Attrition prediction

Project 4

Members:
Dale Munroe
Nick Taheri
Jose Sandoval

Overview

Dataset

Data Source:
https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset

Original Source Format: *csv "imported into postgresql"*

Dependent Variable : *Attrition* - (Boolean Y/N)

Independent Variable(s): *35* (e.g. basic info, work Info, satisfaction, salary and time related)

Data exploration & cleaning.

Data display & null values check

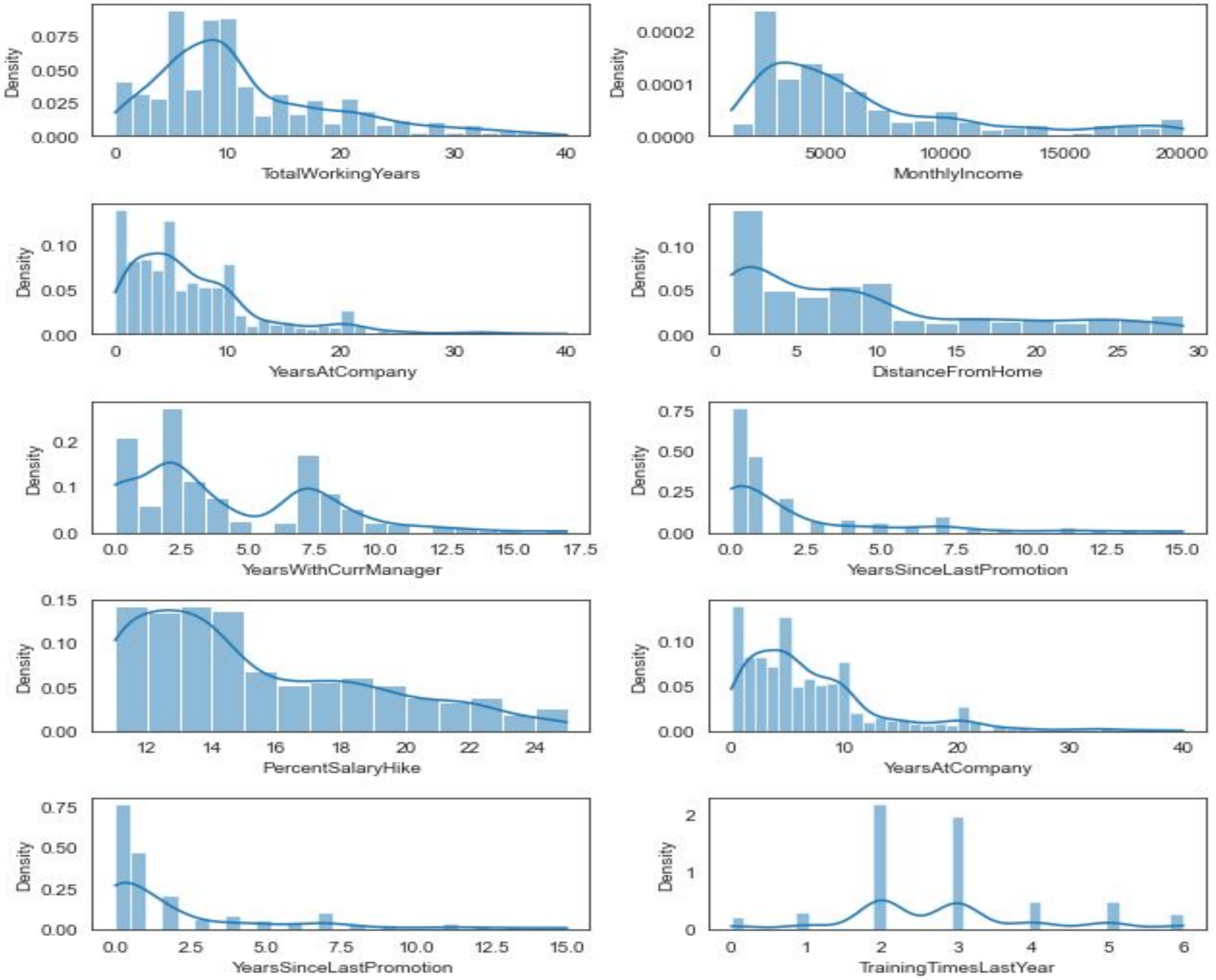
```
#Create DataFrame from SQL table
pg_connection = f'{protocol}://{username}:{password}@{host}:{port}/{database_name}'
conn = pg.connect(pg_connection)
raw_ibm_df = psql.read_sql('SELECT * FROM ibm_employee_data', conn)

# Display all the DataFrame Columns
pd.options.display.max_columns = None
display(raw_ibm_df)
```

✓ 0.3s

```
# check missing data for each feature
df[df.isnull().any(axis=1)]
```

	age	attrition	businesstravel	dailyrate	department	distancefromhome	education	educationfield	employeecount	employeenumber	environmentsatisfaction	gender	hourlyrate	jobinvolvement	joblev
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	2	Female	94	3	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	3	Male	61	2	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	4	Male	92	2	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	4	Female	56	3	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	1	Male	40	3	

Data Distribution



Insignificant Data

Columns with only 1 unique value & the employee # was removed

```
Age                           43
Attrition                      2
BusinessTravel                 3
DailyRate                    886
Department                     3
DistanceFromHome              29
Education                      5
EducationField                 6
EmployeeCount                  1
EmployeeNumber              1470
EnvironmentSatisfaction        4
Gender                         2
HourlyRate                    71
JobInvolvement                 4
JobLevel                       5
JobRole                        9
JobSatisfaction                4
MaritalStatus                  3
MonthlyIncome               1349
MonthlyRate                 1427
NumCompaniesWorked            10
Over18                         1
OverTime                       2
PercentSalaryHike             15
PerformanceRating              2
RelationshipSatisfaction       4
StandardHours                  1
StockOptionLevel               4
TotalWorkingYears             40
TrainingTimesLastYear          7
WorkLifeBalance                4
YearsAtCompany                37
YearsInCurrentRole            19
YearsSinceLastPromotion       16
YearsWithCurrManager          18
dtype: int64
```

```python
# drop columns that are irrelevant
df = df.drop(['Over18','EmployeeNumber', 'EmployeeCount', 'StandardHours'], axis = 1)
print(df.shape)
df.head()
```

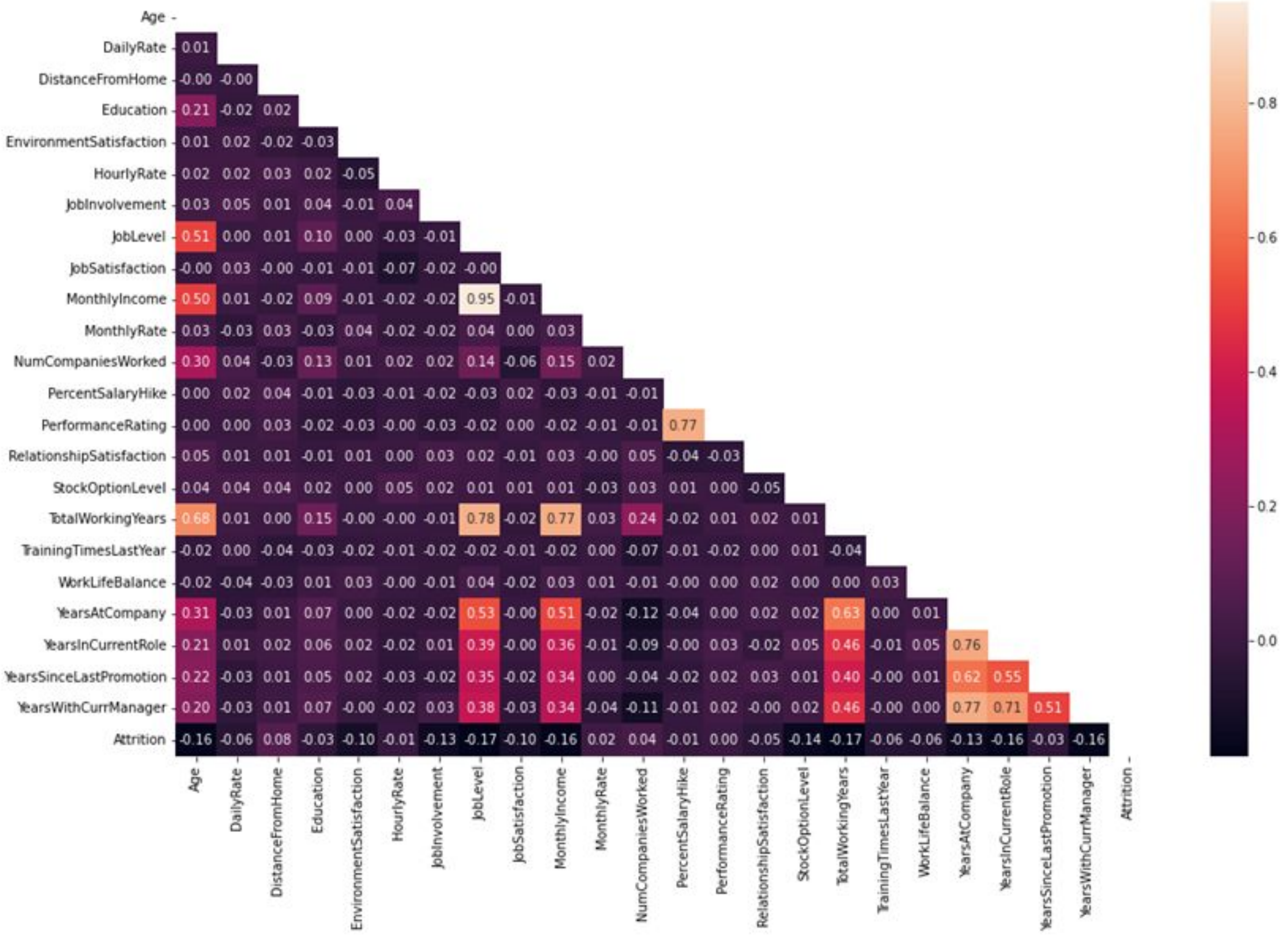
✓ 0.7s

```
(1470, 31)
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	Gender	HourlyRate	JobInvolvement
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	2	Female	94	3
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	3	Male	61	2
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	4	Male	92	2
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	4	Female	56	3
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	Male	40	3

Feature Engineering

- Correlation Matrix for variables
- One Hot Encoding



```python
# apply one hot encoding to non numerical features
df_clean = pd.get_dummies(df_clean, columns = ['BusinessTravel', 'Gender','MaritalStatus'], drop_first = True)
df_clean = pd.get_dummies(df_clean)
df_clean.head()
```

✓ 0.1s

	Age	DailyRate	DistanceFromHome	Education	EnvironmentSatisfaction	HourlyRate	JobInvolvement	JobLevel	JobSatisfaction	MonthlyIncome	MonthlyRate	NumCompaniesWorked	OverTime	PercentSalaryHike
0	41	1102	1	2	2	94	3	2	4	5993	19479	8	1	11
1	49	279	8	1	3	61	2	2	2	5130	24907	1	0	23
2	37	1373	2	2	4	92	2	1	3	2090	2396	6	1	15
3	33	1392	3	4	4	56	3	1	3	2909	23159	1	1	11
4	27	591	2	1	1	40	3	1	2	3468	16632	9	0	12

Data Scaling

```
# filter out features that needs to be standarized
col_tobe_standard = ['Age', 'DailyRate', 'DistanceFromHome', 'Education', 'EnvironmentSatisfaction',
                     'HourlyRate', 'JobInvolvement', 'JobLevel', 'JobSatisfaction', 'MonthlyIncome',
                     'MonthlyRate', 'NumCompaniesWorked', 'PercentSalaryHike', 'PerformanceRating',
                     'RelationshipSatisfaction', 'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',
                     'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
                     'YearsWithCurrManager']
```

✓ 0.5s

```
# standarization on numercial features so that all the numerical features are having the same type of normal distribution
scaler = StandardScaler()
for col in col_tobe_standard:
    df_clean[col] = df_clean[col].astype(float)
    df_clean[[col]] = scaler.fit_transform(df_clean[[col]])
df_clean.head()
```

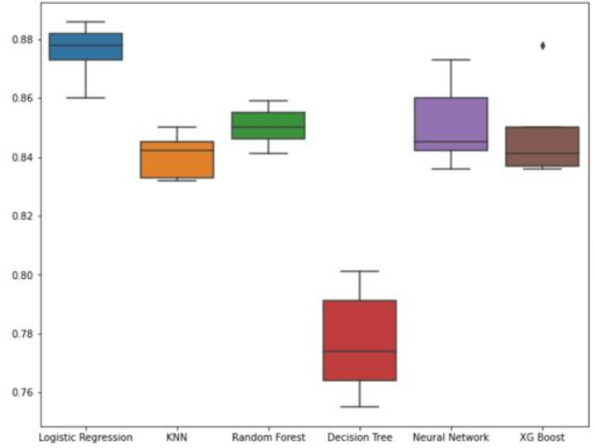
✓ 0.3s

	Age	DailyRate	DistanceFromHome	Education	EnvironmentSatisfaction	HourlyRate	JobInvolvement	JobLevel	JobSatisfacti
0	0.446350	0.742527	-1.010909	-0.891688	-0.660531	1.383138	0.379672	-0.057788	1.1532
1	1.322365	-1.297775	-0.147150	-1.868426	0.254625	-0.240677	-1.026167	-0.057788	-0.6608
2	0.008343	1.414363	-0.887515	-0.891688	1.169781	1.284725	-1.026167	-0.961486	0.2462
3	-0.429664	1.461466	-0.764121	1.061787	1.169781	-0.486709	0.379672	-0.961486	0.2462
4	-1.086676	-0.524295	-0.887515	-1.868426	-1.575686	-1.274014	0.379672	-0.961486	-0.6608

89 %
To good to be true?

Do we need to deal with data imbalance / overprediction

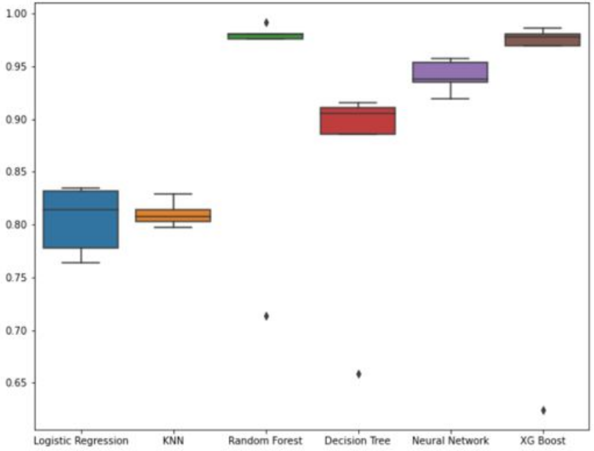
Data Imbalance Not Addressed



Predicted
Accuracy = 0.891
Precision = 0.771
Recell = 0.458
F1 = 0.574

"To good to be true?"

Data Imbalance Not Addressed (SMOTE)



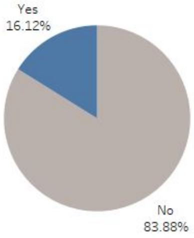
Predicted
Accuracy = 0.731
Precision = 0.344
Recell = 0.746
F1 = 0.471

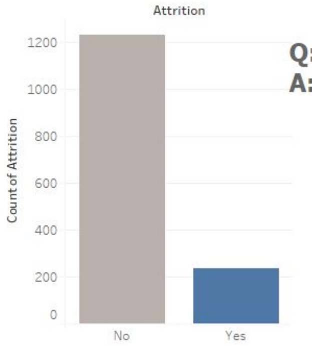
Challenges
Data Imbalance

Solution

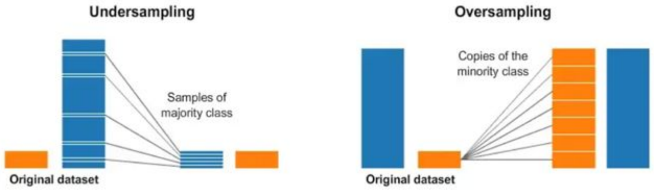
imblearn
- Oversampling
- Undersampling
- SMOTE

(Synthetic Minority Oversampling Technique)

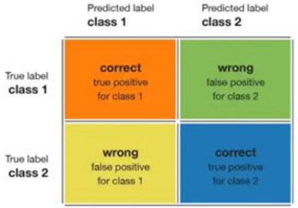


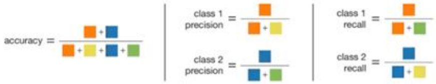


Q: How do we Deal with a Imbalanced DataSet
A: Use imblearn Library



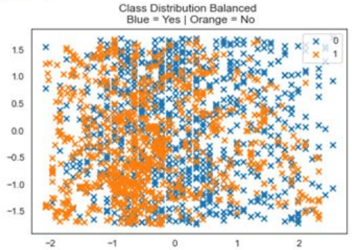
Sampling for imbalanced data set





Using: Oversampling & Undersampling

Using SMOTE



"ACCURACY" is not the right matrix when working for the Imbalanced data set. When you use the confusion matrix, we can evaluate how well the model is really performing.

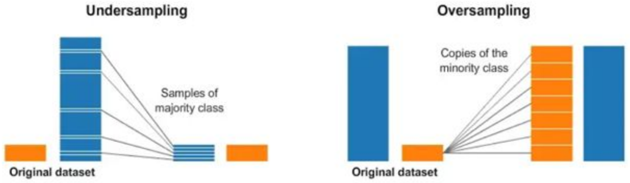
- The accuracy of the model is basically the total number of correct predictions divided by total number of predictions.
- The precision of a class define how trustable is the result when the model answer that a point belongs to that class.
- The recall of a class expresses how well the model is able to detect that class.
- The F1 score of a class is given by the harmonic mean of precision and recall (2×precision×recall / (precision + recall)), it combines precision and recall of a class in one metric.

Oversampling & Undersampling

Using Under & Over Sampling Methods

"ACCURACY" is not the right matrix when working for the Imbalanced data set. When you use the confusion matrix, we can evaluate how well the model is really performing.

- The accuracy of the model is basically the total number of correct predictions divided by total number of predictions.
- The precision of a class define how trustable is the result when the model answer that a point belongs to that class.
- The recall of a class expresses how well the model is able to detect that class.
- The F1 score of a class is given by the harmonic mean of precision and recall (2×precision×recall / (precision + recall)), it combines precision and recall of a class in one metric.



Sampling for imbalanced data set

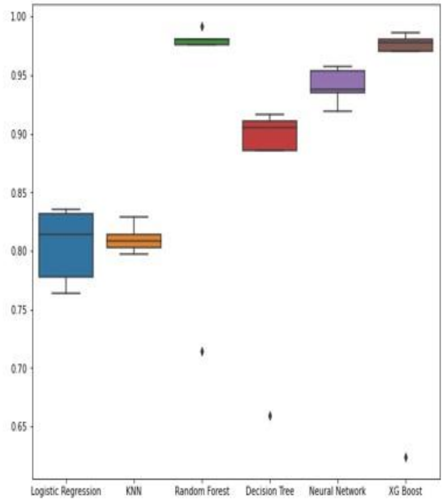
	method	mean score	std score	recall score
0	Logistic Regression	0.315079	0.315079	0.322034
1	KNN Classifier	0.140794	0.032205	0.169492
2	Random Forest Classifier	0.106984	0.060737	0.220339
3	Decision Tree Classifier	0.400159	0.100461	0.474576
4	MLP Classifier	0.196984	0.095284	0.474576
5	XG Boost	0.297937	0.043541	0.322034

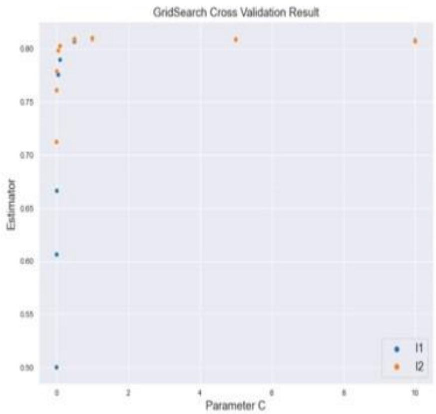
	method	mean score	std score	recall score
0	Logistic Regression UnderSampling	0.315079	0.702063	0.779661
1	KNN Classifier UnderSampling	0.568730	0.127682	0.762712
2	Random Forest Classifier UnderSampling	0.697460	0.079316	0.762712
3	Decision Tree Classifier UnderSampling	0.674127	0.094545	0.559322
4	MLP Classifier UnderSampling	0.592381	0.205470	0.966102
5	XG Boost UnderSampling	0.714127	0.087748	0.728814

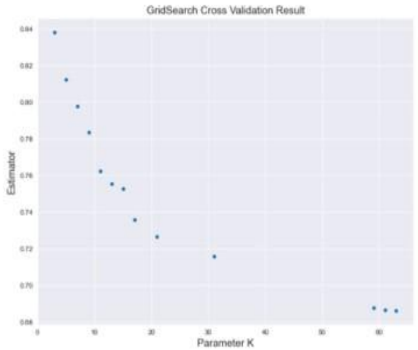
	method	mean score	std score	recall score
0	Logistic Regression OverSampling	0.702063	0.063689	0.796610
1	KNN Classifier OverSampling	0.568730	0.127682	0.762712
2	Random Forest Classifier OverSampling	0.202222	0.045051	0.338983
3	Decision Tree Classifier OverSampling	0.263651	0.073520	0.491525
4	MLP Classifier OverSampling	0.752540	0.180019	0.949153
5	XG Boost OverSampling	0.330952	0.059370	0.474576

SMOTE

Model Training and Performance Evaluation

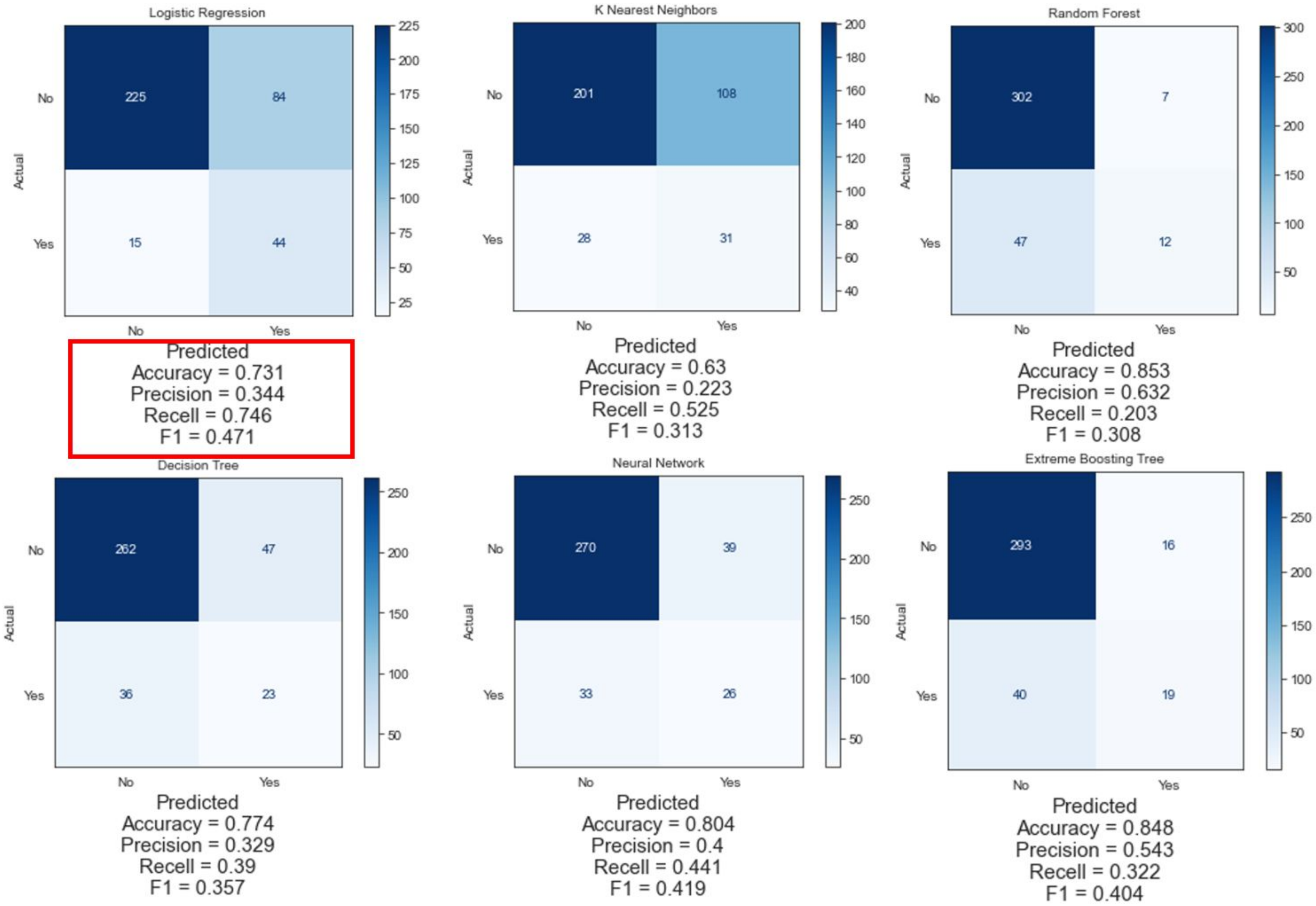






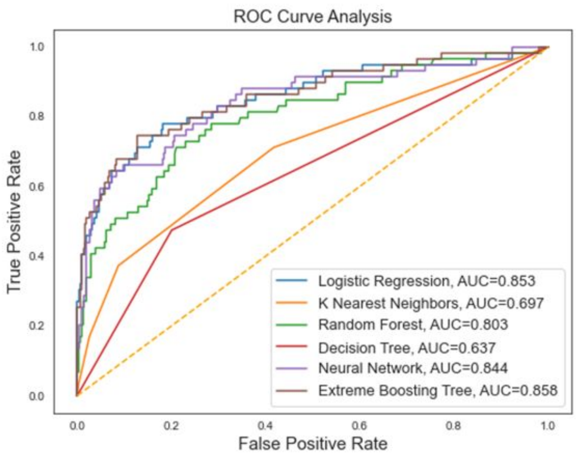
Confusion Matrix

SMOTE

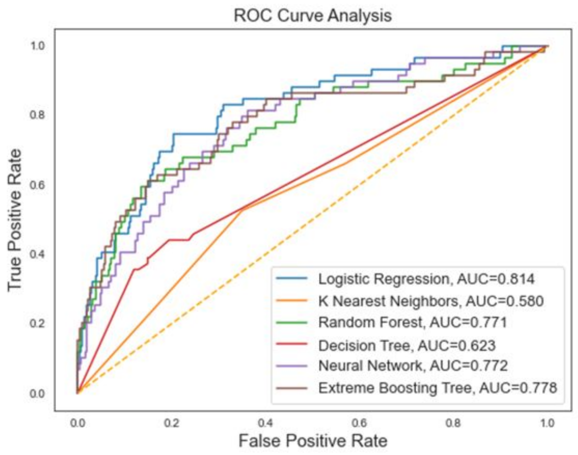


ROC Curve Analysis Comparison

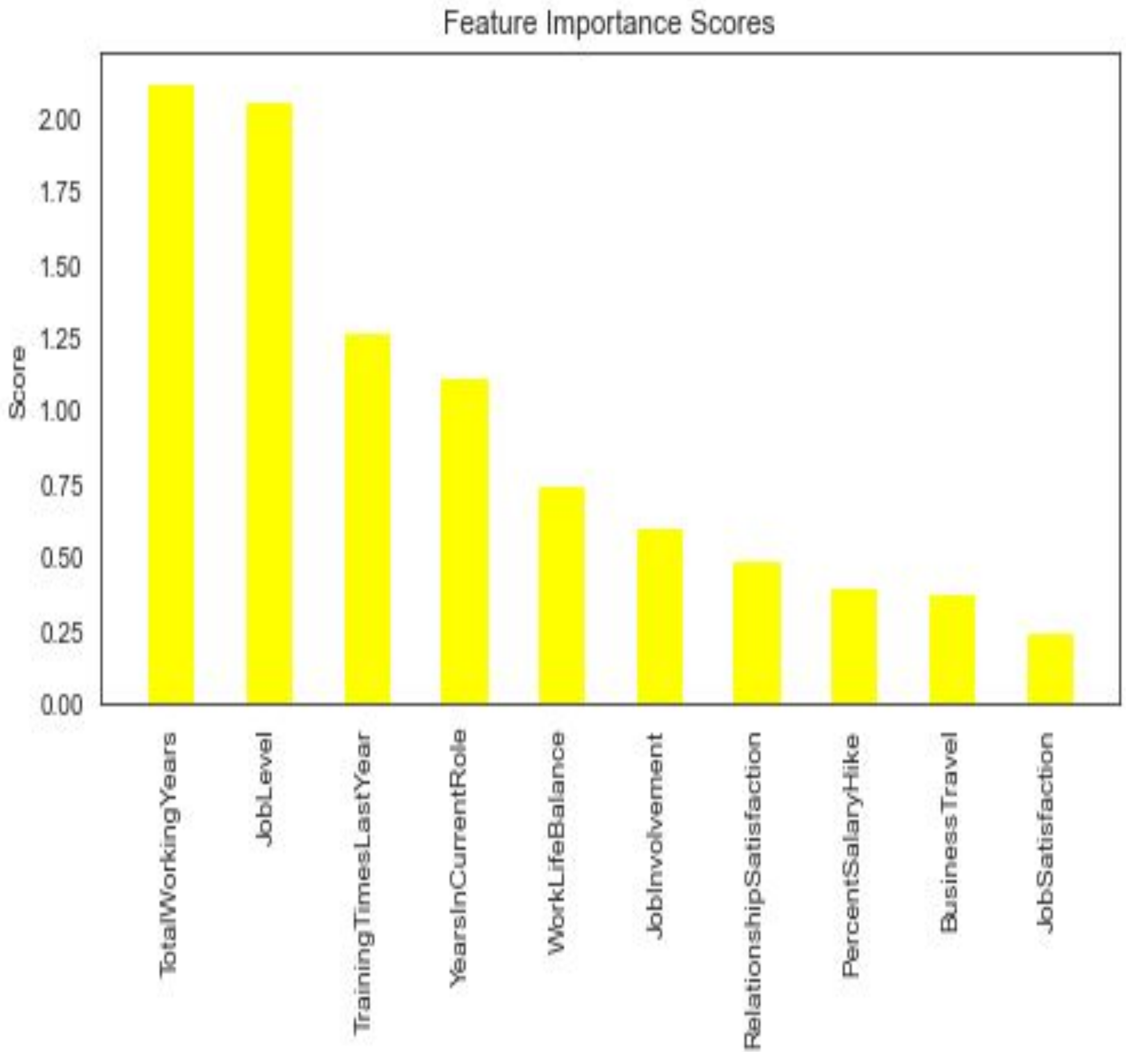
Data Imbalance Not Addressed



Data Imbalance Not Addressed (SMOTE)



Feature Importance



Conclusions

"Thankyou https://medium.com/"

<u>Conclusions from the Project</u>.

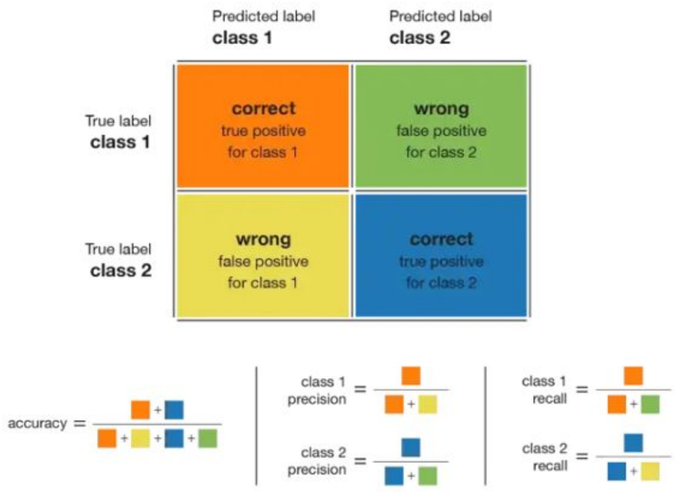
We built a model that was achieving a score between 77 – 88 % depending on the method used.

On reflection this is to be expected as the data is bias towards NO. (84% to 16%).

So, the model learns that if it picks NO, it will be right 84% of the time. The model was doing the right thing but the way of training the model is wrong and we are focusing on the wrong thing.

"ACCURACY" is not the right matrix when working for the Imbalanced data set.

When you use the confusion matrix, we can evaluate how well the model is really performing.



- The accuracy of the model is basically the total number of correct predictions divided by total number of predictions.
- The precision of a class define how trustable is the result when the model answer that a point belongs to that class.
- The recall of a class expresses how well the model can detect that class.
- The F1 score of a class is given by the harmonic mean of precision and recall (2×precision×recall / (precision + recall)), it combines precision and recall of a class in one metric.