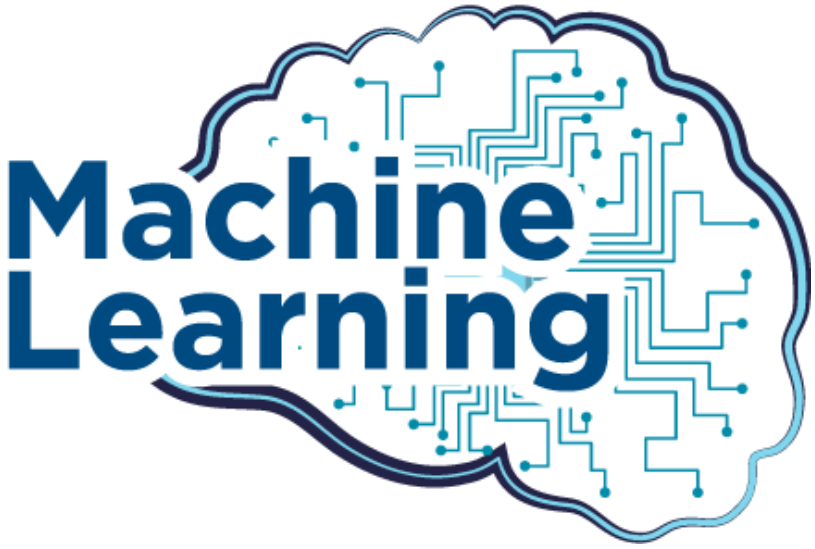


Employee Attrition Prediction Using Machine Learning Algorithms.

Machine Learning Project Report
November 11, 2022

TABLE OF CONTENT

S. No	TOPICS	Page No.
1	DATASET	03
2	EXPLORATORY DATA ANALYSIS	03
3	DATA PRE-PROCESSING	10
4	METHODOLOGY	11
4	MODEL EVALUATION	12
5	FEATURE SELECTION	16
6	MODEL IMPLEMENTATION	17
7	RESULTS AND DEPLOYEMENT	27
8	CONCLUSION	27

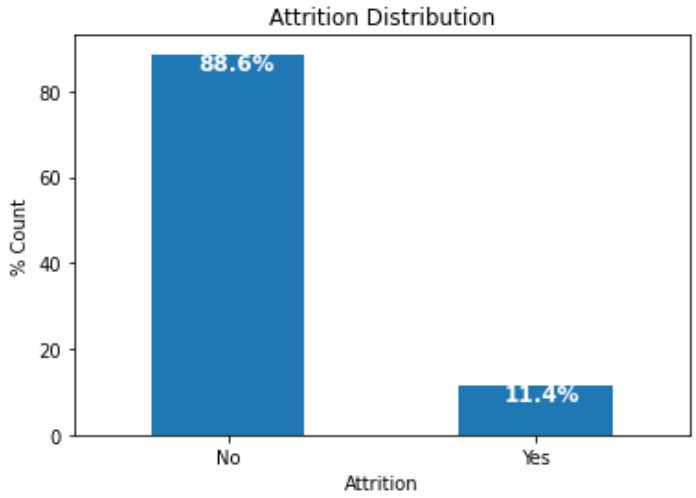
DATASET: -

Dataset we are using in this case study is Employee attrition analysis dataset and contains organizational information of employees. Dataset contains 721 employees' records and 21 features. We are using "Attrition" as our target variable. It is a Binary class classification problem, where value of class is 0 and 1 (0 means no attrition and 1 means attrition). Dataset is imbalanced with 80%-20% ratio and consist of 20 attributes other than target attribute. We are using Oversampling technique to balance out this data with 50% values of each class.

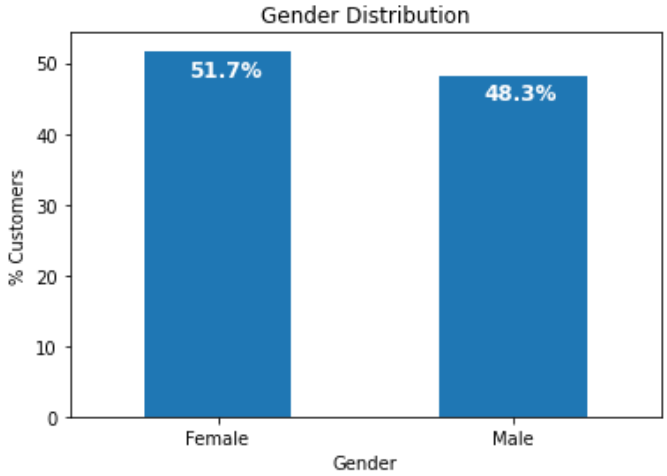
EXPLORATORY DATA ANALYSIS: -

EDA is a technique which is used to explore more about the data and fetch some useful information from it using graphical representations. It is to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods.

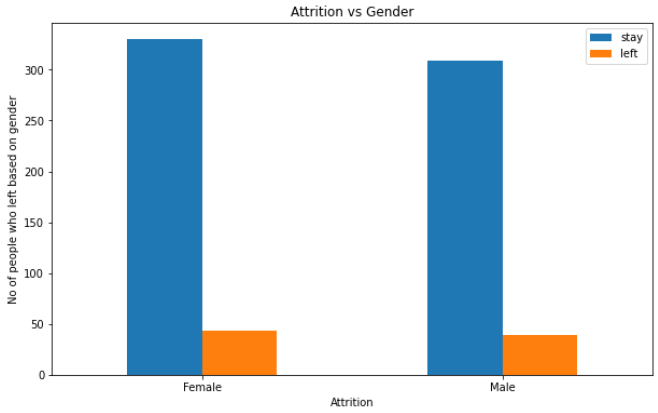
In our data we are also using EDA to get some knowledge about the data. In our dataset we have total 721 employee's data, out of which 639 are not leaving whereas 82 people leave the organization.



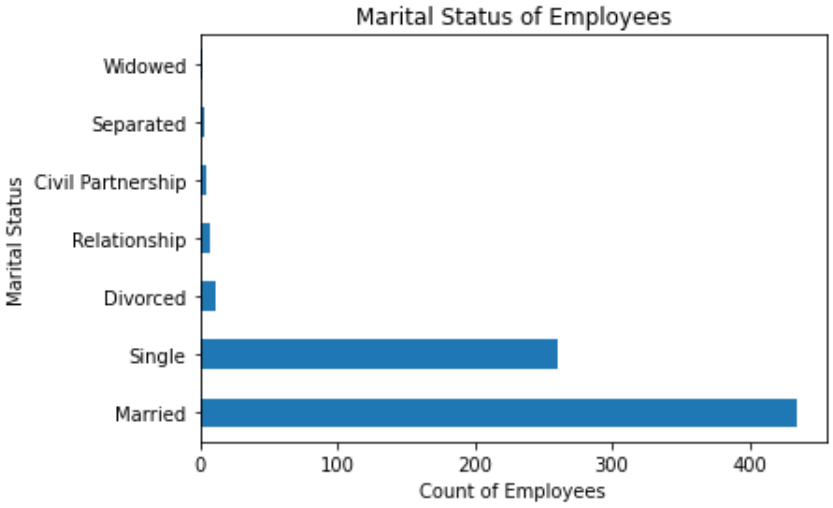
Out of 721 employees, 373 are females, whereas 348 are males. % Of female are little higher than male in the organization.



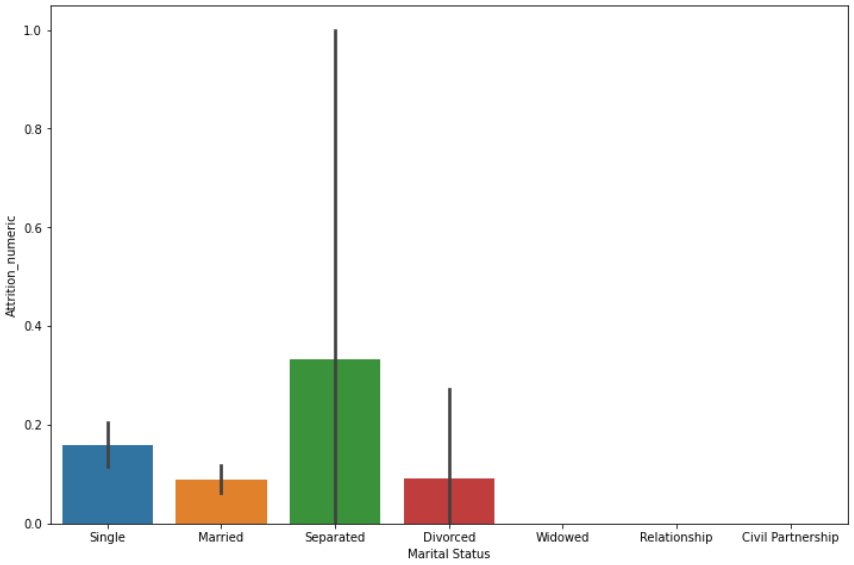
Rate of attrition for both male and female are almost same. This mean that both are leaving with same rate.



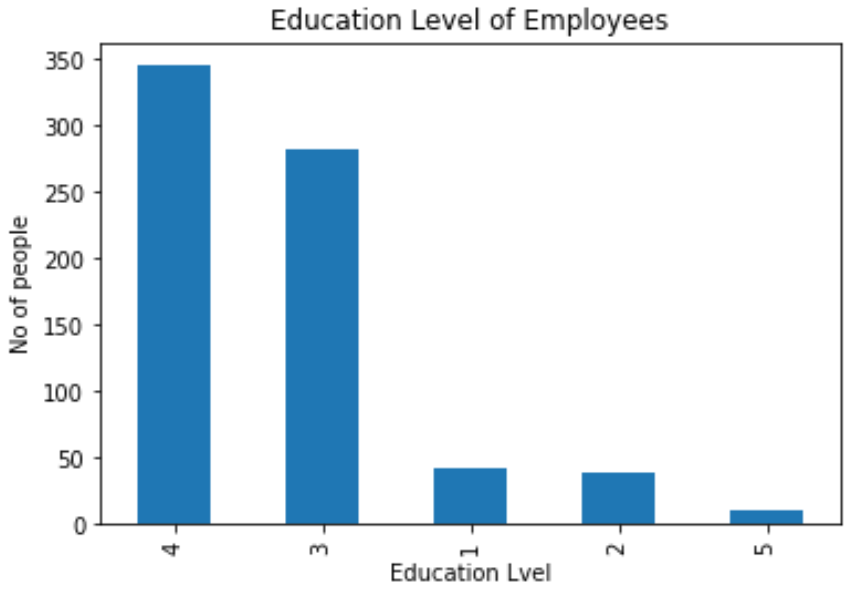
People have different marital status in the organization but most of the people are married, followed by singles. Out of 721 around 450 are married and 200 are singles.



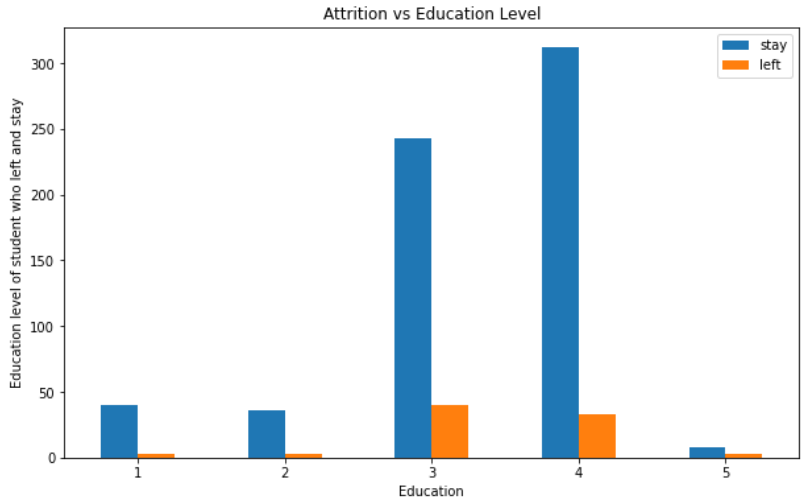
Attrition rate of separated employees are higher as compared to others. Out of 3, one person who is facing separation is leaving.



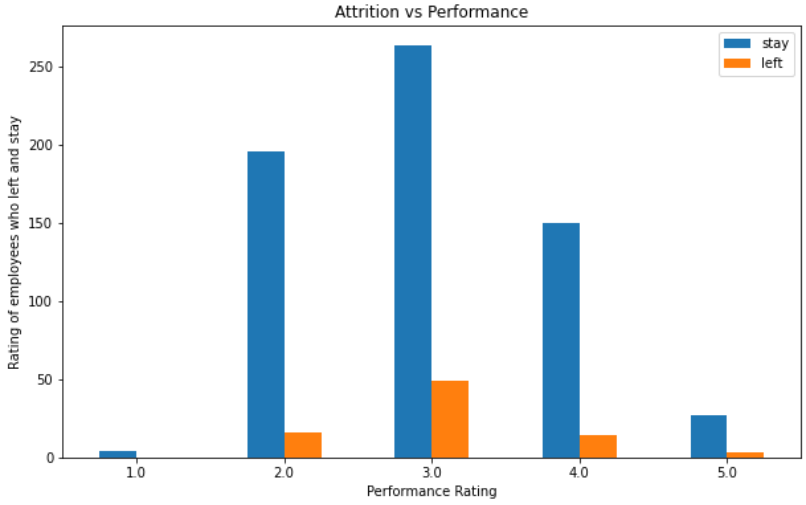
We have around five level of education in our data set out of these five most of the employees has level 4 education followed by a Level 3, very few employees have level 5 education which is highest level.



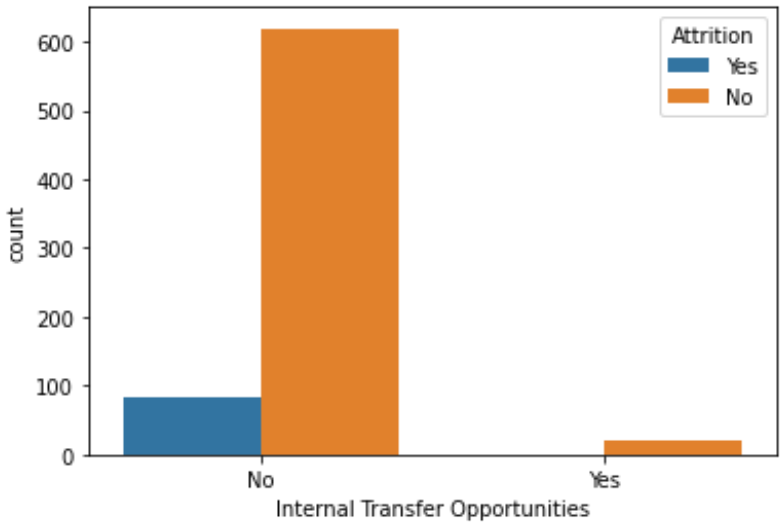
Employees with level 4 and Level 3 education are having higher rate of Attrition as compared to the other level of education.



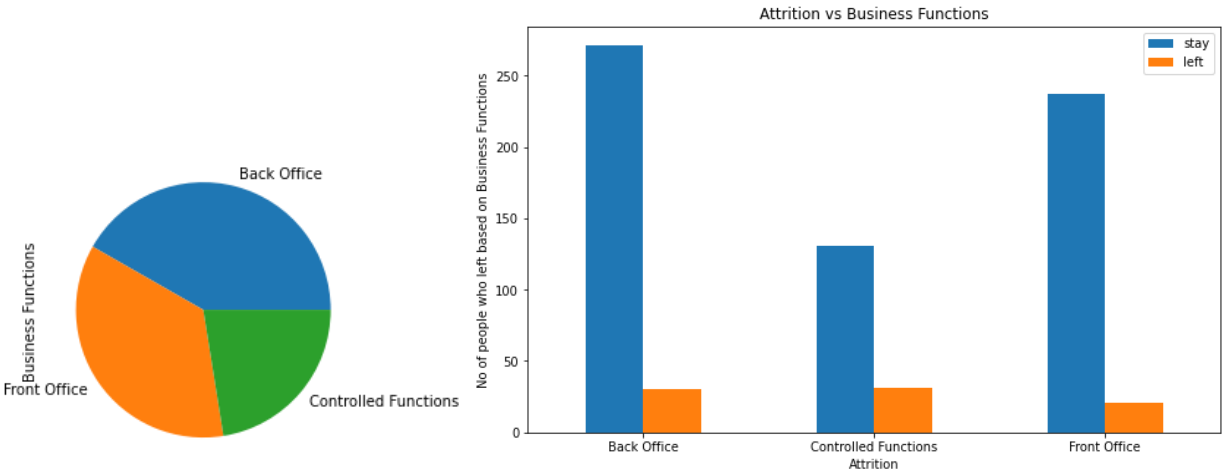
We have five standards of performance ratings in our dataset. Most of the employees has performance rating of three and four, but the rate of Attrition is also high among employees who has performance rating of 3 and 4.



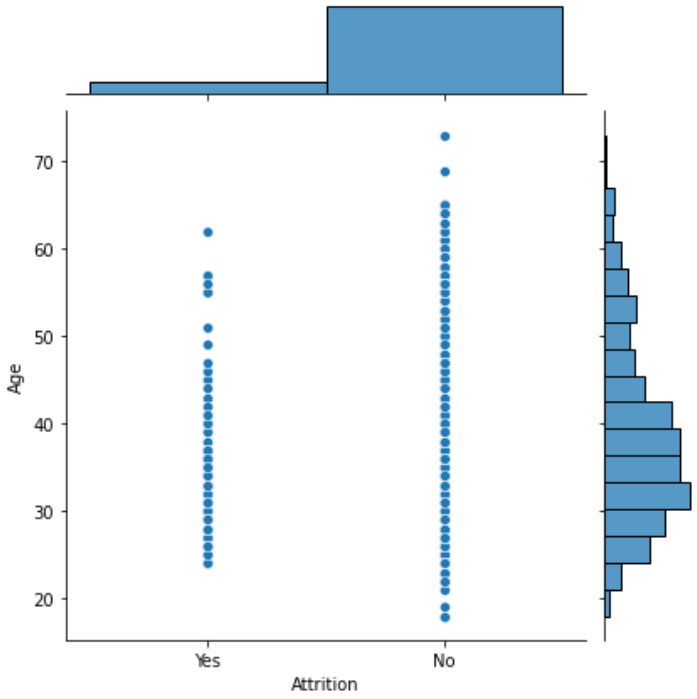
Most of the employees have no transfer opportunities, but those who have transfer opportunities are not leaving the organization.



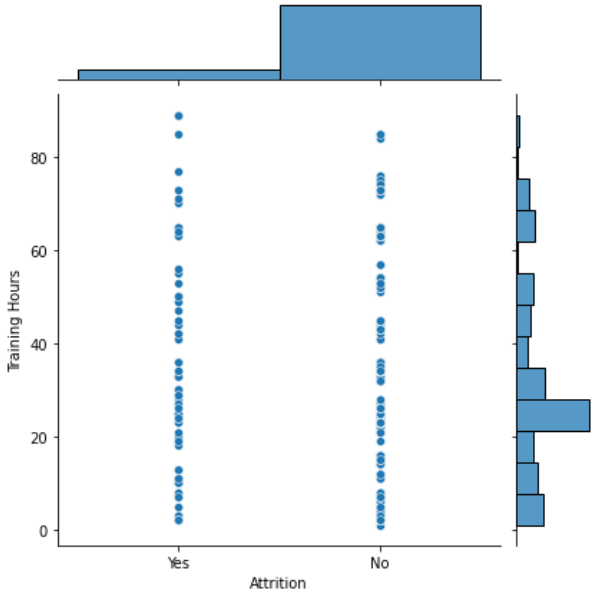
There are total three business functions in our data set most of the employees of control functions are leaving the organization, whereas back office and front office has very low rate of attrition



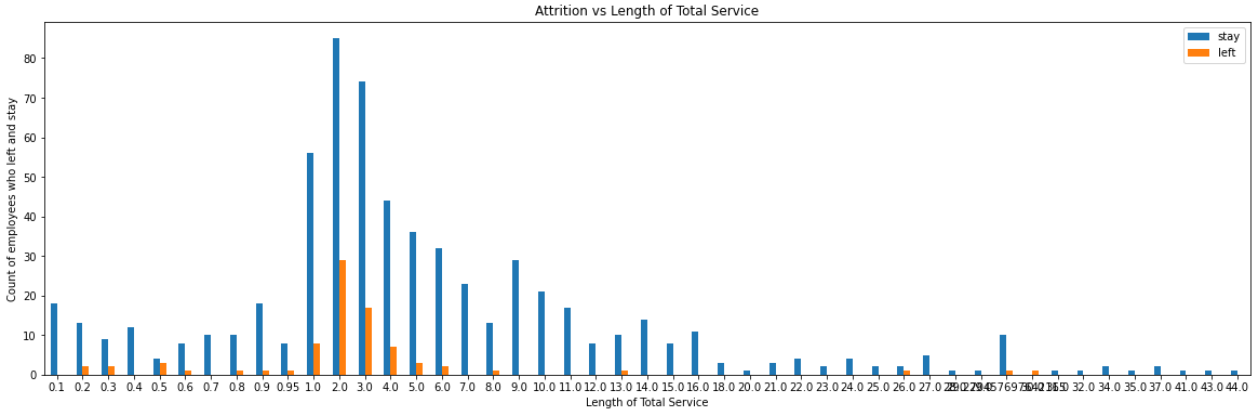
People from age 20 to age 70 are working in the organization but most of the people are of age between 30 years to 40 years.



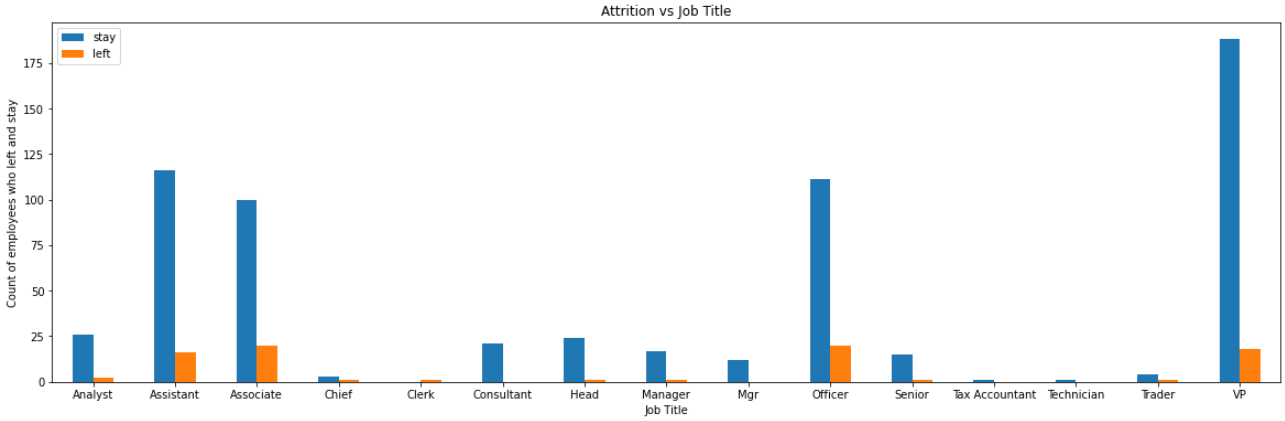
People who have training of around 20 hours is expected more to leave the organization as compared to the other people having training higher that this range.



People who are working in the organization and have experience of around one year to three-years are more expected to leave the organization as compared to two as compared to the people having experience less than one year or more than three-years.



People who are assistant, associate, and officer are having higher rate of attrition as compared to the other job positions.



Train – Test Split: -

When machine learning algorithms are used to make predictions on data that was not used to train the model, their performance is estimated using the train-test split technique. Here we are splitting our dataset in train dataset and test dataset with the ratio of 70-30% where 70% is our train dataset and 30% is our test dataset.

Data Pre- Processing: -

Before using the data in the machine learning algorithms, preparing the data is a crucial step. Data preprocessing is a data mining technique used to turn the raw data into a format that is both practical and effective.

For our dataset, we also must perform some data pre-processing. First, we load our dataset using pandas and check shape of our dataset which is 721 and 21 columns. Dataset is normalized and contain both integer, string, and float values. Most important thing we are doing in this pre-processing part is converting our binary string columns values into ‘0’ and ‘1’. So that after doing encoding dimension of the data don’t increase too much. We are reducing unique value from the string columns by replacing it from base value.

Data has some null values in some columns which we removed by adding median and mean of values. After that we dropped some unnecessary column such index id, resignation reason and

Ethic Origin. We dropped resignation reason column because it contains a lot of null values and not relevant to the target variable.

One-Hot Encoding: -

One hot encoding is a process of converting categorical data variables so they can be provided to machine learning algorithms to improve predictions. One hot encoding is a crucial part of feature engineering for machine learning. our machine learning algorithms only understand numbers, so we must provide them numbers by converting categorical variables into continuous variables. It Generate different columns and assign them binary values.

One hot encoding is essential before running machine learning algorithm on data set. Some algorithms can understand categorical data directly such as decision tree but most of the supervised learning algorithm cannot operate on categorical data, they require all input variables to be numeric and generate output in numeric value. This technique of transforming columns into binary variables data set is quite famous in Supervised learning algorithm. These binary variables are also known as dummy variables in statistics So, after performing one-hot encoding on our dataset, we now have around 78 columns. Before it was 21. So now, we can see number of features transformed and increased after encoding.

METHODOLOGY: -

We are using different machine learning algorithms in this project for employee attrition prediction and comparing results of each algorithm to check out which model is fitting and performing well on training data set. We are also using some model enhancement techniques to improve performance of the model. These techniques are K-fold cross validation and ensemble-based model (voting classifier). Our sole purpose is to improve performance of model with fine accuracy.

First, we are applying our models on all attributes and checking performance of our models. After that, we are using feature selection technique and selecting topmost impactful features using information gain approach. We are also using oversampling technique to balance our dataset and applying machine learning algorithm on sampled data. In the end, we are using sampling and feature selection technique to further enhance our machine learning model results.

Model Evaluation: -

We are evaluating performance of our model's using accuracy, confusion matrix, F1-Score, ROC_AUC score, sensitivity, specificity, miss rate, precision, and recall values. Before we move forward to model implementation, it is necessary to know following terms.

1) Precision: -

The proportion of True Positives to All Positives is known as precision. For our problem statement, that would be the measure of cases that we correctly identify having a disease out of all the cases having it.

2) Recall: -

The recall is the number of our model rightly identifying True Positives values. Thus, for all the cases who have disease, recall tells us how many we correctly identified as having a disease.

3) Sensitivity: -

The sensitivity of a machine learning model indicates how effectively it can recognize successful examples. Other names for it include the true positive rate (TPR) or recall. Sensitivity is used to evaluate model performance since it allows us to ascertain how many examples the model was able to correctly identify.

4) Specificity: -

Specificity is the number of true negatives that are rightly predicted by the model.

5) Miss Rate: -

The miss rate is commonly known as false positive rate – is the likelihood that the test will fail to detect a true positive. FN/FN+TP, where FN is the number of false negatives and

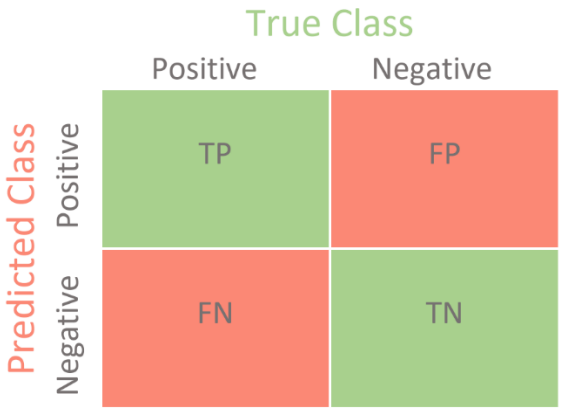
TP is the number of true positives (FN+TP being the total number of positives), is the formula used to compute it.

6) False Positive Rate: -

The ratio of false positives to true negatives, or false positives to true negatives plus true negatives (FP+TN), is used to compute the false positive rate. It's the likelihood that an error will occur, meaning that a positive result will be reported when a negative value exists.

7) Confusion Matrix: -

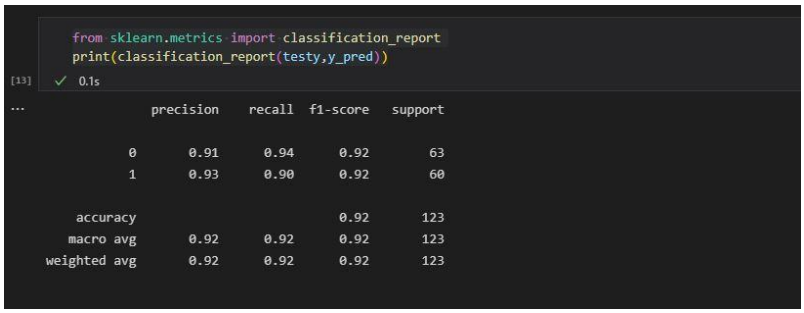
A classification problem's prediction outcomes are compiled in a confusion matrix. Count values are used to describe the number of accurate and inaccurate predictions for each class. This is the confusion matrix.



8) Classification Report: -

The accuracy of predictions made by a classification algorithm is evaluated using a classification report. How many of the forecasts came true, and how many were wrong?

More specifically, the metrics of a categorization report are predicted using True Positives, False Positives, True Negatives, and False Negatives.



```
from sklearn.metrics import classification_report
print(classification_report(testy,y_pred))
```

[13] ✓ 0.1s

```
              precision    recall  f1-score   support

           0       0.91      0.94      0.92        63
           1       0.93      0.90      0.92        60

    accuracy                           0.92       123
   macro avg       0.92      0.92      0.92       123
weighted avg       0.92      0.92      0.92       123
```

9) F-1 Score: -

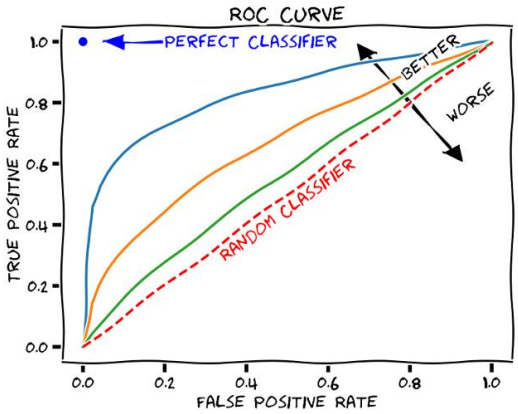
The accuracy of a model on a dataset is gauged by the F-score, also known as the F1-score. It is employed to assess binary categorization schemes that divide examples into "positive" and "negative" categories.

The precision and recall of the model are combined through the F-score.

10) ROC – AUC Curve: -

A measurement tool for binary classification issues is the Receiver Operator Characteristic (ROC) curve. In essence, it separates the "signal" from the "noise" by plotting the TPR against the FPR at different threshold values. The capacity of a classifier to differentiate between classes is measured by the Area Under the Curve (AUC), which is used as a summary of the ROC curve.

The model performs better at differentiating between the positive and negative classes the higher the AUC. The classifier can accurately discriminate between all Positive and Negative class points when AUC = 1. The classifier would be predicting all Negatives as Positives and all Positives as Negatives, however, if the AUC had been 0.



K-FOLD Cross Validation: -

A resampling technique called cross-validation is used to assess machine learning models on a small data sample.

The process contains a single parameter, k, that designates how many groups should be created from a given data sample. As a result, the process is frequently referred to as k-fold cross-validation. When k is set to a specific value, that value may be used in place of k when referring to the model, such as when k=5 is substituted for 5-fold cross-validation.

In applied machine learning, cross-validation is mostly used to gauge how well a machine learning model performs on untrained data. That is, to use a small sample to gauge how the model will generally perform when used to make predictions on data that was not included during the model's training.

In our project, we are using 10-fold cross validation, in which we are calculating cross validation score on our training data. It is returning us accuracies of 10 folds and mean accuracy. K-fold CV help in decreasing model complexity and in some cases, it also improves model results.

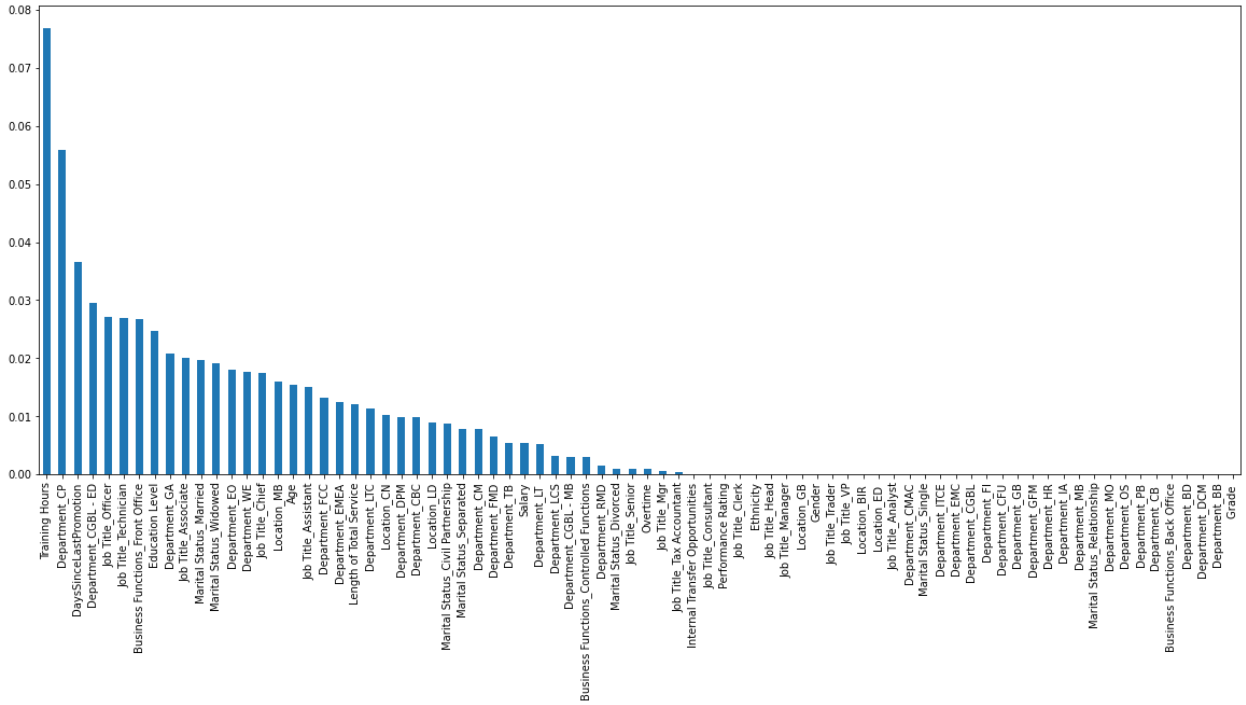
Feature Selection: -

It is almost never the case that all the variables in a dataset can be used to develop a machine learning model in practice. Repetitive factors decrease a classifier's capacity to generalize and may also lower the classifier's overall accuracy. Additionally, a model becomes more complex overall as more variables are added to it.

First, we will apply all machine learning models on all our features then after that we will select some features using Information Gain approach and apply models on selected features.

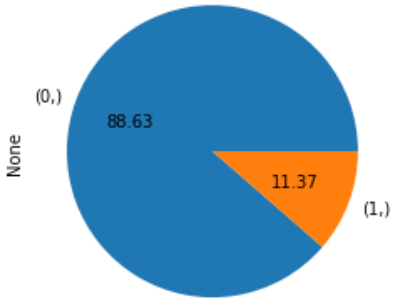
Information gain calculates the reduction in entropy or surprise from transforming a dataset in some way. Information gain is used for feature selection, by evaluating the gain of each variable

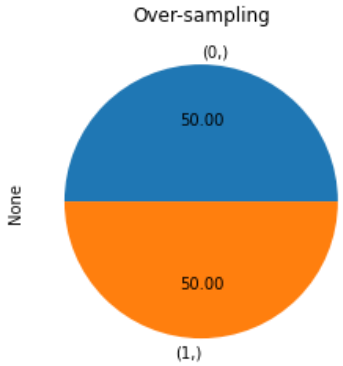
in the context of the target variable. In this slightly different usage, the calculation is referred to as mutual information between the two random variables.



Over Sampling Technique: -

We are using oversampling technique to balance our unbalance dataset. Initially our data contain 80% ‘NO’ class and 20% ‘YES’ class in the data. After doing oversampling both classes will be 50% each. So, that our model will get train on equal number of classes data.





Model Implementation: -

At first, we are implementing different machine learning classifier on our dataset first without any feature selection and then after doing sampling on balanced dataset. After that we are implementing models on selected features unbalanced dataset and after that on selected features balanced dataset. section. We have 79 attributes after encoding, and our data is 70-30% divided into train and test sets.

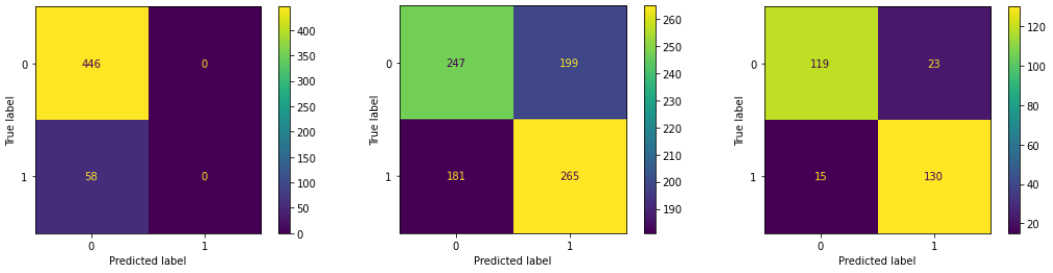
1) Logistic Regression: -

The first model we implement is a classification model logistic regression model, which is frequently used to foretell whether a given instance belongs to a particular class or not. It employs the sigmoid function, which returns a number between 0 and 1, to predict the probability of a specific class. In our situation, logistic regression provides accuracy on our test data set of more than 80%. With an AUC value of only 0.5, it provides accuracy of 83%, which is not up to the mark. Additionally, its performance improved following sampling but still not very good.

Features	Without feature selection	After Over Sampling	After Feature Selection
Model	Logistic Regression	Logistic Regression	Logistic Regression
Accuracy	88.94%	62.67%	88.47%
AUC Score	0.5	0.57	0.51
Specificity	1.0	0.642	0.98
Sensitivity	0.0	0.5	0.04
Recall	0.0	0.5	0.04
Precision	0.0	0.693	0.33
Miss Rate (FNR)	1.0	0.5	0.95
Miss Rate (FPR)	1.0	0.35	0.01

K-Folds Cross – Validation Confusion Metrics for Logistic Regression: -

Following are the results of K-Fold Cross validation in term of confusion metrics, when applied on whole data, sampled data (balanced) and after feature respectively. First metrics from left is with all features, middle one is result with sampled and left one is confusion metrics result with features selection.



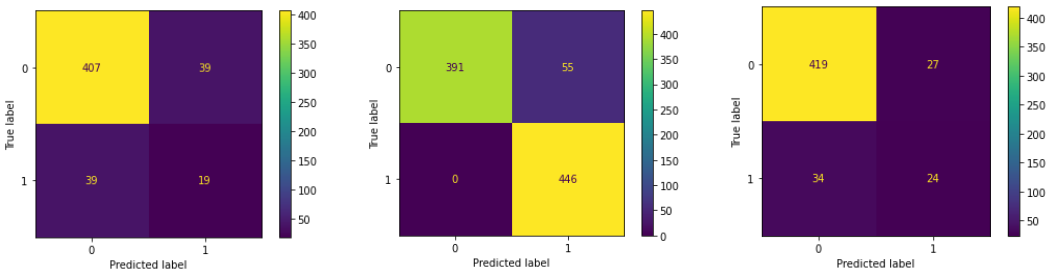
2) Decision Tree: -

The second model we're using on our train dataset is a decision tree that uses decision trees to classify results. It is a well-known machine learning algorithm that is used for both classification and regression issues. Although we used a decision tree on several criteria in our research, the outcome of ROC score prediction was not satisfactory. Nevertheless, the accuracy of the decision tree was 84.0% at all features, and it performed better following feature selection. These are some of the outcomes from the decision tree implementation.

Features	Without feature selection	After Over Sampling	After Feature Selection
Model	Decision Tree	Decision Tree	Decision Tree
Accuracy	84.0%	80.18%	86.63%
AUC Score	0.54	0.56	0.63
Specificity	0.92	0.870	0.932
Sensitivity	0.16	0.25	0.33
Recall	0.166	0.25	0.33
Precision	0.210	0.193	0.38
Miss Rate (FNR)	0.833	0.75	0.66
Miss Rate (FPR)	0.077	0.129	0.067

K-Folds Cross – Validation Confusion Metrics for Decision Tree: -

Following are the results of K-Fold Cross validation in term of confusion metrics, when applied on whole data, sampled data (balanced) and after feature respectively. First metrics from left is with all features, middle one is result with sampled and left one is confusion metrics result with features selection.



3) K- Nearest Neighbor: -

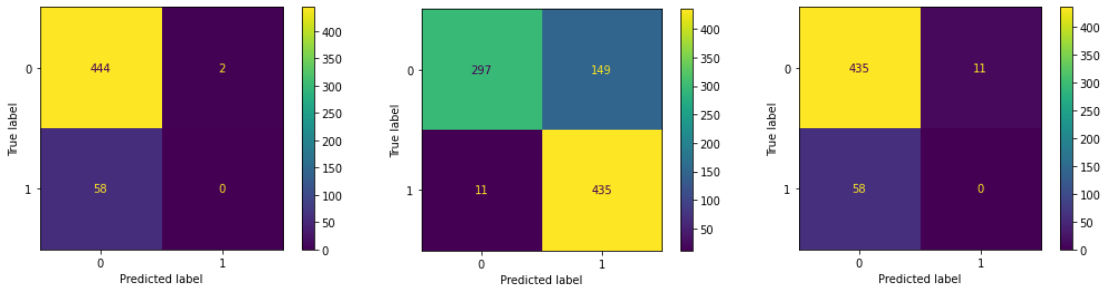
The supervised machine learning approach known as the k-nearest neighbors (KNN) model is straightforward and simple to apply. It can be used to tackle classification and regression issues. KNN searches for nearby nodes and returns results based on them. It determines its closest neighbors by calculating the distance to various neighbors. According to similarity in a particular group of nearby data points, KNN classifies data points.

We are also using KNN-model in this project; we implement model with different number of neighbors and its performance increase after feature selection.

Features	Without feature selection	After Over Sampling	After Feature Selection
Model	KNN	KNN	KNN
Accuracy	88%	61.75%	86.6%
AUC Score	0.49	0.47	0.505
Specificity	0.98	0.658	0.96
Sensitivity	0.0	0.291	0.04
Recall	0.0	0.291	0.04
Precision	0.0	0.095	0.14
Miss Rate (FNR)	1.0	0.708	0.958
Miss Rate (FPR)	0.010	0.341	0.031

K-Folds Cross – Validation Confusion Metrics for KNN Classifier: -

Following are the results of K-Fold Cross validation in term of confusion metrics, when applied on whole data, sampled data (balanced) and after feature respectively. First metrics from left is with all features, middle one is result with sampled and left one is confusion metrics result with features selection.



4) Random Forest Classifier: -

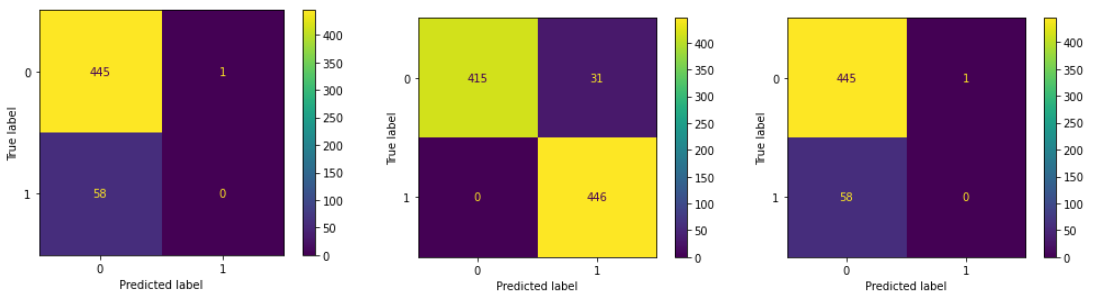
One of the well-known supervised learning methods is random forest. It's an ensemble model that integrates the results of various decision trees to make a learner that is more potent. RF is resistant to overfitting and has a strong noise resistance. The two basic ideas behind Random Forest are bagging and random selection.

In our project, performance of random forest is quite well. It is giving us ROC score better than previous models maybe because it is good with handling large number of features. It is good with high dimension data as we know that it works with making subsets by replacement. So, maybe this is the reason it is giving us good accuracy and roc score on train dataset.

Features	Without feature selection	After Over Sampling	After Feature Selection
Model	Random Forest	Random Forest	Random Forest
Accuracy	88.88%	76.95%	88.94%
AUC Score	0.49	0.706	0.536
Specificity	0.99	0.787	0.989
Sensitivity	0.0	0.625	0.083
Recall	0.0	0.625	0.083
Precision	0.0	0.267	0.5
Miss Rate (FNR)	1.0	0.375	0.916
Miss Rate (FPR)	0.005	0.212	0.010

K-Folds Cross – Validation Confusion Metrics for Random Forest: -

Following are the results of K-Fold Cross validation in term of confusion metrics, when applied on whole data, sampled data (balanced) and after feature respectively. First metrics from left is with all features, middle one is result with sampled and left one is confusion metrics result with features selection.



5) Gradient Boosting: -

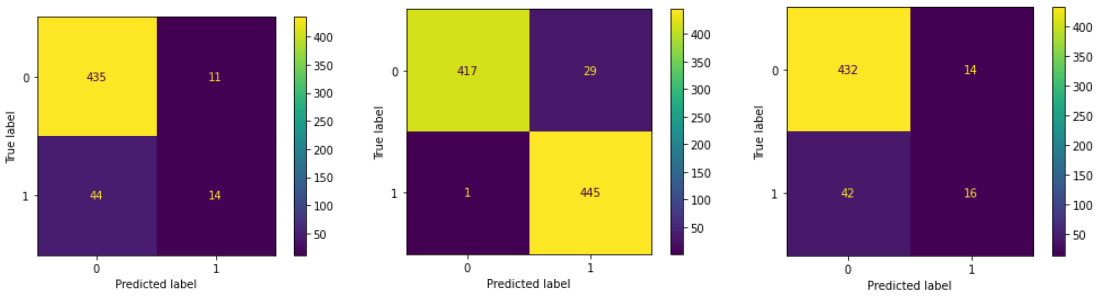
The supervised machine learning algorithm gradient boosting is used to solve classification and regression problems. By transforming weak learners into strong learners, we can enhance the model prediction of any given algorithm using this ensemble technique. The weak student is successively corrected by predecessors, becoming a strong learner as a result. When compared to other algorithms, gradient boosting classifiers are frequently quick and require less storage.

In this project, we also use Gradient boosting classifier and it's giving us optimum results as compared to other algorithms. Gradient boosting as good as random forest because it converts and enhance weak learner by reducing its error, but if the data is noisy than accuracy of gradient may affect sometime. We implement gradient boosting and it's fitting quite well.

Features	Without feature selection	After Over Sampling	After Feature Selection
Model	Gradient Boosting	Gradient Boosting	Gradient Boosting
Accuracy	88.47%	88.17%	91.72%
AUC Score	0.57	0.700	0.661
Specificity	0.974	0.94	0.989
Sensitivity	0.166	0.45	0.333
Recall	0.166	0.45	0.333
Precision	0.444	0.35	0.8
Miss Rate (FNR)	0.833	0.708	0.66
Miss Rate (FPR)	0.025	0.067	0.010

K-Folds Cross – Validation Confusion Metrics for Gradient Boosting: -

Following are the results of K-Fold Cross validation in term of confusion metrics, when applied on whole data, sampled data (balanced) and after feature respectively. First metrics from left is with all features, middle one is result with sampled and left one is confusion metrics result with features selection.



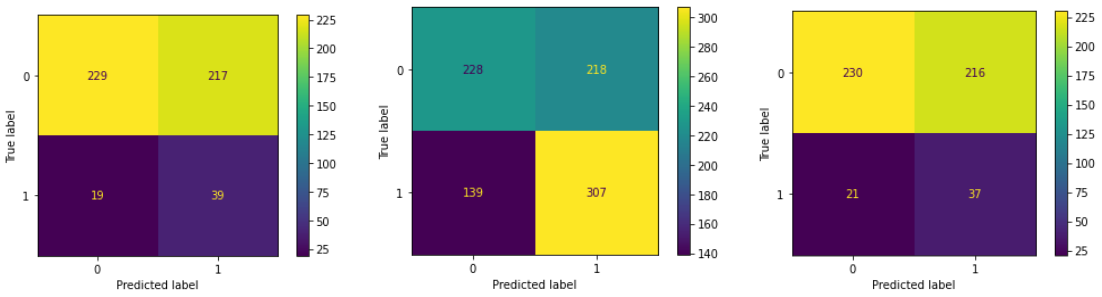
6) Naive Bayes Classifier: -

A group of supervised learning algorithms known as "naive" Bayes methods utilize Bayes' theorem with the "naive" assumption that every pair of features is conditionally independent given the value of the class variable. On our dataset, Naïve bayes is performing average as compared to the other models.

Features	Without feature selection	After Over Sampling	After Feature Selection
Model	Naïve Bayes	Naïve Bayes	Naïve Bayes
Accuracy	51.67%	50.69%	52.07%
AUC Score	0.545	0.583	0.548
Specificity	0.507	0.497	0.512
Sensitivity	0.583	0.583	0.583
Recall	0.583	0.583	0.583
Precision	0.12	0.126	0.129
Miss Rate (FNR)	0.416	0.416	0.359
Miss Rate (FPR)	0.492	0.502	0.115

K-Folds Cross – Validation Confusion Metrics for Naïve Bayes Classifier: -

Following are the results of K-Fold Cross validation in term of confusion metrics, when applied on whole data, sampled data (balanced) and after feature respectively. First metrics from left is with all features, middle one is result with sampled and left one is confusion metrics result with features selection.



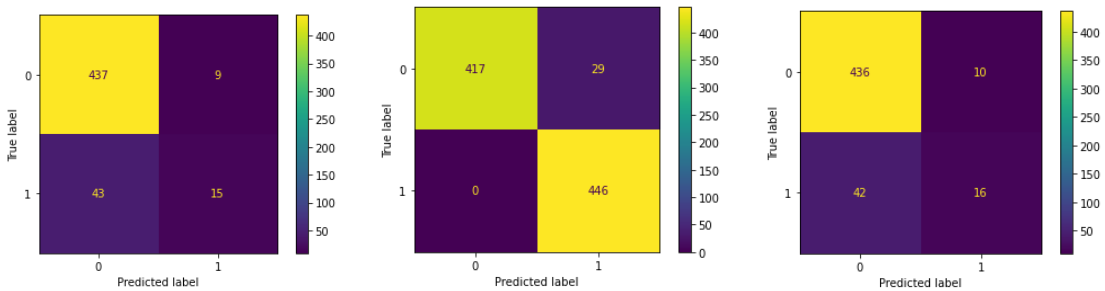
7) XGBOOST: -

XGBOOST is a decision-tree-based ensemble Machine Learning algorithm that uses a framework. In our case study, XGboost is performing very well and giving us accuracy of 91.70% with good roc, precision and recall values after feature selection.

Features	Without feature selection	After Over Sampling	After Feature Selection
Model	XGBoost	XGBoost	XGBoost
Accuracy	89.40%	89.86%	88.70%
AUC Score	0.630	0.705	0.679
Specificity	0.968	0.95	0.940
Sensitivity	0.291	0.45	0.416
Recall	0.291	0.45	0.416
Precision	0.538	0.455	0.476
Miss Rate (FNR)	0.708	0.541	0.583
Miss Rate (FPR)	0.031	0.04	0.05

K-Folds Cross – Validation Confusion Metrics for XGBoost: -

Following are the results of K-Fold Cross validation in term of confusion metrics, when applied on whole data, sampled data (balanced) and after feature respectively. First metrics from left is with all features, middle one is result with sampled and left one is confusion metrics result with features selection.



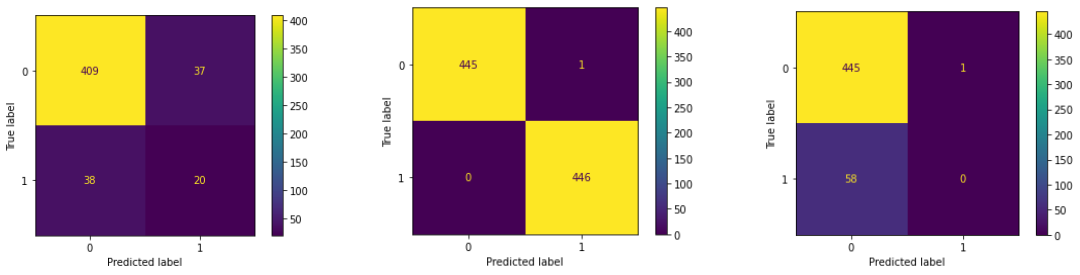
8) Support Vector Machine: -

Support Vector Machine" (SVM) is a supervised that can be used for both classification and regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is several features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well. In our project, SVM is performing good in term of accuracy after feature selection but average with 30 features.

Features	Without feature selection	After Over Sampling	After Feature Selection
Model	SVM	SVM	SVM
Accuracy	88.90%	88.18%	88.63%
AUC Score	0.5	0.5	0.5
Specificity	1.0	1.0	1.0
Sensitivity	0.0	0.0	0.0
Recall	0.0	0.0	0.0
Precision	0.0	0.0	0.0
Miss Rate (FNR)	1.0	1.0	1.0
Miss Rate (FPR)	0.0	0.0	0.0

K-Folds Cross – Validation Confusion Metrics for SVM Classifier: -

Following are the results of K-Fold Cross validation in term of confusion metrics, when applied on whole data, sampled data (balanced) and after feature respectively. First metrics from left is with all features, middle one is result with sampled and left one is confusion metrics result with features selection.



Proposed Ensemble Method (Voting Classifier): -

Voting classifier is basically an ensemble technique which unite different base classifiers and based on result of these classifiers it computes new result. It is a technique that may be used to improve model performance, ideally achieving better performance than any single model used in the ensemble. We are using soft voting technique here which it computes probability based on probabilities and weights associated with different classifiers.

In voting classifier, we are using our best models. We are using 3 models which are random forest models, gradient boosting model and XGBOOST which were giving us best results. So, after combining all our best performing model we create a voting classifier model, which is giving us good results. It's accuracy is better than all other base model.

Features	Without feature selection	After Over Sampling	After Feature Selection
Model	Voting Classifier	Voting Classifier	Voting Classifier
Accuracy	91.0%	91.24%	92.66%
AUC Score	0.52	0.68	0.548
Specificity	0.93	0.97	0.931
Sensitivity	0.88	0.416	0.882
Recall	0.88	0.416	0.584
Precision	0.931	0.66	0.555
Miss Rate (FNR)	0.11	0.583	0.117
Miss Rate (FPR)	0.068	0.02	0.068

Model Deployment: -

The section presents the result after executing voting classifier model (GB + RF + XGBoost) and other base models such as GB, RF, LR, DT, SVM, NB and KNN.

So, here we can see from above results that random forest, Gradient boosting, and XGBoost are giving best ROC Score of 0.7 with accuracy of 90% on sampled data (balanced dataset). Accuracy of XGBoost is little higher than other algorithms. So, we can say XGB classifier as our best model. Among other base models, Random Forest is good accuracy and ROC score, same with Gradient Boosting. Decision tree is good in term of accuracy, but its ROC score is not up to the mark.

LR, SVM, NB and KNN were not performing up to the mark. Whereas XGboost is also giving good roc score with 91%% accuracy. We also implement ensemble technique model voting classifier, but its' ROC score is little lower then XGBoost model.

After considering all the results, XGBoost shows the highest performance results to predict employee attrition. So, we are deploying our model in the end, we are generating 10 random samples from original dataset (only independent features) and passing that 10 employee records from the model for prediction, and model is returning list array of employees who are expectedly leaving and staying.

Conclusion: -

The difficulty of keeping skilled workers poses a challenge to business owners. Employers can lose a lot of money due to employee attrition because it costs a lot to replace their knowledge and productivity. As a result, an ensemble model has been used to integrate several machine learning approaches to identify the various root causes of these significant business issues. Furthermore, a variety of performance metrics, including ROC, Precision, Accuracy, Recall, specificity and F1 score, have been used to identify the most efficient machine learning approaches.

So, from above results we can say that XGboost model is performing better in predicting attrition from employee's data. We can also use other model such as random forest, Gradient boosting as their performance is good too, but little less than XGboost.