging values deletd we can impute missing values with median
for column in ['DistanceFromHome','EmployeeCount','EmployeeNumber','Application_ID','HourlyRate','JobSatisfaction','MonthlyIncome
        train_data[column].fillna(train_data[column].median(), inplace=True)
```

❑ HANDLING OUTLIERS

- “ Box-plotting ” is done to check whether outliers are present or not.
- If found remove the outliers using IQR method.



```
#checking outlier using boxplot
object_ = train_data.select_dtypes(include=["object"]).columns
count = 1
plt.figure(figsize=(10,10))
for i in train_data.columns:
    if i not in object_:
        plt.subplot(6,5,count)
        sns.boxplot(train_data[i])
        count = count+1
plt.tight_layout()
```

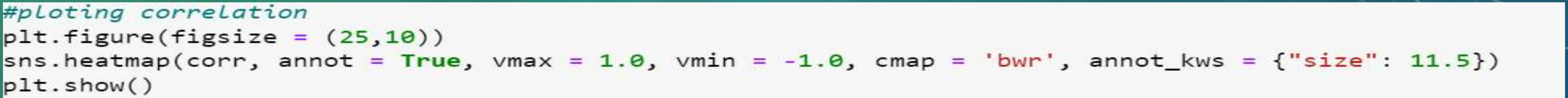


CHECKING THE CORRELATION BY PLOTTING HEAT MAP

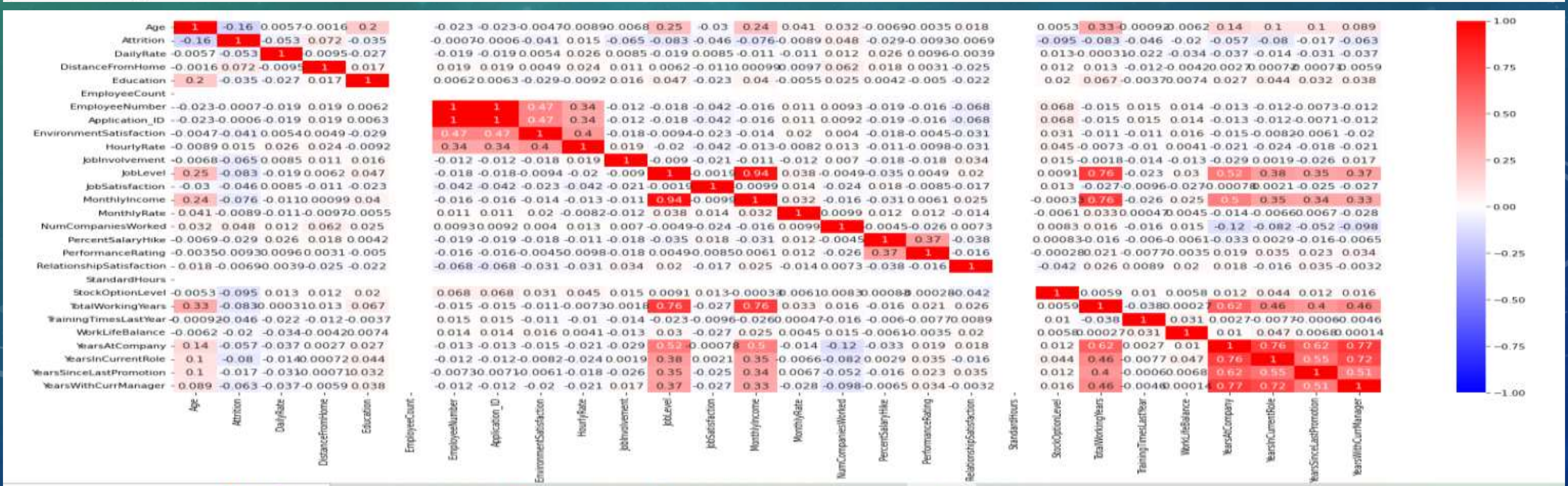
- Correlation is a statistical technique which determines how one variables moves/changes in relation with the other variable. It’s a bi-variant analysis measure which describes the association between different variables.

Usefulness of Correlation matrix :-

- If two variables are closely correlated, then we can predict one variable from the other.
- Correlation plays a vital role in locating the important variables on which other variables depend.
- Proper correlation analysis leads to better understanding of data.

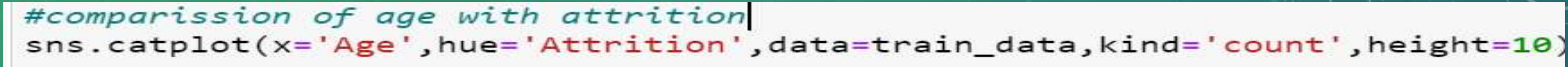


```
#ploting correlation
plt.figure(figsize = (25,10))
sns.heatmap(corr, annot = True, vmax = 1.0, vmin = -1.0, cmap = 'bwr', annot_kws = {"size": 11.5})
plt.show()
```

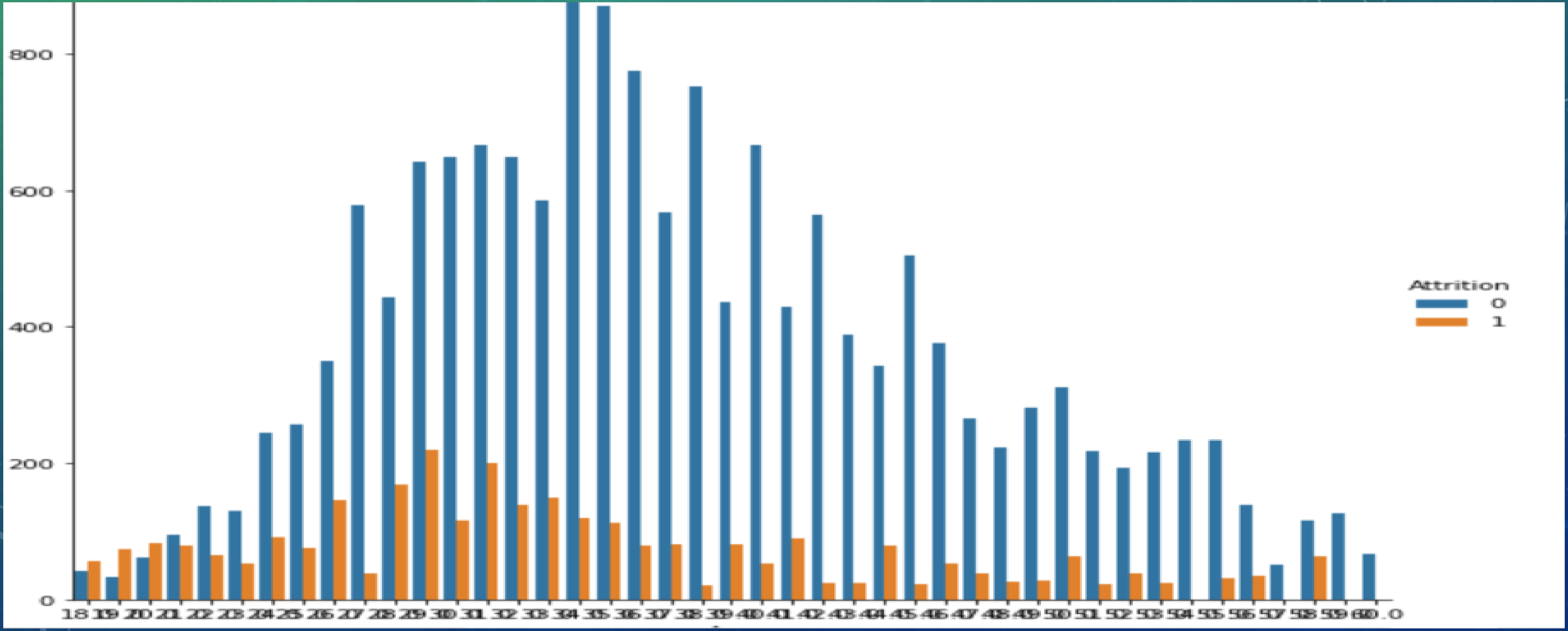


DATA VISUALIZATION(EDA)

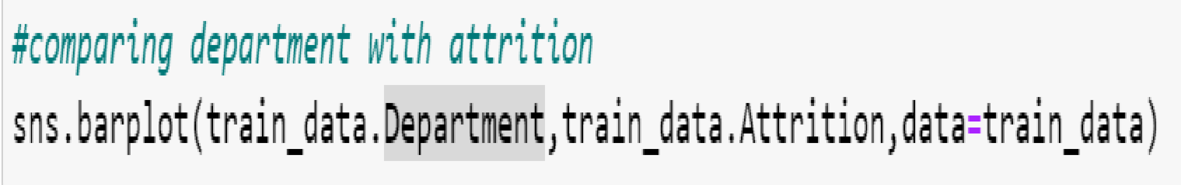
1. ATTRITION V/S “AGE”:



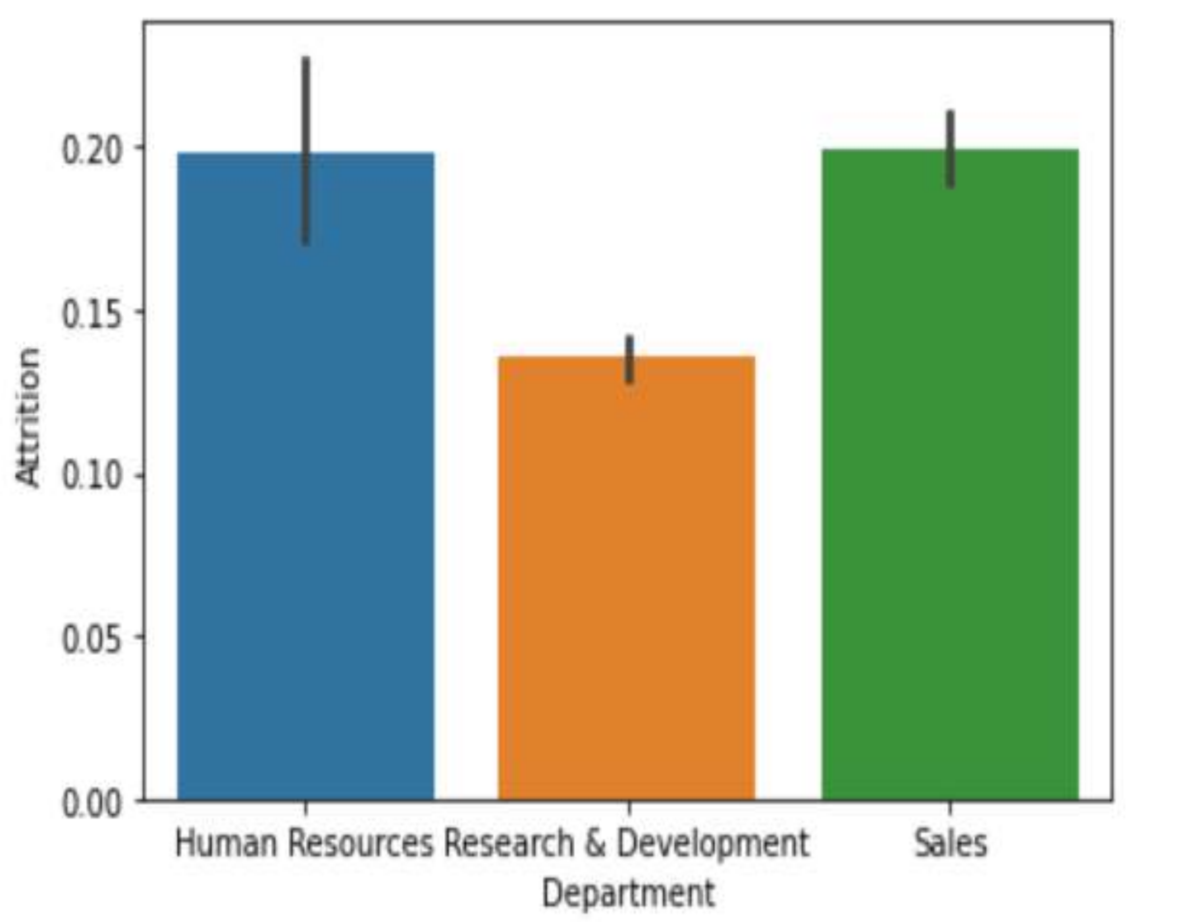
```
#comparission of age with attrition
sns.catplot(x='Age',hue='Attrition',data=train_data,kind='count',height=10)
```



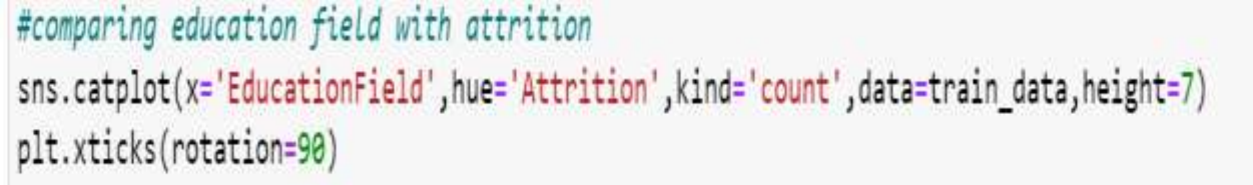
2. Attrition V/s “Department”:



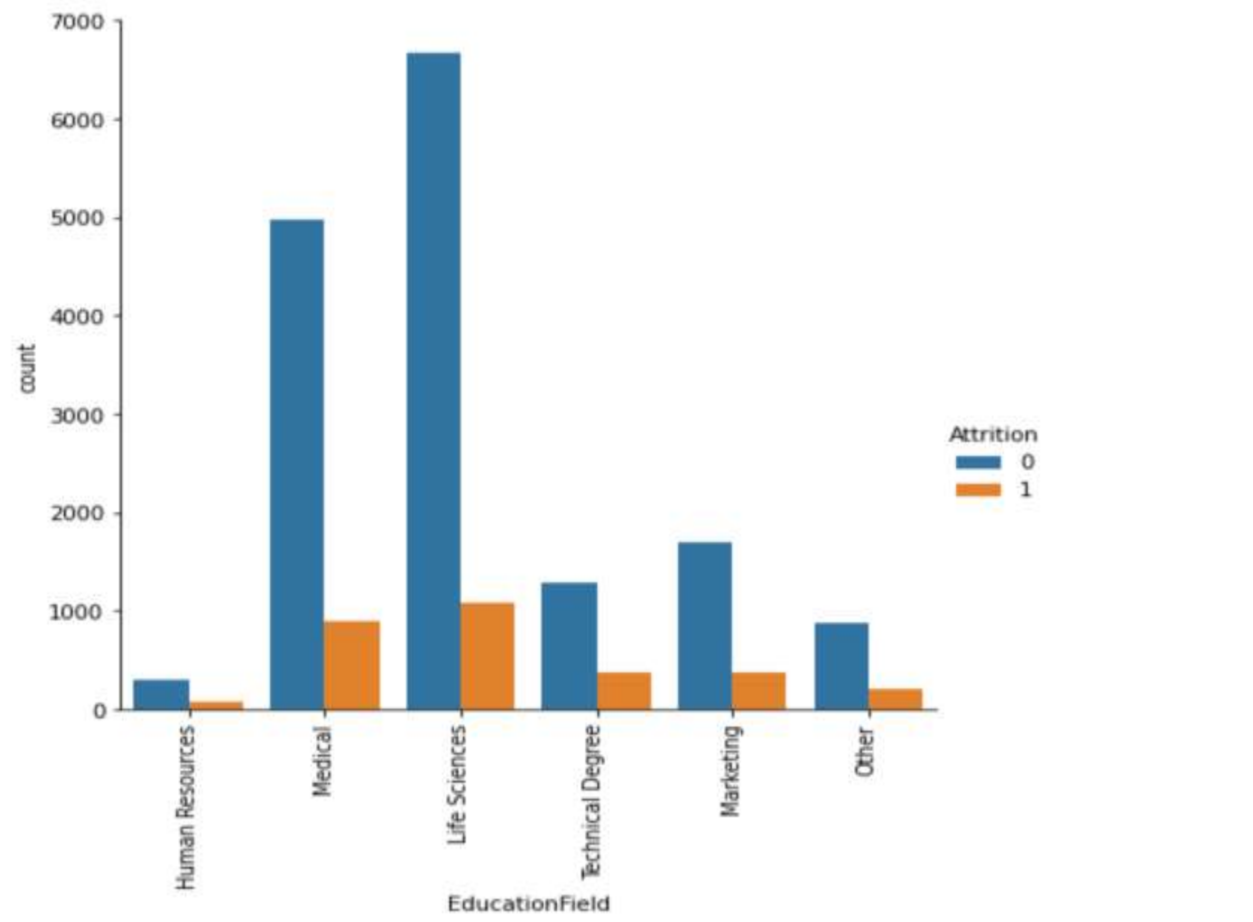
```
#comparing department with attrition
sns.barplot(train_data.Department,train_data.Attrition,data=train_data)
```



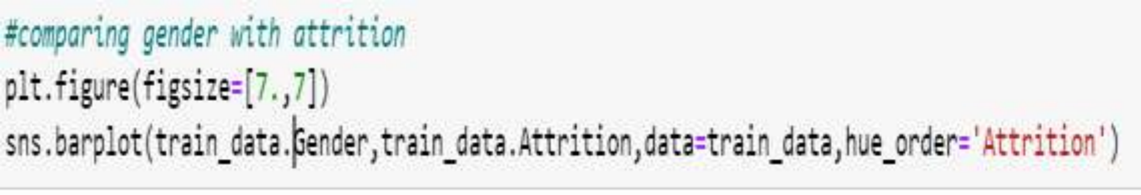
3. Attrition V/s “EducationField”:



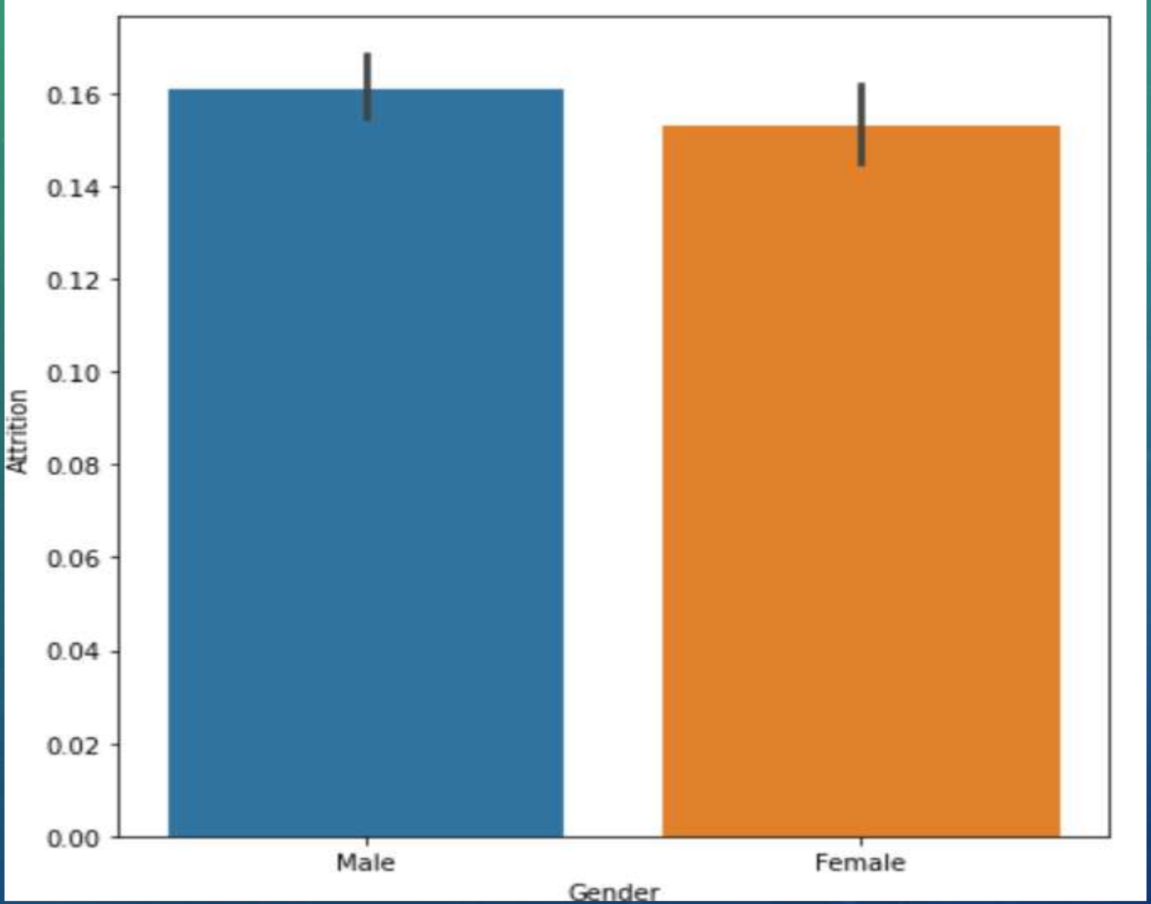
```
#comparing education field with attrition
sns.catplot(x='EducationField',hue='Attrition',kind='count',data=train_data,height=7)
plt.xticks(rotation=90)
```



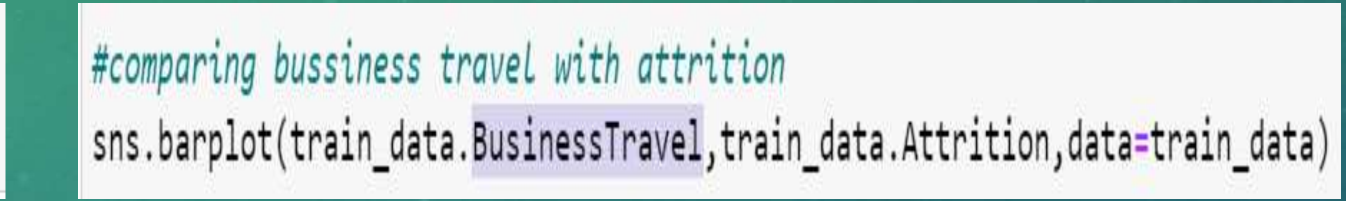
4. Attrition V/s “Gender”:



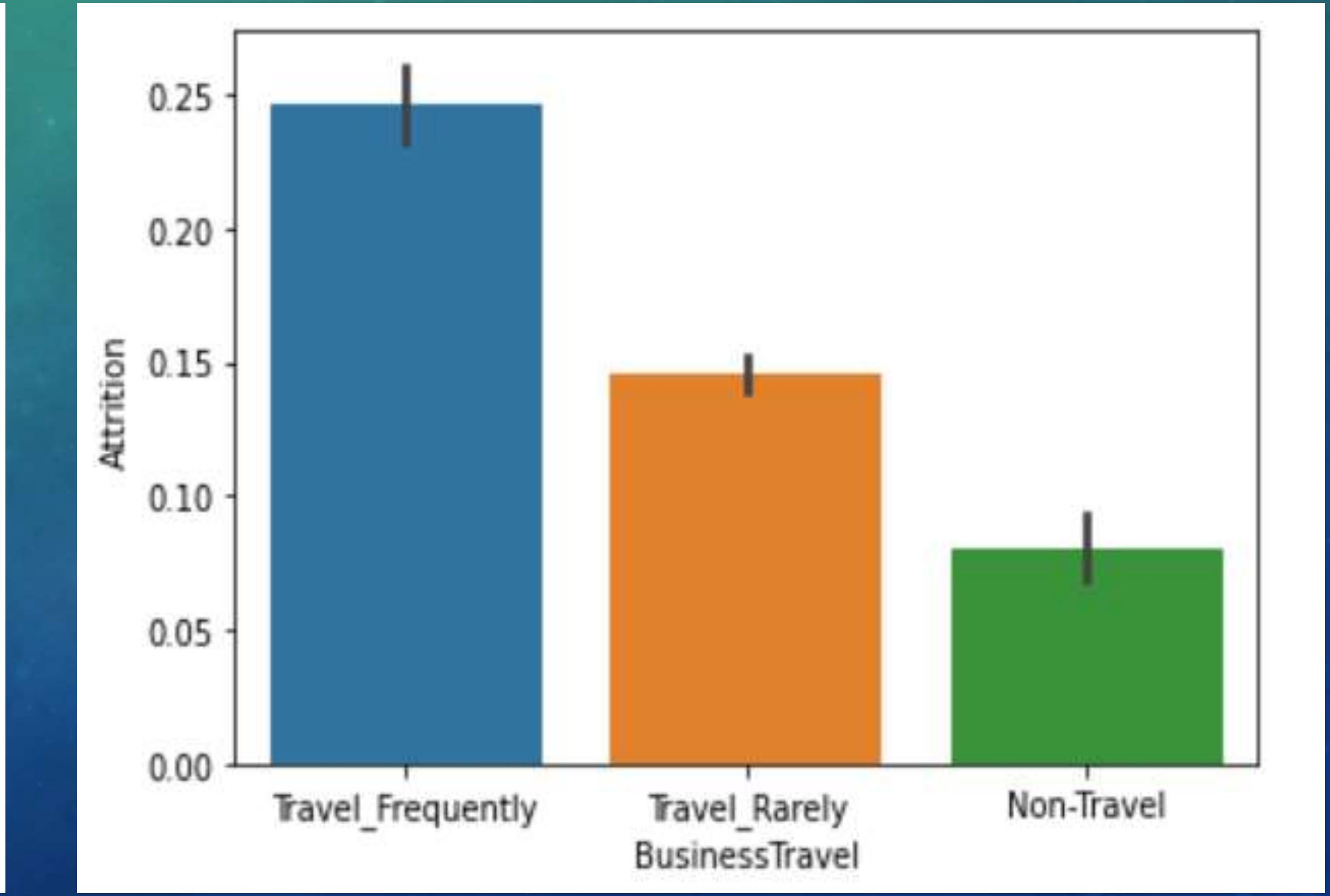
```
#comparing gender with attrition
plt.figure(figsize=[7.,7])
sns.barplot(train_data.Gender,train_data.Attrition,data=train_data,hue_order='Attrition')
```



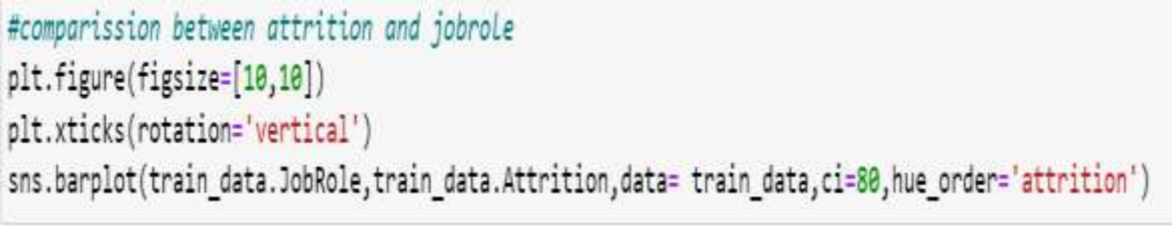
5. Attrition V/s “BusinessTravel”:



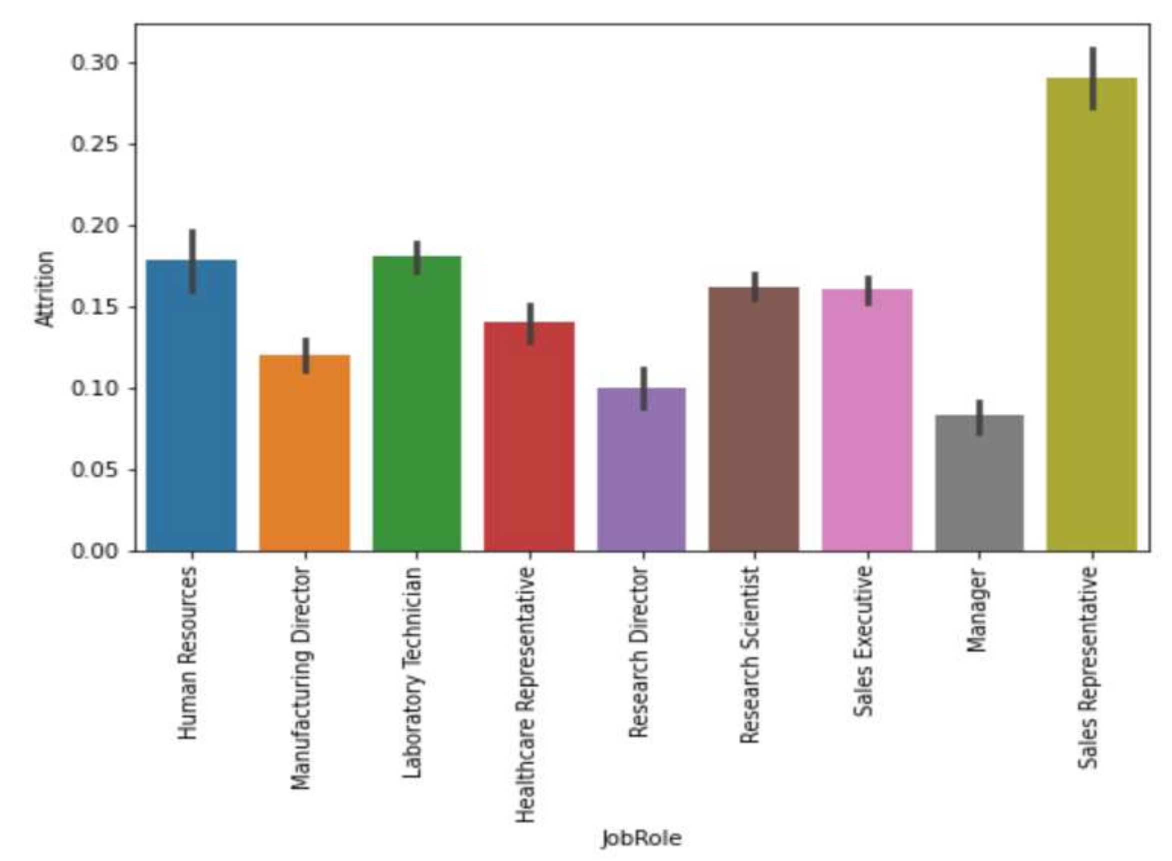
```
#comparing bussiness travel with attrition
sns.barplot(train_data.BusinessTravel,train_data.Attrition,data=train_data)
```



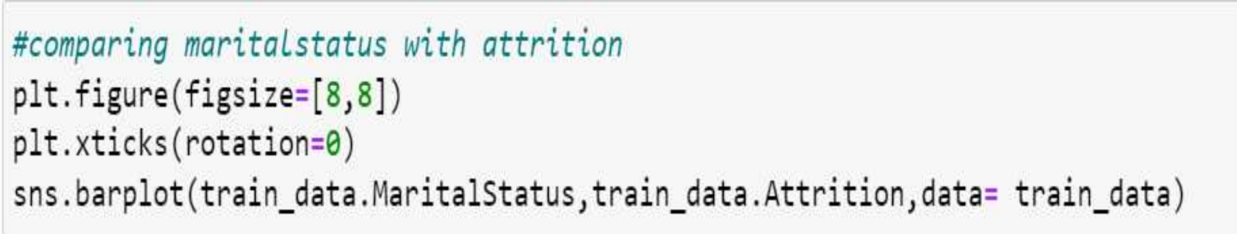
6. Attrition V/s “JobRole”:



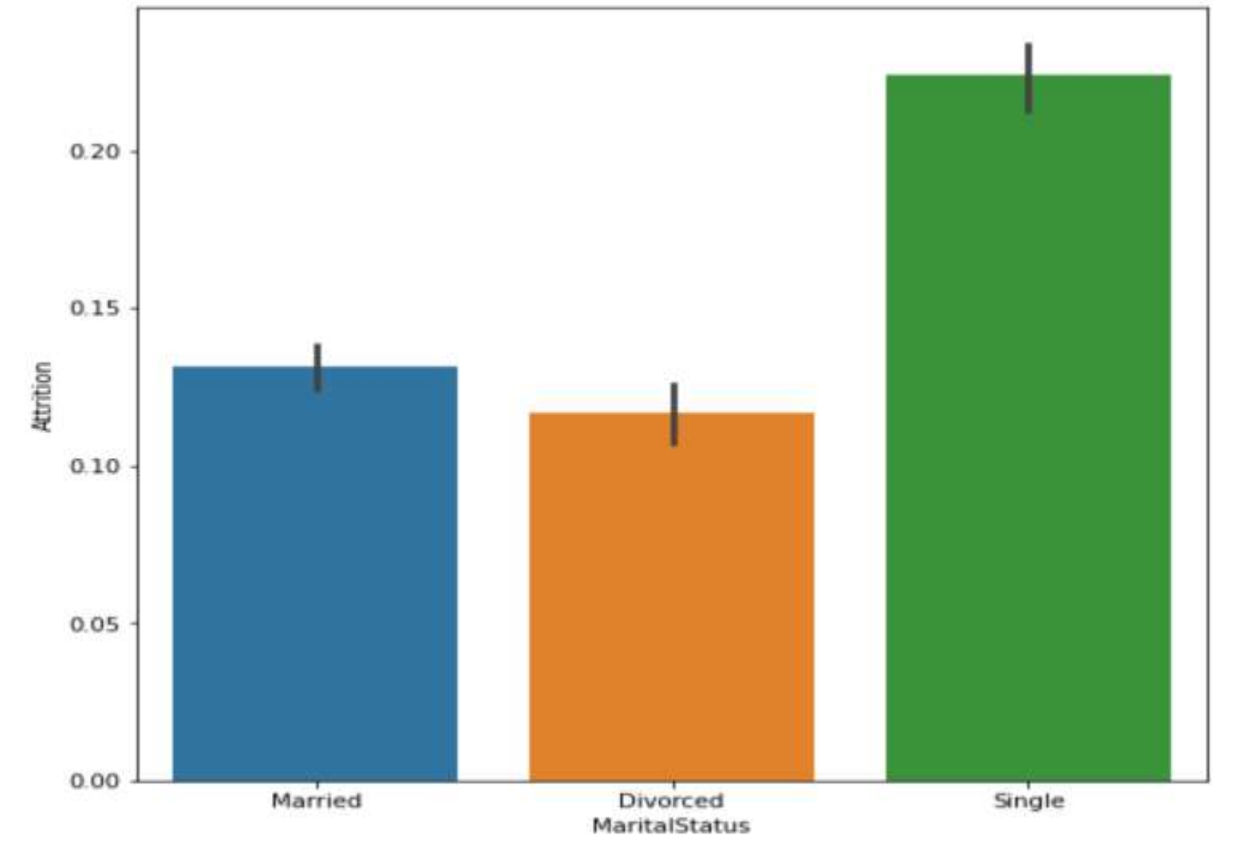
```
#comparission between attrition and jobrole
plt.figure(figsize=[10,10])
plt.xticks(rotation='vertical')
sns.barplot(train_data.JobRole,train_data.Attrition,data= train_data,ci=80,hue_order='attrition')
```



7.Attrition V/s “MaritalStatus”:



```
#comparing maritalstatus with attrition
plt.figure(figsize=[8,8])
plt.xticks(rotation=0)
sns.barplot(train_data.MaritalStatus,train_data.Attrition,data= train_data)
```



DATA PRE - PROCESSING

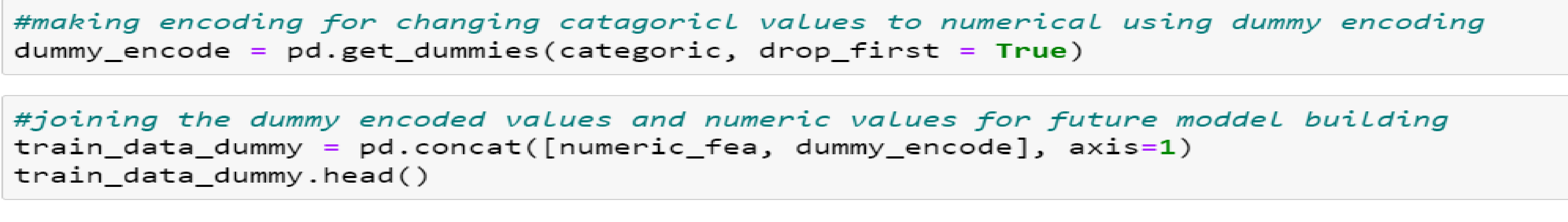
- Refers to data mining technique that transforms raw data into an understandable format
- Useful in making the data ready for analysis

❑ **Steps Involved** :

- Taking care of missing data and dropping non-relevant features
- Feature extraction
- Converting categorical features into numeric form
- Binarization of the converted categorical features
- Feature scaling
- Understanding correlation of features with each other
- Splitting data into training and test data sets

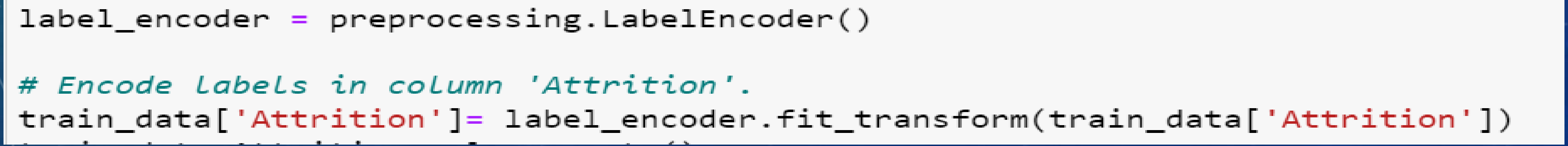
❑ ENCODING FOR CATEGORICAL FEATURES

- **One Hot Encoder:**
- **Label Encoder:**
  - It is used to perform "binarization" of the categorical features and include it as a feature to train the model.
  - It takes a column which has categorical data that has been encoded, and then splits the column into multiple columns.
  - The numbers are replaced by 1's and 0's, depending on which column has what value.



```
#making encoding for changing catagoricl values to numerical using dummy encoding
dummy_encode = pd.get_dummies(categoric, drop_first = True)
```

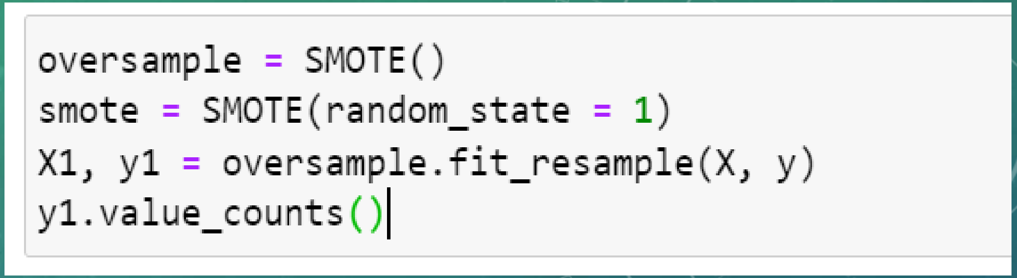
```
#joining the dummy encoded values and numeric values for future moddel building
train_data_dummy = pd.concat([numeric_fea, dummy_encode], axis=1)
train_data_dummy.head()
```



```
label_encoder = preprocessing.LabelEncoder()

# Encode labels in column 'Attrition'.
train_data['Attrition']= label_encoder.fit_transform(train_data['Attrition'])
```

HANDLING IMBALANCE DATASET USING SMOTE

SMOTE is an oversampling technique where the synthetic samples are generated for the minority class. This algorithm helps to overcome the overfitting problem posed by random oversampling.



```
oversample = SMOTE()
smote = SMOTE(random_state = 1)
X1, y1 = oversample.fit_resample(X, y)
y1.value_counts()
```

MODEL BUILDING AND TESTING

The process of modeling means training a machine learning algorithm to predict the labels from the features, tuning it for the business need, and validating it on holdout data.

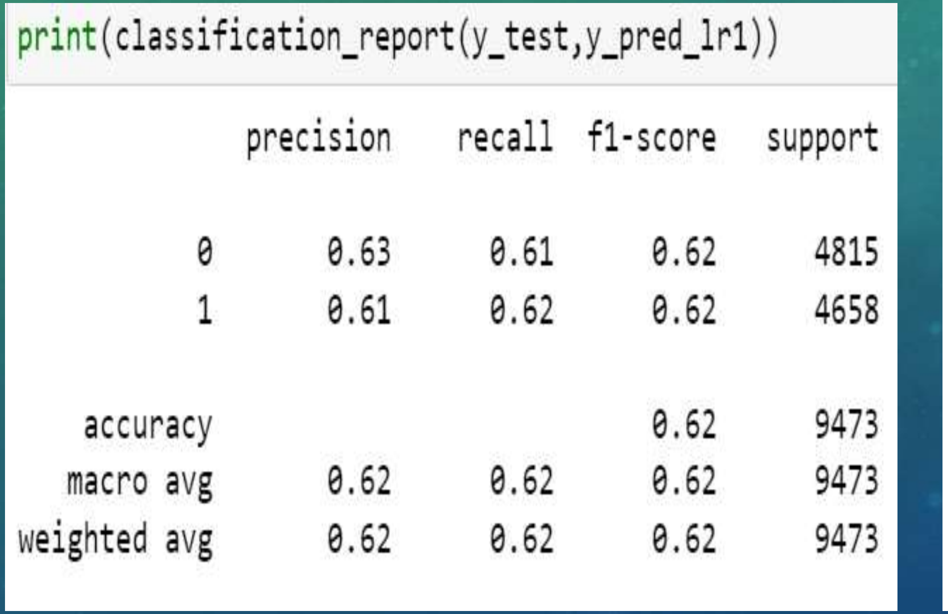
MODELS USED FOR EMPLOYEE ATTRITION

- Logistic Regression
- Random Forest
- Decision Tree
- Support vector machine
- K-Nearest Neighbour
- XG Boost
- Random Forest Grid Search

❑ LOGISTIC REGRESSION

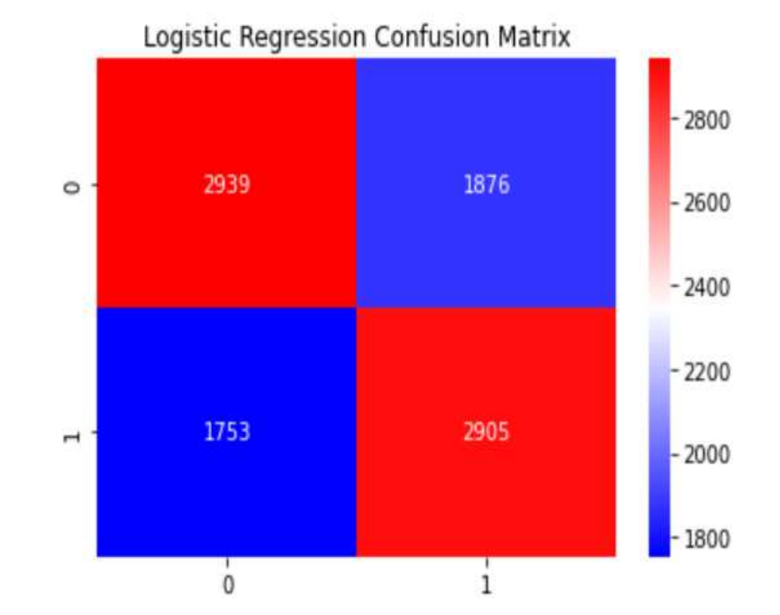
- Logistic Regression is one of the most basic and widely used machine learning algorithms for solving a classification problem.
- It is a method used to predict a dependent variable (Y), given an independent variable (X), given that the dependent variable is categorical.

➢ Testing model:

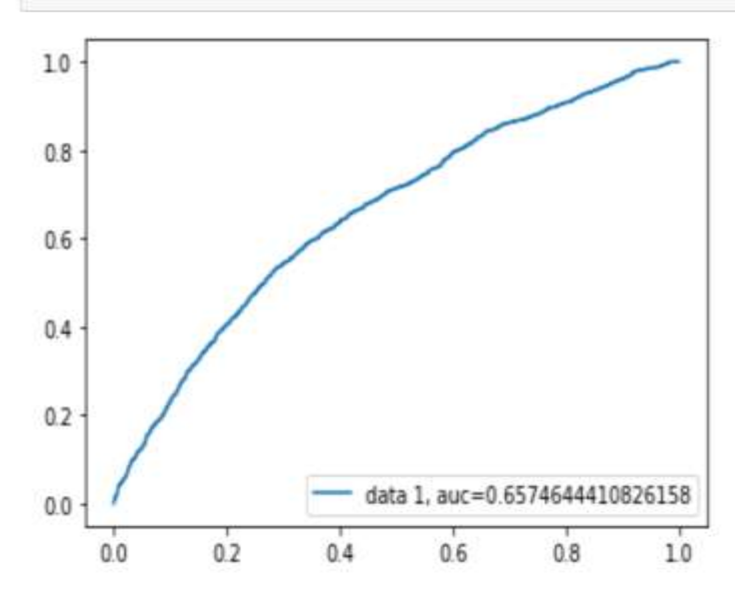


```
print(classification_report(y_test,y_pred_lr1))

              precision    recall  f1-score   support

           0       0.63      0.61      0.62      4815
           1       0.61      0.62      0.62      4658

    accuracy                           0.62      9473
   macro avg       0.62      0.62      0.62      9473
weighted avg       0.62      0.62      0.62      9473
```

➢ Confusion matrix:



➢ AUC-ROC Curve:

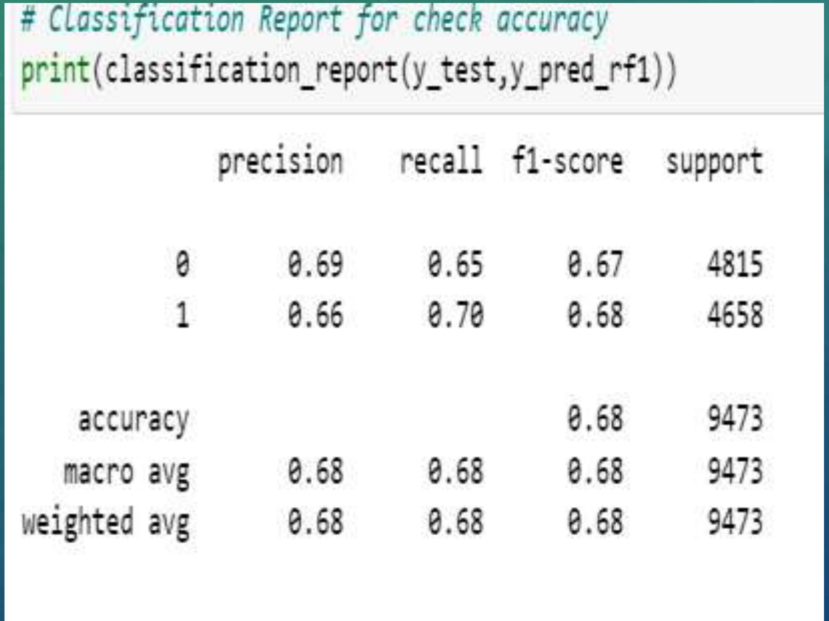


Using Logistic Regression algorithm, we got the accuracy score of 62% and roc-auc score of 0.65

❑ RANDOM FOREST

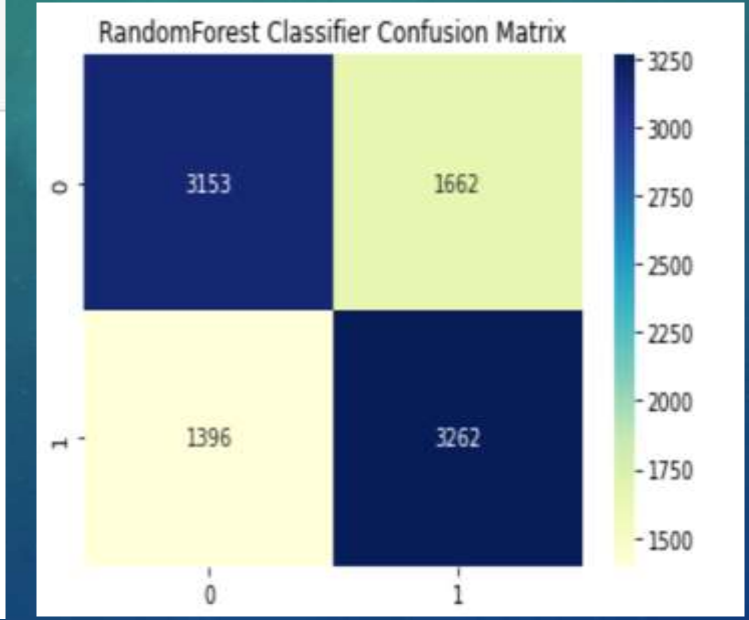
- Random Forest is a supervised learning algorithm.
- It creates a forest and makes it random based on bagging technique.
- In Random Forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node.

➢ Testing model:

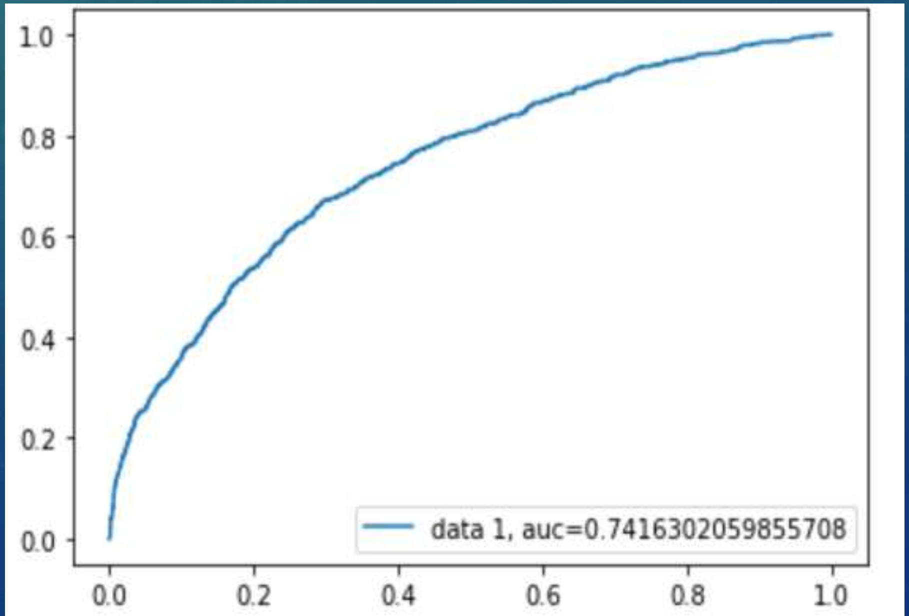


```
# Classification Report for check accuracy
print(classification_report(y_test,y_pred_rf1))

              precision    recall  f1-score   support

           0       0.69      0.65      0.67      4815
           1       0.66      0.70      0.68      4658

    accuracy                           0.68      9473
   macro avg       0.68      0.68      0.68      9473
weighted avg       0.68      0.68      0.68      9473
```

➢ Confusion matrix:



➢ AUC-ROC Curve:

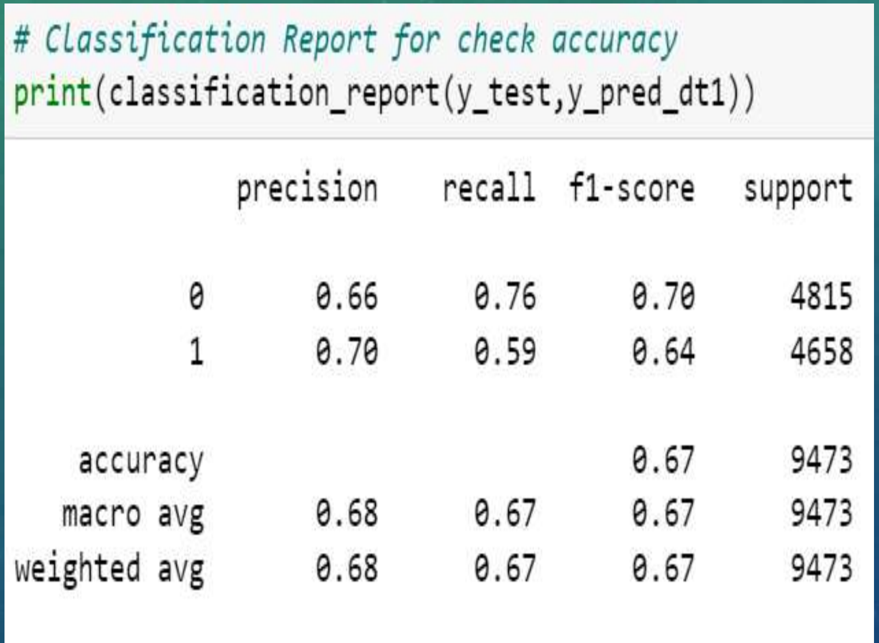


Using Random Forest algorithm, we got the accuracy score of 68% and roc-auc score of 0.74

❑ DECISION TREE

- Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems.
- It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

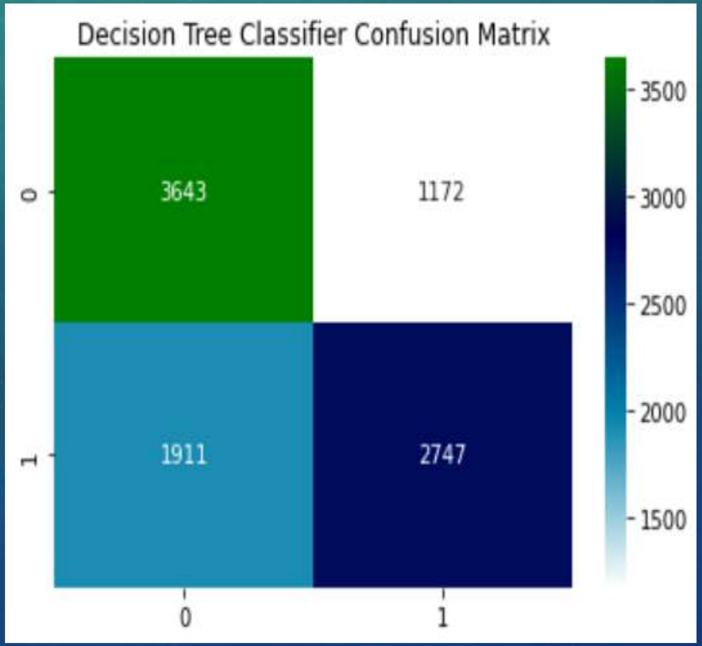
➢ **Testing model**:



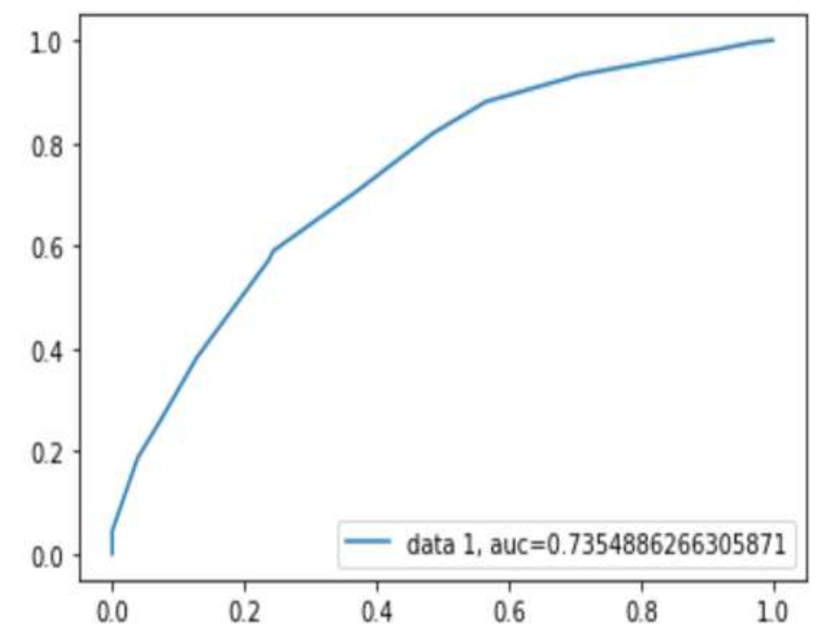
```
# Classification Report for check accuracy
print(classification_report(y_test,y_pred_dt1))
```

```
              precision    recall  f1-score   support

           0       0.66      0.76      0.70      4815
           1       0.70      0.59      0.64      4658

    accuracy                           0.67      9473
   macro avg       0.68      0.67      0.67      9473
weighted avg       0.68      0.67      0.67      9473
```

➢ **Confusion matrix**:



➢ **AUC-ROC Curve**:

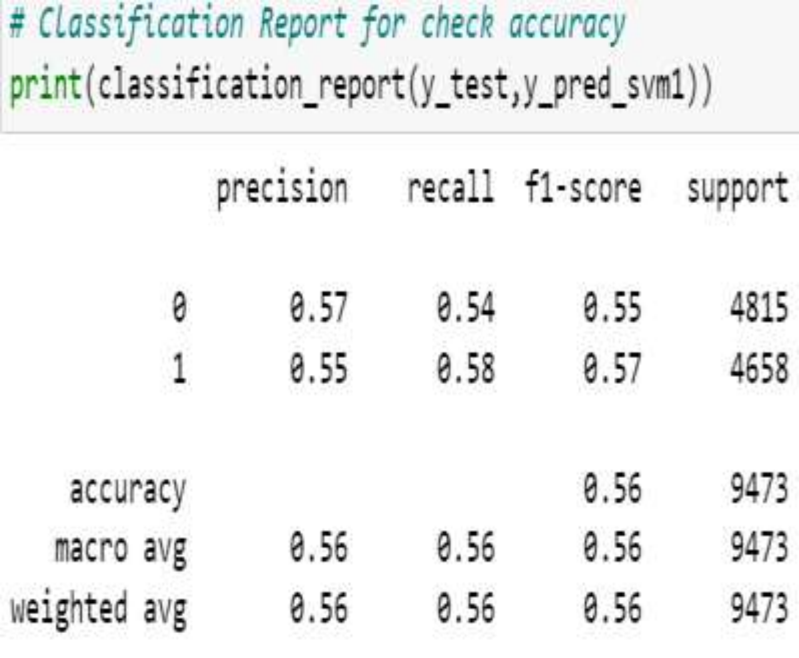


Using Decision Tree, we got the accuracy score of 67% and roc-auc score of 0.73

❑ SUPPORT VECTOR MACHINE

- SVM is a supervised machine learning algorithm used for both regression and classification problems.
- Objective is to find a hyperplane in an N -dimensional space.

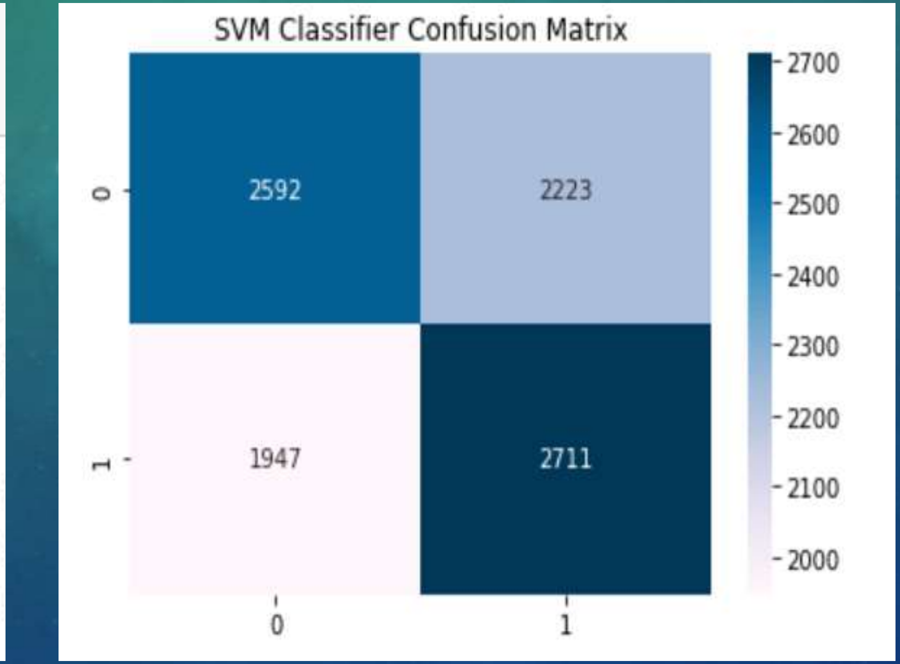
➢ Testing model:



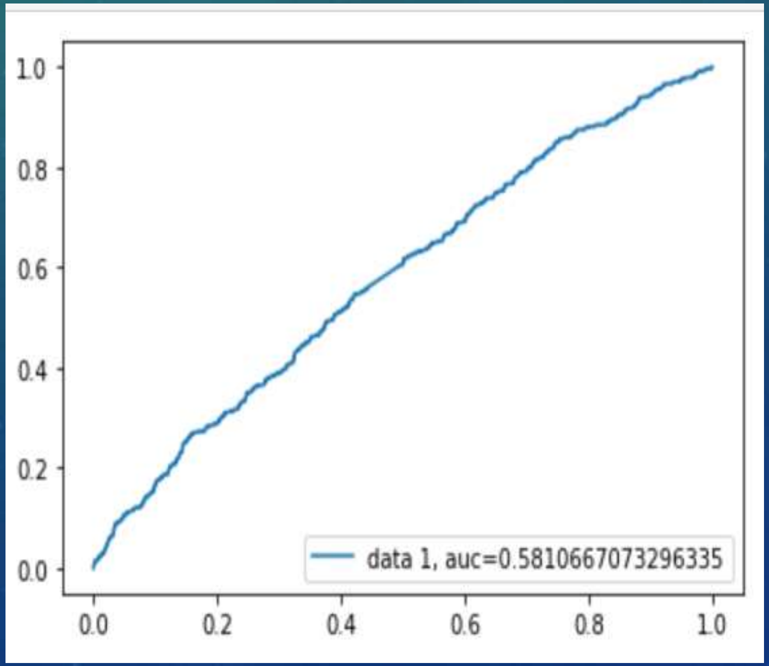
```
# Classification Report for check accuracy
print(classification_report(y_test,y_pred_svm1))
```

```
              precision    recall  f1-score   support

           0       0.57      0.54      0.55      4815
           1       0.55      0.58      0.57      4658

    accuracy                           0.56      9473
   macro avg       0.56      0.56      0.56      9473
weighted avg       0.56      0.56      0.56      9473
```

➢ Confusion matrix:



➢ AUC-ROC Curve:

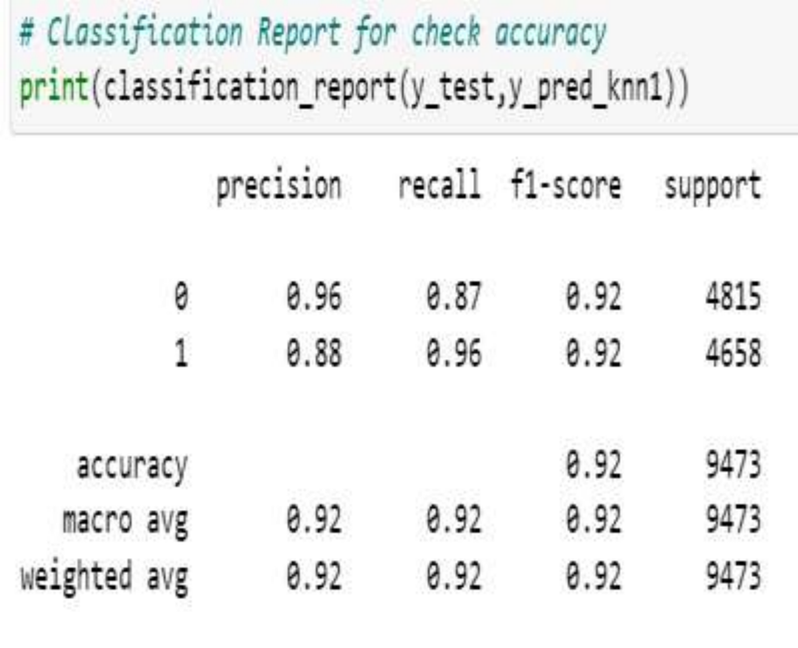


Using Support Vector Machine , we got the accuracy score of 56% and roc-auc score of 0.58

❑ K-NEAREST NEIGHBOUR

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories

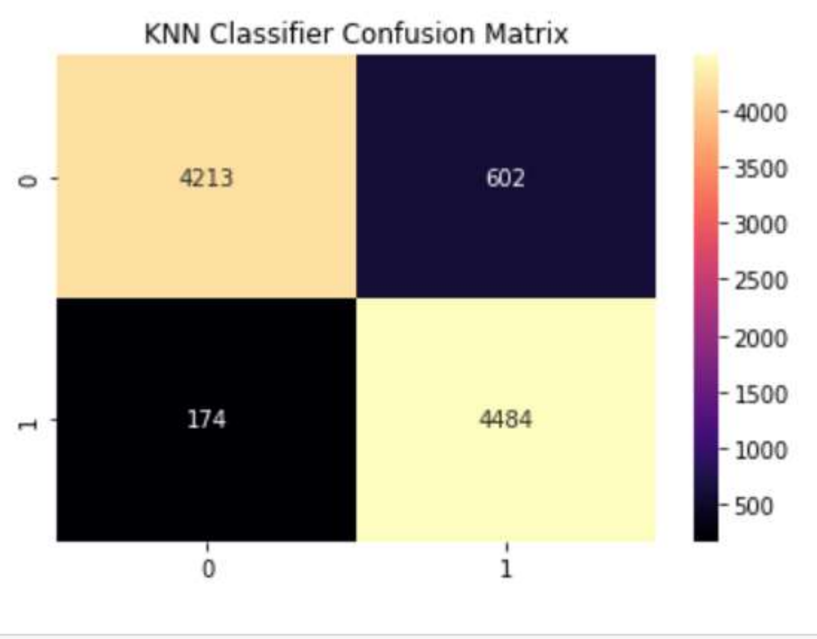
➢ Testing model:



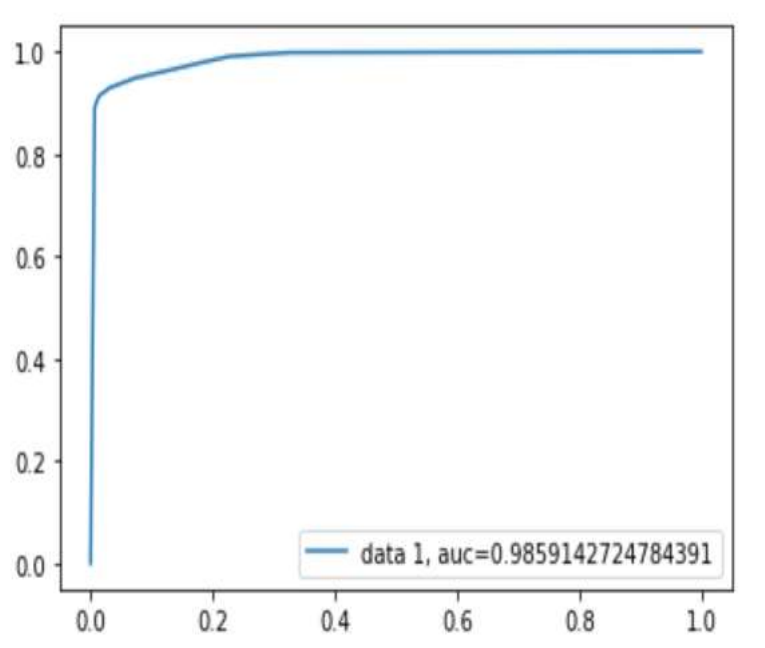
```
# Classification Report for check accuracy
print(classification_report(y_test,y_pred_knn1))
```

```
              precision    recall  f1-score   support

           0       0.96      0.87      0.92      4815
           1       0.88      0.96      0.92      4658

    accuracy                           0.92      9473
   macro avg       0.92      0.92      0.92      9473
weighted avg       0.92      0.92      0.92      9473
```

➢ Confusion matrix:



➢ AUC-ROC Curve:

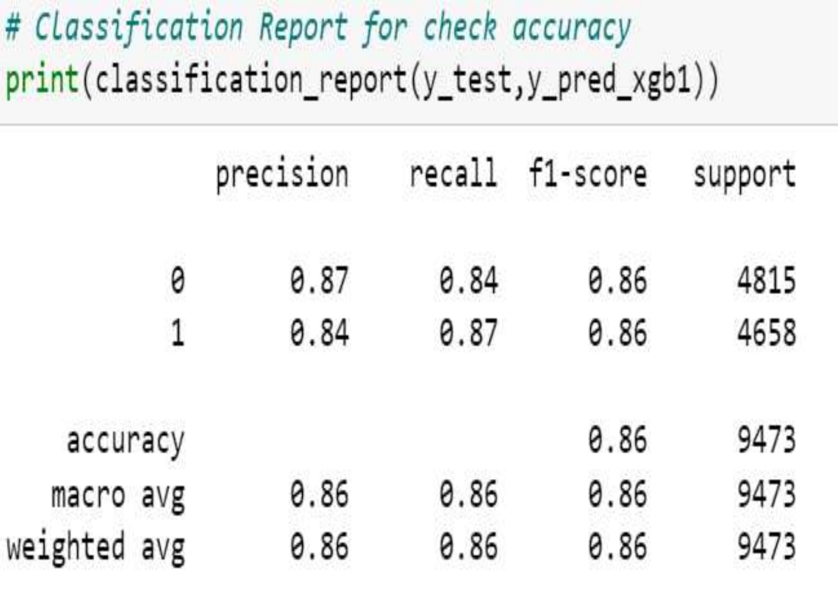


Using K-Nearest Neighbour , we got the accuracy score of 92% and roc-auc score of 0.98. i.e it shows model is in over fit or under fit condition .

❑ XG BOOST

- XG Boost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework.
- XG Boost belongs to a family of boosting algorithms that convert weak learners into strong learners.
- It is a sequential process, i.e., trees are grown using the information from a previously grown tree one after the other, iteratively, the errors of the previous model are corrected by the next predictor.

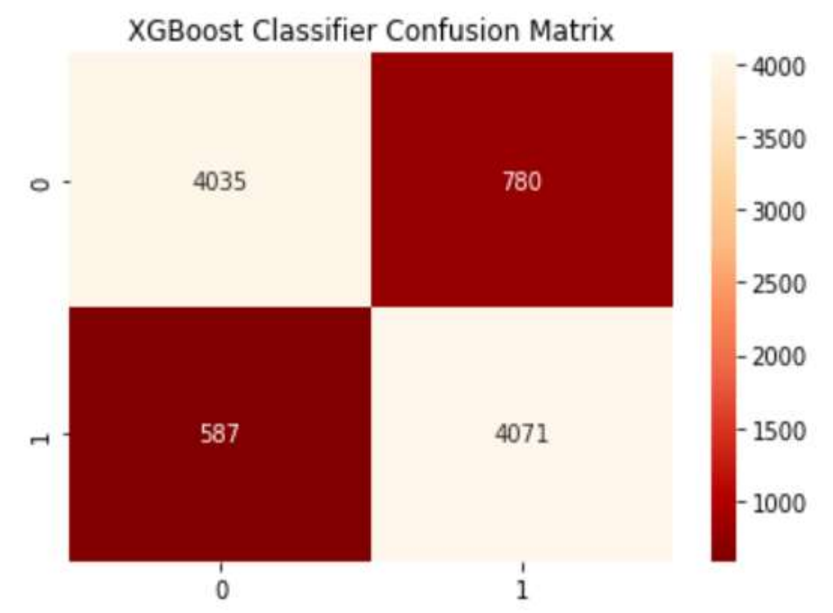
➢ Testing model:



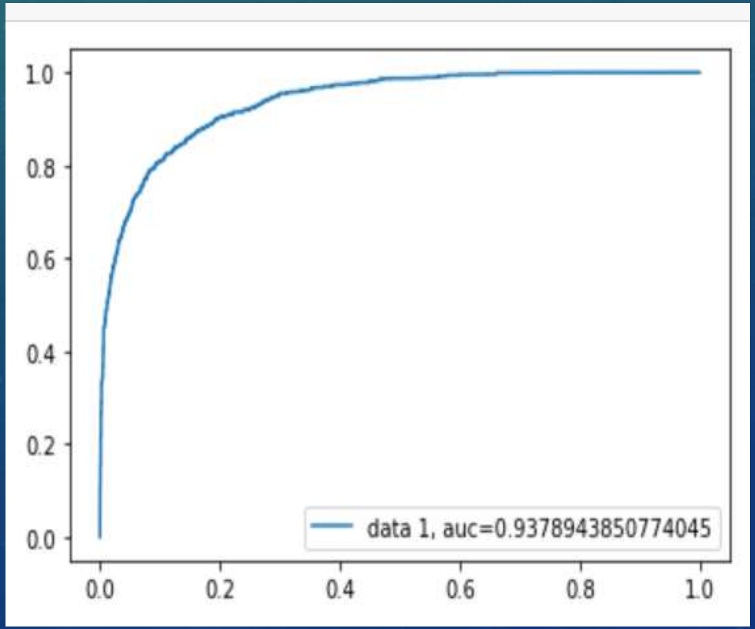
```
# Classification Report for check accuracy
print(classification_report(y_test,y_pred_xgb1))
```

```
              precision    recall  f1-score   support

           0       0.87      0.84      0.86      4815
           1       0.84      0.87      0.86      4658

    accuracy                           0.86      9473
   macro avg       0.86      0.86      0.86      9473
weighted avg       0.86      0.86      0.86      9473
```

➢ Confusion matrix:



➢ AUC-ROC Curve:

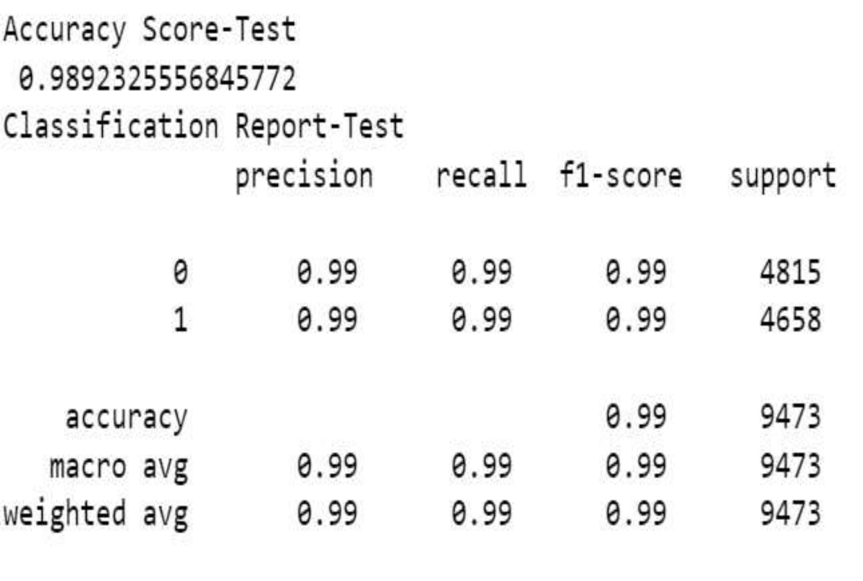


Using XG Boost , we got the accuracy score of 86% and roc-auc score of 0.93. i.e it shows model is in over fit or under fit condition .

❑ RANDOM FOREST WITH HYPER-PARAMETER TUNING

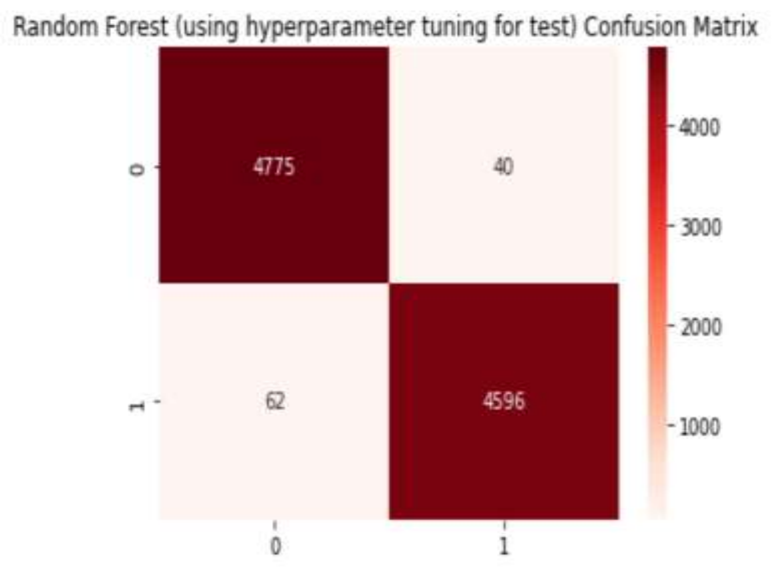
- **Hyperparameter tuning** (or hyperparameter optimization) is the process of determining the right combination of hyperparameters that maximizes the model performance.
- Here we are using **GridSearchCV** approach in the model.
- In the gridsearch method, we create a grid of possible values for hyperparameters. Each iteration tries a combination of hyperparameters in a specific order
- It fits the model on each and every combination of hyperparameters possible and records the model performance. Finally, it returns the best model with the best hyperparameters.

➢ Testing model:

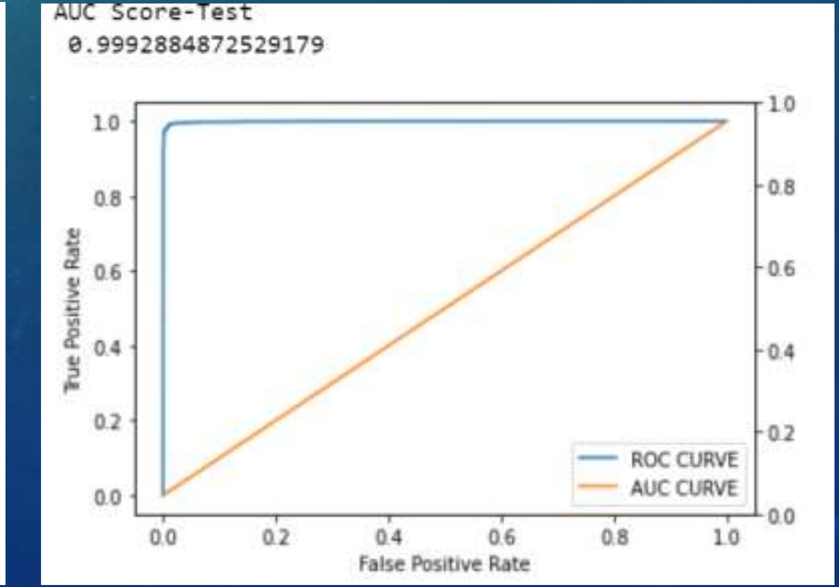


```
Accuracy Score-Test
 0.9892325556845772
Classification Report-Test
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      4815
           1       0.99      0.99      0.99      4658

    accuracy                           0.99      9473
   macro avg       0.99      0.99      0.99      9473
weighted avg       0.99      0.99      0.99      9473
```

➢ Confusion matrix:



➢ AUC-ROC Curve:



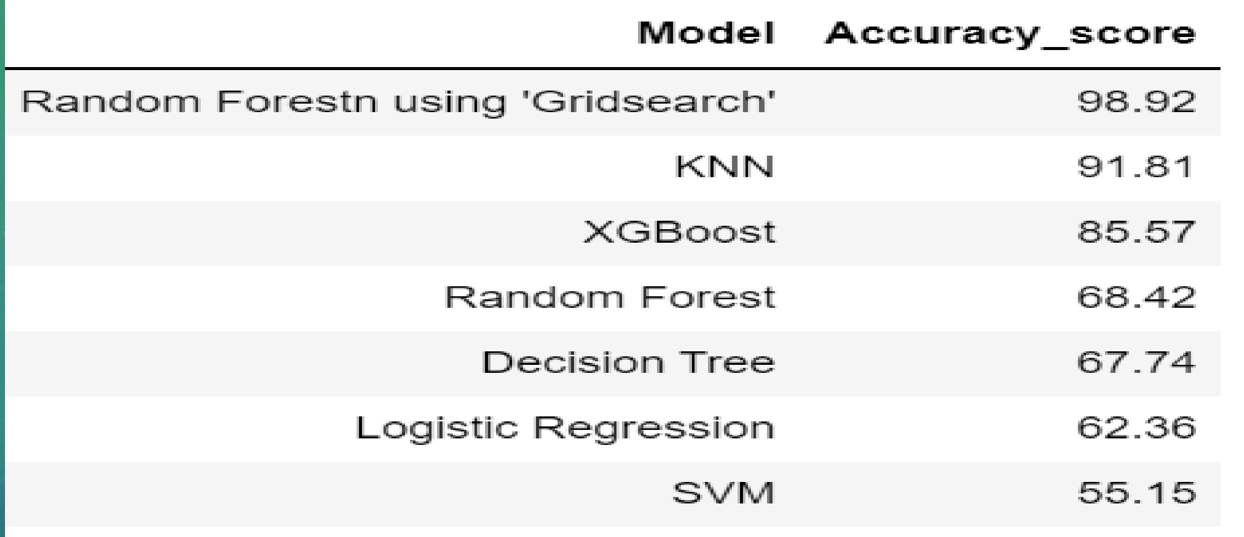
```
AUC Score-Test
 0.9992884872529179
```

Using Gridsearch , we got the accuracy score of 98% and roc-auc score of 0.99. it shows model cannot be overfit or underfit

❑ COMPARISON OF MODELS

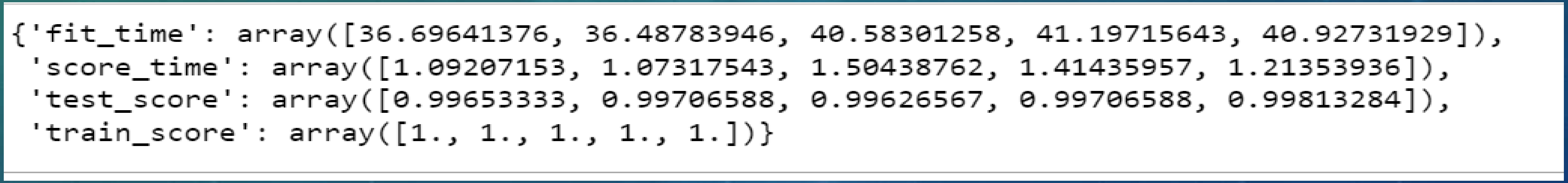
The goal of comparing machine learning algorithms:

- Better performance
- Longer lifetime
- Easier retraining
- Speedy production



Model	Accuracy_score
Random Forestn using 'Gridsearch'	98.92
KNN	91.81
XGBoost	85.57
Random Forest	68.42
Decision Tree	67.74
Logistic Regression	62.36
SVM	55.15

❑ CROSS VALIDATION



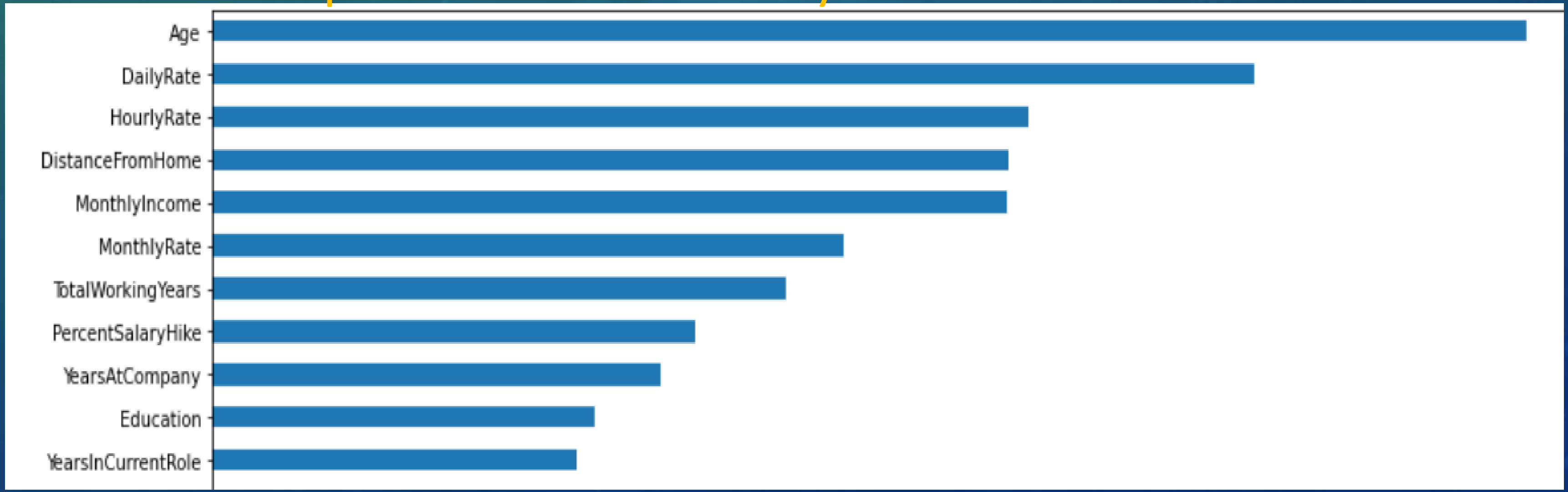
```
{'fit_time': array([36.69641376, 36.48783946, 40.58301258, 41.19715643, 40.92731929]),
 'score_time': array([1.09207153, 1.07317543, 1.50438762, 1.41435957, 1.21353936]),
 'test_score': array([0.99653333, 0.99706588, 0.99626567, 0.99706588, 0.99813284]),
 'train_score': array([1., 1., 1., 1., 1.])}
```

- By cross validation we got to know Random forest with grid search model is not to be overfitted or underfitted it giving same score to both test and train.
- It can be observed by the table that Random forest with grid search out performs all other models.
- Hence, based on these results we can conclude that, Random forest with grid search will be the best model to predict future Employee Attrition for this company.

❑ KEY FINDINGS

- The top factor for employee attrition in this hypothetical organization seems to be **monetary**, as ‘OverTime’ and ‘MonthlyIncome’ emerged at the top.
- The next important factor seems to be **personal relationships** with fellow workers, where current manager and job role could be the main contributing reasons for attrition.
- The strongest positive correlations with the target features are:
- Distance from home, Job satisfaction, marital status, overtime and business travel
- The strongest negative correlations with the target features are:
- Performance Rating and Training times last year
- **Employee engagement** is a critical satisfaction factor, and the organization should keep employees constantly involved and motivated.

Top Features Derived By Random Forest Classifier



❑ RECOMMENDATIONS

- Transportation should be provided to employees living in the same area, or else transportation allowance should be provided.
- Plan and allocate projects in such a way to avoid the use of overtime
- Employees who hit their two-year anniversary should be identified as potentially having a higher-risk of leaving.
- Gather information on industry benchmarks to determine if the company is providing competitive wages.

❑ CONCLUSION

- HR Analytics is gaining traction in organisations that embrace digital transformation.
- The scope has expanded from analytics of employee work performance to providing insights so that decisive improvements can be made to organisational processes.
- While some level of attrition is inevitable, it should be kept at the minimal possible level.
- We also want to make some suggestions to the company through this research, hoping that they will care more about their employees and improve their job satisfaction.
- Simultaneously, they must pay more attention to human resources employees because they have very low job satisfaction.
- Besides, the company should allow employees to have enough time to rest and spend time with their families

THANK YOU