Bank Customer Churn Prediction
DATA 240: Data Mining
Sai Santosh Boga

Introduction

The percentage or proportion of the company's subscribers who pull out of a contract during the active period of a contract is called Customer Churn. This metric can be considered as an indication of customer discontent or the market competition with the current policies. Customer churn rate is well coupled with the customer lifetime within the company giving an approximation of the time the customer would stay in the company. Churn analysis plays a key role in decreasing the costs of customer addition as the price of adding a new customer to the business is pricier than retaining an existing one. Moreover, the revenue from a returning or existing customer would be comparatively higher than a new one. This piece of information can prove to be valuable in taking countermeasures to retain the customers.

Financial institutions such as banks and credit companies are prone to Customer churn as well, and a timely prediction can help save significant percentages of their revenues. Few features that can help predict the churn rate are the dormancy of the account, the customer satisfaction indices, and pricing policies of the institution.

Therefore, the need to understand the customer's requirements against a service acts as the motive of this project and while also focusing on its sentiments and choices.

Why this topic?

FinTech – Financial Technology has always kept me motivated because it uses technology to facilitate regular conventional financial activities or transactions. It is the field where one gets immediate rewards when offered a feasible financial solution with the current technology. Data Science, Machine Learning and AI are widely used in Financial Institutions to identify prospective customers, find fraudulent customers, recommendations systems, etc.

This specific topic also falls under the same genre where we will be identifying or predicting the customers who are about to churn or leave the bank. Knowing the customers who are about to churn beforehand gives an added advantage to the bank. Furthermore, we can implement special offers or schemes to retain customers who are about to churn. As they say, customer retention is easy than finding a new customer.

Description of the Dataset

As the bank data source has always remained confidential; Used Kaggle's dataset in this project. The dataset contains a total of 10,000 records with 14 features or attributes.

These are the following features

1. **RowNumber**: Indicates the index of the dataset
2. **CustomerId**: A Unique Identification number for Customers
3. **Surname**: Last Name of the customers
4. **CreditScore**: Score reported by the credit bureau and ranges from (350-900)

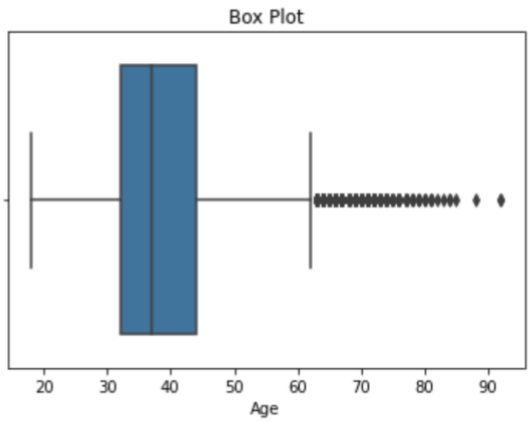
5. **Geography**: Deals with 3 locations (Germany, France, Spain)
6. **Gender**: (Male, Female)
7. **Age**: Ranging from (20-90)
8. **Tenure**: Number of years a customer is with the bank; range (1-10)
9. **Balance**: Account Balance of the customer
10. **NumberOfProducts**: Number of products customer is associated with bank(1-5)
11. **HasCrCard**: Customer has a credit card or not (0 or 1)
12. IsActiveMember: Is he an active member or not (0 or 1)
13. EstimatedSalary: Salary of the customer (Continuous Variable)
14. Exited: Target Variable; if the customer exited the bank or not (0 or 1)

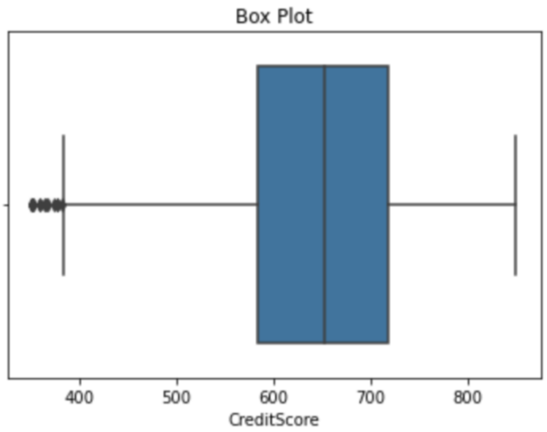
Methodology

The methodology used to solve this problem is the CRISP DM method, which follows data understanding, data cleaning, data processing, data modelling and deployment.

Data Cleaning & Data Processing

Since the dataset is from Kaggle the dataset had no null values. After thoroughly analyzing and going through all the distribution plots; I found some Age and Credit Score features anomalies. To detect these anomalies, I plotted the box plots for them and removed the outliers from the dataset.



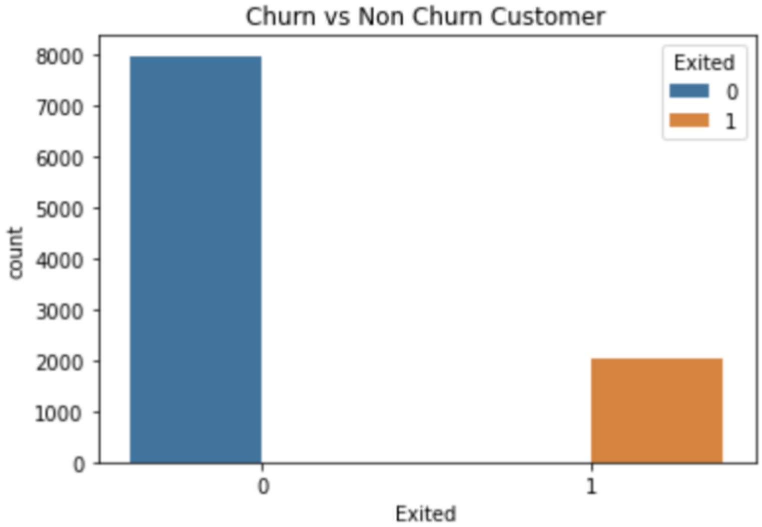


From the Age box plot, we can clearly say that people above 60 years are outliers in the dataset, and similarly for the Credit Score box plot, customers with credit scores less than 400 fall in the outliers category. Since there are considerably few outliers, I deleted the dataset's outlier records as they skew the model’s performance.

Exploratory Data Analysis

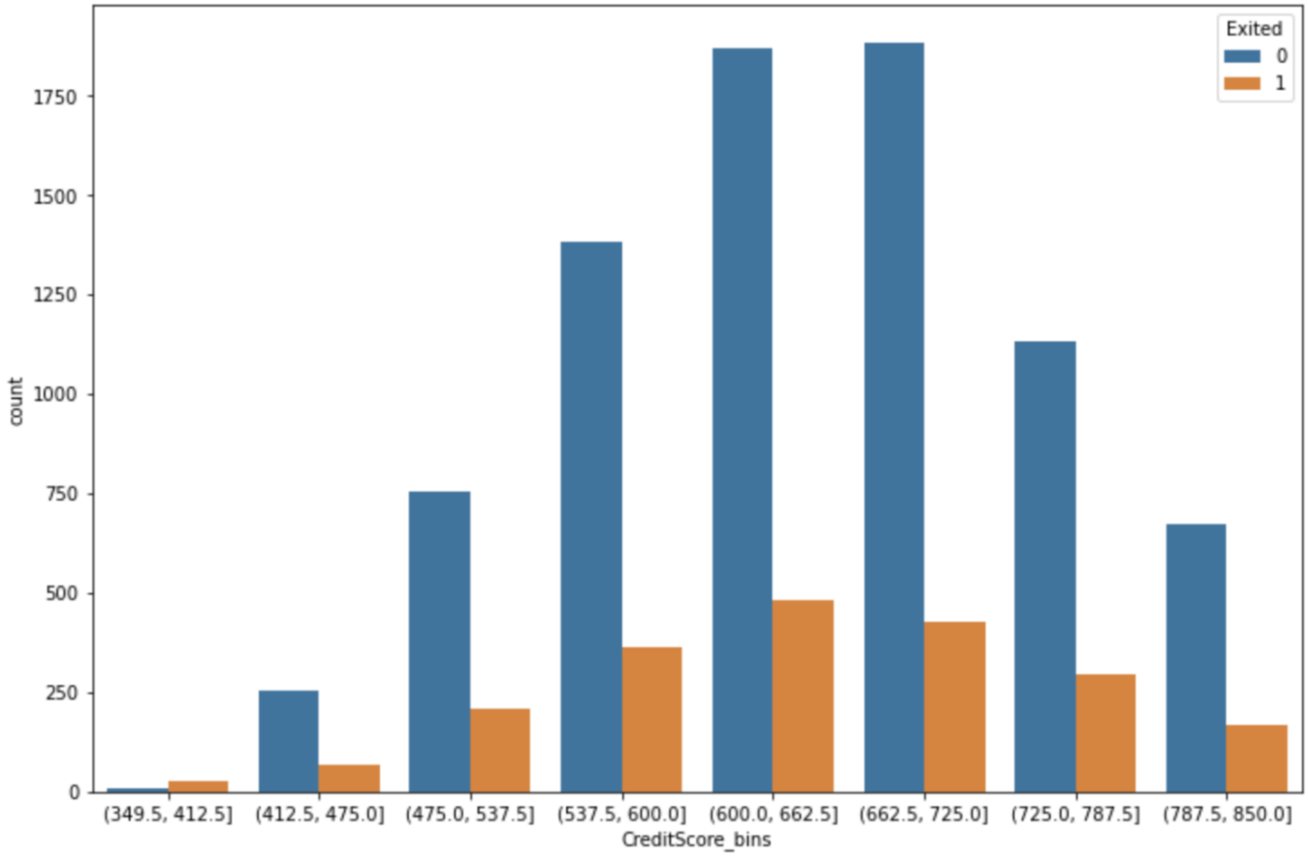
Understanding the data is crucial as it helps in understanding patterns or correlations in the data. Visualizations are the best way to understand the data as one can directly perceive the information without actually looking into the data. EDA helps in generating insights and also helps in knowing which features play an essential role in training the model.

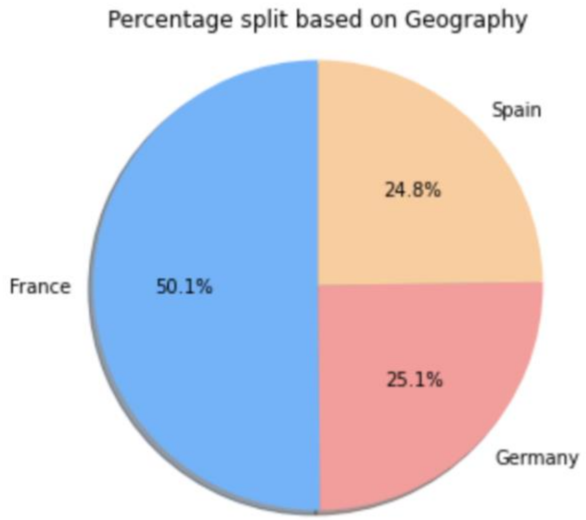
In the Bank Churn dataset performed, the following visualizations;

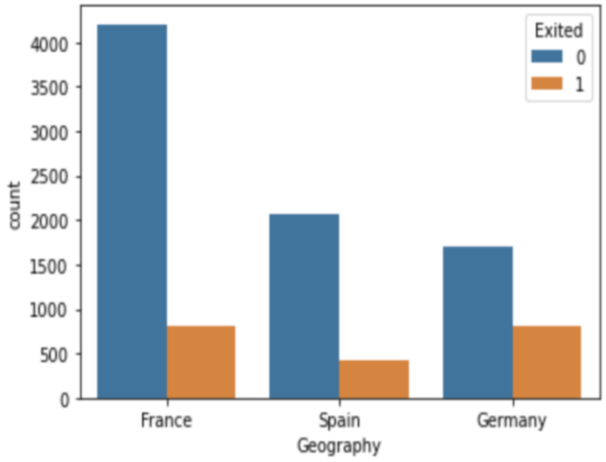


- To understand the dataset balance; initially did a count plot of Churn vs Non-Churn Customers
- The plot clearly states that the dataset is purely imbalances, and there are only 20% of churn customers

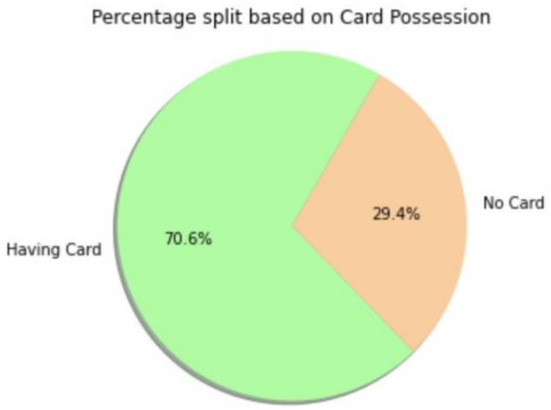
After exploring the dataset's balance, analyzed every feature with respect to the customer churn, and tried to identify specific data patterns. In the below graph; binned the credit score values into 8 different bins using pd.cut() function and plotted it against the target variable. From the graph, we can clearly say that most people who churn have credit scores ranging from 530 -720 and account to almost 55% of the dataset.

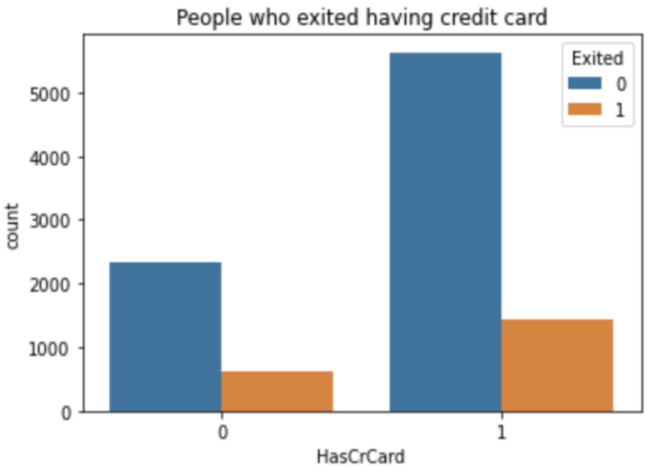




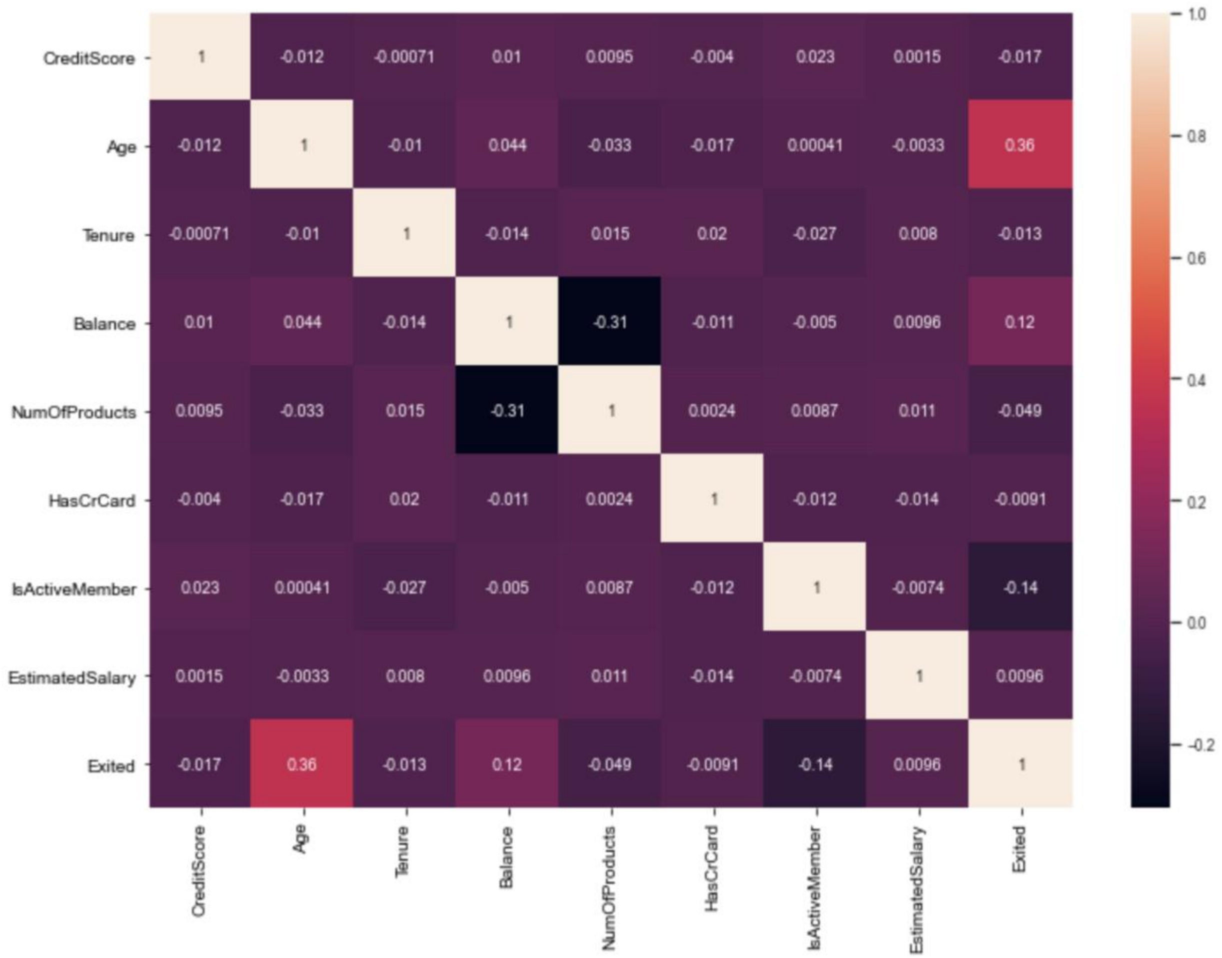


The above two plot gives us the data distribution among three different countries i.e France, Spain, and Germany. France has the highest customer base, and Germany has the highest number of churning customers.





70% of customers have credit cards. From the second graph, we can infer that; people with credit cards tend to churn or leave the bank. This is a useful insight for decision making.



The above diagram is the correlation matrix plotted using heatmap; from the diagram, we can clearly state that Age, Balance and Estimated Salary are positively correlated with the Target feature.

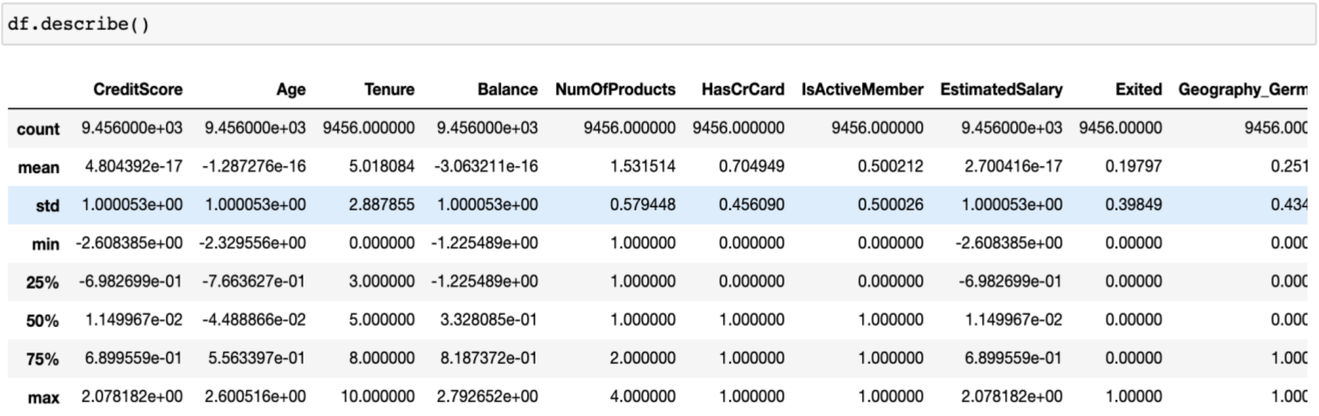
Data Pre-processing:

For features like Credit Score, Age, Balance, Estimated Salary, the values have different ranges. To scale these values to a mean scale, used StandardScalar() method from sklearn library and scaled them normally.

While categorical features like Gender and Geography are labeled using pd.get_dummies()

After analyzing and exploring the dataset, understood that RowNumber, CustomerId, SurName columns are dropped as they do not contribute to the algorithms' performance.

After preprocessing, the statistics of the dataset looks like the following



```
df.describe()
```

	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Geography_Germ
count	9.456000e+03	9.456000e+03	9456.000000	9.456000e+03	9456.000000	9456.000000	9456.000000	9.456000e+03	9456.00000	9456.00
mean	4.804392e-17	-1.287276e-16	5.018084	-3.063211e-16	1.531514	0.704949	0.500212	2.700416e-17	0.19797	0.251
std	1.000053e+00	1.000053e+00	2.887855	1.000053e+00	0.579448	0.456090	0.500026	1.000053e+00	0.39849	0.434
min	-2.608385e+00	-2.329556e+00	0.000000	-1.225489e+00	1.000000	0.000000	0.000000	-2.608385e+00	0.00000	0.00
25%	-6.982699e-01	-7.663627e-01	3.000000	-1.225489e+00	1.000000	0.000000	0.000000	-6.982699e-01	0.00000	0.00
50%	1.149967e-02	-4.488866e-02	5.000000	3.328085e-01	1.000000	1.000000	1.000000	1.149967e-02	0.00000	0.00
75%	6.899559e-01	5.563397e-01	8.000000	8.187372e-01	2.000000	1.000000	1.000000	6.899559e-01	0.00000	1.00
max	2.078182e+00	2.600516e+00	10.000000	2.792652e+00	4.000000	1.000000	1.000000	2.078182e+00	1.00000	1.00

Why does this topic need data mining method?

Understanding the relations and patterns between two or three variables can be done by plotting the scatter plots and distribution plots to understand the data trend. However, to understand the whole data pattern and predict the target variable, we need complex algorithms to work with, and these are machine learning algorithms.

Since we have only two outcomes to predict i.e., churn vs. not churn, it turns out to be a classic binary classification problem, and we have several data mining methods and models to solve this problem. In this project, I made use of algorithms like Logistic Regression, Random Forest, KNN.

I also made use of an essential data mining method that samples the dataset. There are two different types of sampling

1. Over Sampling: Sampling the minority class to match the balance the dataset
2. Under Sampling: Deleting the majority class records to match the dataset balance.

In this project, I used Over Sampling to generate synthetic data on minority dataset to balance the dataset equally. Initially, trained all of the models without sampling and then trained them with sampling, i.e., having an equal number of records for both the classes and comparing them on evaluation metrics.

Description of Data Mining Methods

Different Data Mining Methods used in this project are as follows

1. Logistic Regression
    a. Finding Best Parameter using GridSearchCV
    b. Selecting important features (using SelectFromModel using sklearn’s feature selection module) and training them back again
2. Random Forest
    a. Finding Best Parameters using RandomizedSearchCV
    b. Training with best parameter values and important features using feature importance
3. K-Nearest Neighbors
    a. Training with different k values like 2,3,4
4. Naïve Bayes
5. Over Sampling the dataset and training it with Random Forest

Logistic Regression:
For binary classification; logistic regression is considered as the baseline model to evaluate the results. It is a statistical model which uses logistic function to model a binary target variable. Logistic regression works similar to the linear or multi-linear regression; the only difference is that it uses logistic function which squishes the input to range of [0,1]. These values are then used to classify or predict according to the threshold set.

In this project, the dataset is being trained on the default parameter values of Logistic Regression. Moreover, GridSearchCV is performed to find the best parameters.

Random Forest:
Random Forest algorithm is an extension of the decision tree. These are ensemble learning methods for classification and regression. These operate by constructing multiple decision trees at training time and by outputting the class that is the mode of the class. Random Forests outperform decision trees but are prone to overfitting

Ensemble learning algorithms perform better than a single algorithm as they output the mode of all the outputs generated from multiple decision trees. For the bank churn prediction purposes this model is utilized to achieve better accuracy. Also used RandomizedSearchCV to get the best parameters for the model by random search.

K-Nearest Neighbors:
KNN is a non-parametric method used for both classification and regression. It uses K-closest training examples in the feature space to classify or predict the value or outcome.

For this project, trained with 2,3,4,5 K different values to predict the customer churn.

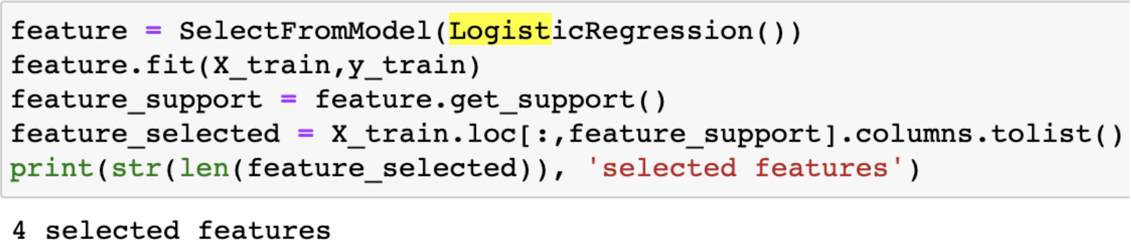
Naïve Bayes:
A simple probabilistic algorithm based on applying Bayes theorem. It assumes that features are strongly independent.
Predicting the probability of customer churn using the Naïve Bayes model

Feature Selection
Feature selection is a crucial step in all of the data science projects. It reduces the model complexity and helps the model to adjust the bias and variance tradeoff. It is equally important not to drop all of the features as the model becomes simple and biased towards a particular feature.

The dataset has 14 features {RowNumber, CustomerId, Surname, Geography, Gender, Age, Tenure, Balance, NumOfProducts, HasCrCard, IsActiveMember, EstimatedSalary, Exited }

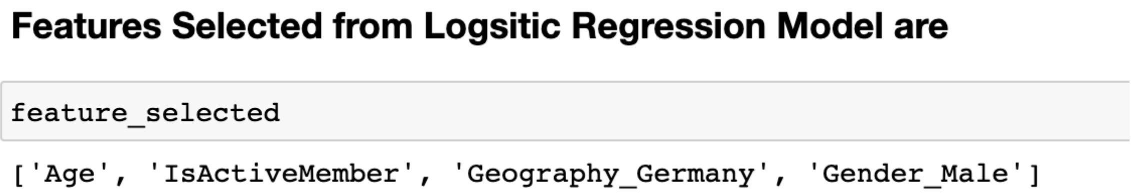
After exploring and understanding the data; intuitively removed RowNumber, CustomerId, Surname features as they don’t contribute much to predicting the target variable. The 11 features; scaled the continuous variables and encoded the categorical variables and trained models like Logistic Regression, Random Forest, Naïve Bayes, and KNN.

1. **Logistic Regression:** Important Features are found using SelectFromModel from sklearn.feature_selection module. The features selected to retrain are {Age, IsActiveMember, Geography_Germany, Gender_Male}



```
feature = SelectFromModel(LogisticRegression())
feature.fit(X_train,y_train)
feature_support = feature.get_support()
feature_selected = X_train.loc[:,feature_support].columns.tolist()
print(str(len(feature_selected)), 'selected features')
```

```
4 selected features
```



Features Selected from Logsitic Regression Model are

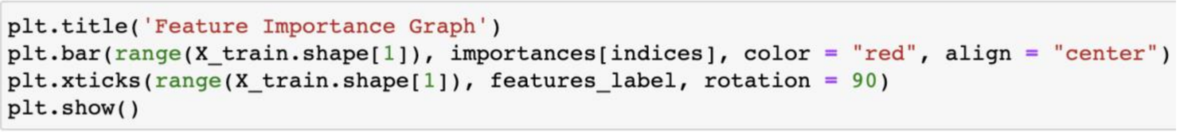
```
feature_selected
```

```
['Age', 'IsActiveMember', 'Geography_Germany', 'Gender_Male']
```

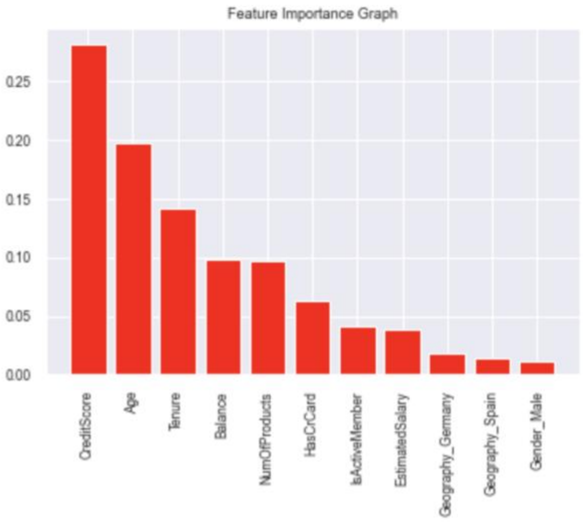
After training with the following features, the model's accuracy wasn't affected; however, the model was simplified and avoided most of the noise that weren't really needed. The accuracy obtained by training with 4 features is 82.39%

2. **Random Forest:** Best parameter is found after iteratively training using RandomizedSearchCV with cross-validation value of 3. Using the best parameter value obtained after training, important features are selected using their feature scores.

- Using Random Forest, the best feature with highest feature score is Credit Score and the following features are plotted in the below plot



```
plt.title('Feature Importance Graph')
plt.bar(range(X_train.shape[1]), importances[indices], color = "red", align = "center")
plt.xticks(range(X_train.shape[1]), features_label, rotation = 90)
plt.show()
```



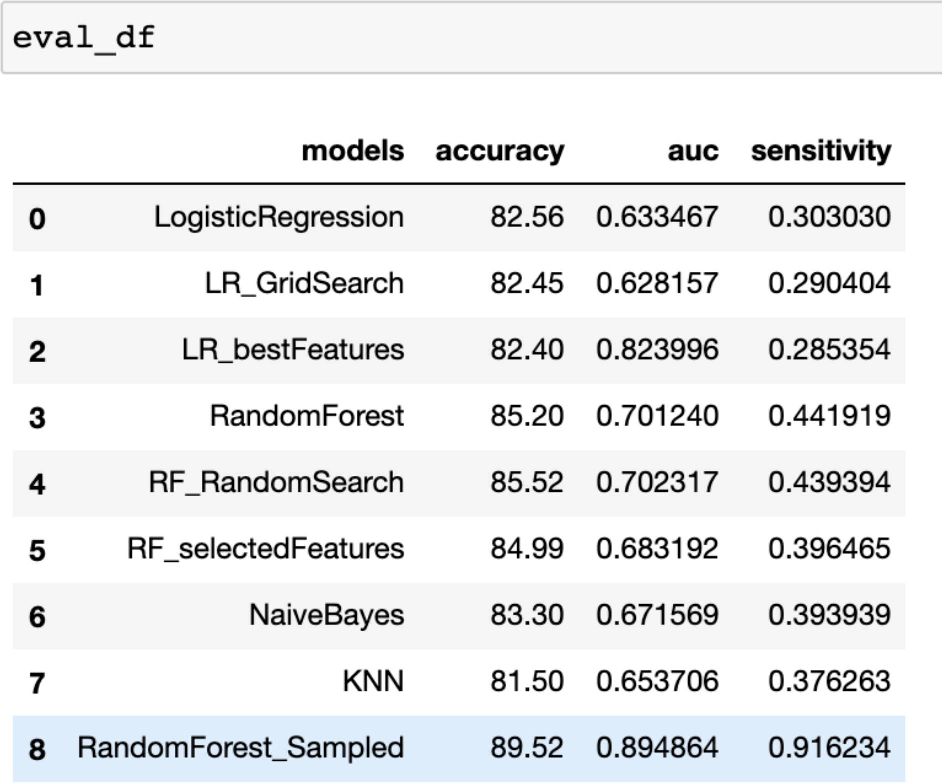
The accuracy obtained after training with features whose feature score was greater than 0.03; Random forest gave the accuracy of 84.83%

Result
Comparison between the methods

All the models trained to predict customer churn are evaluated on three metrics considering the use case specific to this domain. The models are evaluated using Accuracy, AUC (Area Under Curve) and Sensitivity.

The first seven models are trained without sampling and the results obtained are displayed in the below figure. Despite the mean accuracy of all the models between 80 and 85%, it's the sensitivity we are concerned about. Sensitivity gives the True Positive Rate (TPR), which is defined as the number of actual true customers who are about to churn and are predicted correctly.
TPR = TP/(TP+FN)

In the all of the models trained without doing sampling we got decent accuracy of the models that ranged from 82-85%. However, the sensitivities of the models were too low. The main reason behind this is having an imbalanced dataset; that is, one class dominating the other.
The target variable split for churn vs non-churn customers is 20:80, where 20% of records are of churn and 80% for non-churn customers



```
eval_df
```

	models	accuracy	auc	sensitivity
0	LogisticRegression	82.56	0.633467	0.303030
1	LR_GridSearch	82.45	0.628157	0.290404
2	LR_bestFeatures	82.40	0.823996	0.285354
3	RandomForest	85.20	0.701240	0.441919
4	RF_RandomSearch	85.52	0.702317	0.439394
5	RF_selectedFeatures	84.99	0.683192	0.396465
6	NaiveBayes	83.30	0.671569	0.393939
7	KNN	81.50	0.653706	0.376263
8	RandomForest_Sampled	89.52	0.894864	0.916234

Furthermore, to resolve data imbalance, over sampling of minority class is done using resample module. Oversampling generated synthetic data, and with this trained the model using Random

Forest as it gave the highest accuracy of all the models when trained without sampling. Accuracy and sensitivity are 89.52% and 91.6% respectively, which clearly shows the significant performance increase from all of the other models.

models	accuracy	auc	sensitivity
LogisticRegression	82.56	0.633467023	0.303030303
LR_GridSearch	82.45	0.628156566	0.29040404
LR_bestFeatures	82.4	0.823995772	0.285353535
RandomForest	85.2	0.701240345	0.441919192
RF_RandomSearch	85.52	0.702317291	0.439393939
RF_selectedFeatures	84.99	0.683192216	0.396464646
NaiveBayes	83.3	0.671568627	0.393939394
KNN	81.5	0.653706179	0.376262626
RandomForest_Sampled	89.52	0.894863871	0.916233766

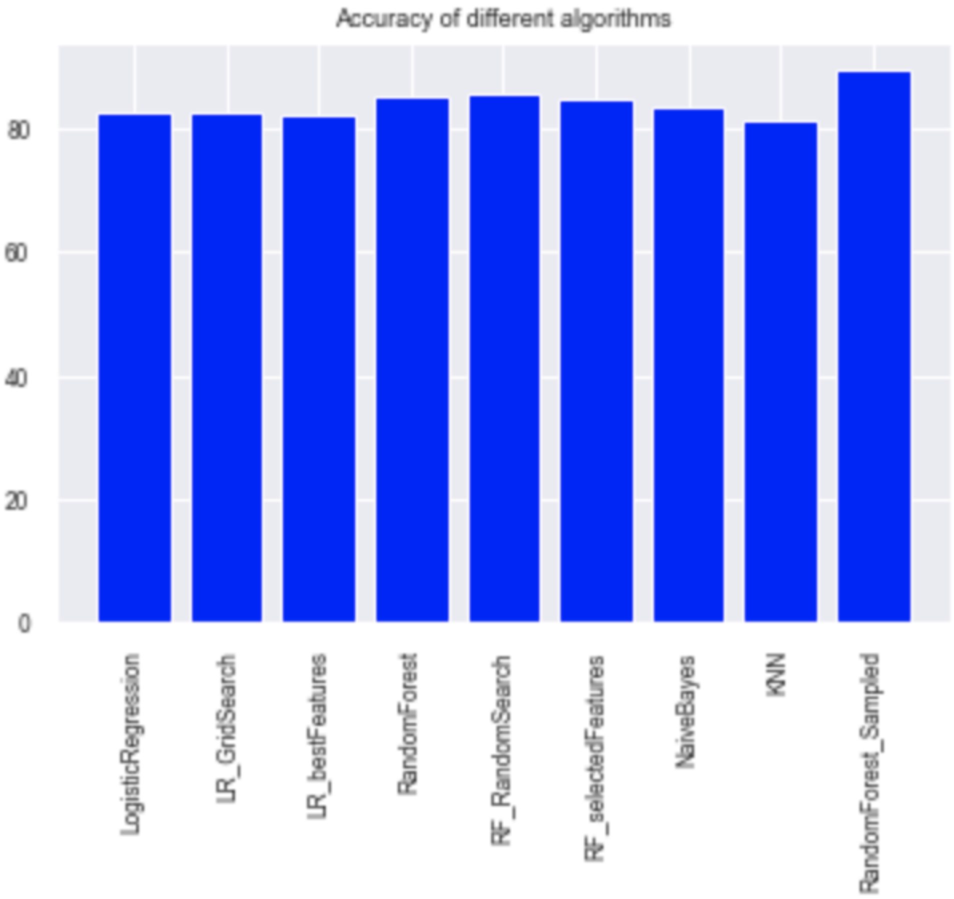


Fig: Comparison of different model accuracies

Hence after comparing all the results and metrics from all the trained models, we can say that Random Forest did the best job among all the other models. Random forest gave an accuracy of 85.2% without sampling and 89.52% with sampling.

The ROC curve for the best model for predicting Bank Customer Churn Prediction is Random Forest with a sampling of minority class and is displayed below

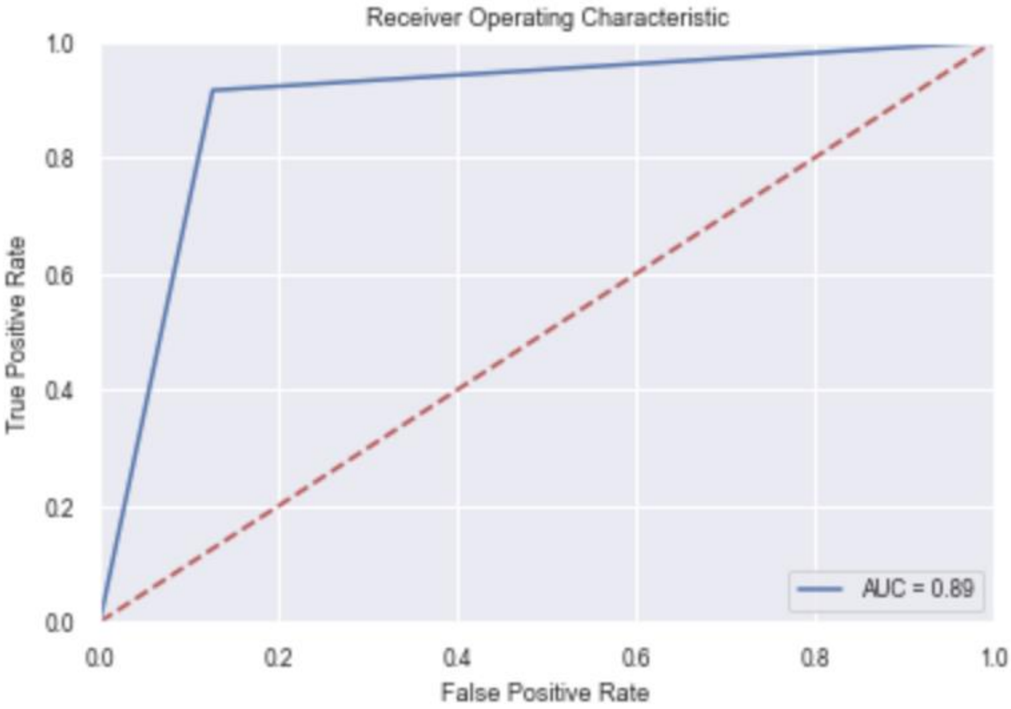


Fig: ROC-AUC Curve for the Random Forest model after sampling

Comparison between all features and selected features

models	Number of Features	accuracy	auc	sensitivity
LR_GridSearch	12	82.45	0.62815657	0.29040404
LR_bestFeatures	4	82.4	0.82399577	0.28535354
RF_RandomSearch	12	85.52	0.70231729	0.43939394
RF_selectedFeatures	9	84.99	0.68319222	0.39646465

Feature selection is really a crucial step in all of the data science projects. It reduces the model complexity and helps the model to adjust the bias and variance tradeoff. It is equally important not to drop all of the features as the model becomes simple and biased towards a particular feature.

- From the above results, LR_GridSearch and LR_bestFeatures(Logistic Regression with best features) almost have the same metric values, and LR_bestFeatures have a better AUC value than that of LR_GridSearch

- Similarly, for Random Forest, both the models have almost equaled metric values. However, in this scenario, Random Forest with Randomized Search did a slightly better job than the selected features model considering AUC and sensitivity.

Discussion

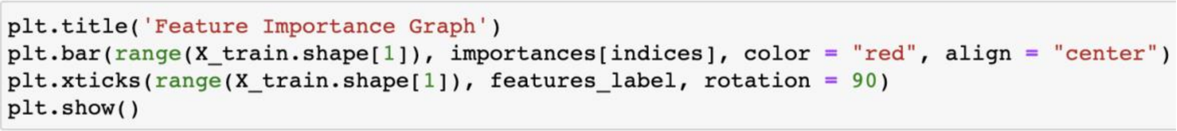
Why one method is better than the other?

Considering the results, one can clearly understand that Random Forest with Random Search performed much better than all the other models. Random Search iterates through all the parameters randomly and finds the best parameter for the model. Furthermore, in this scenario, Random Forest did a better job after generating synthetic data by over-sampling.

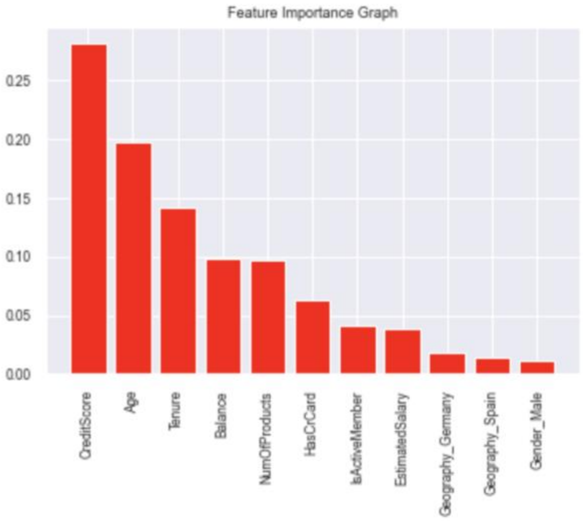
- Random Forest with Random Search is better than all the other models trained in this project because Random forest is an ensemble method and it outputs the mode of all predictions made by the multitude of decision trees. Ensemble methods are effective and efficient because of this reason.
- Sampling techniques are useful in generating more data or deleting data from the dataset. They help in creating balanced datasets. In this project, the sampling technique did an excellent job at improving accuracy and sensitivity and reducing the False Negatives.
    - The main reason for the model to perform well is that initially, it had very little to learn the features. However, when records are resampled model did an excellent job at learning the features and provided good accuracy, AUC and sensitivity scores.

Difference between all features and selected features

There’s no significant difference when the model is trained with all the features and selected features as explained above. However, there is a subtle difference in metric values as shown in the comparison table. By selecting only few features, accuracy didn’t change much, but AUC score and sensitivity have increased.



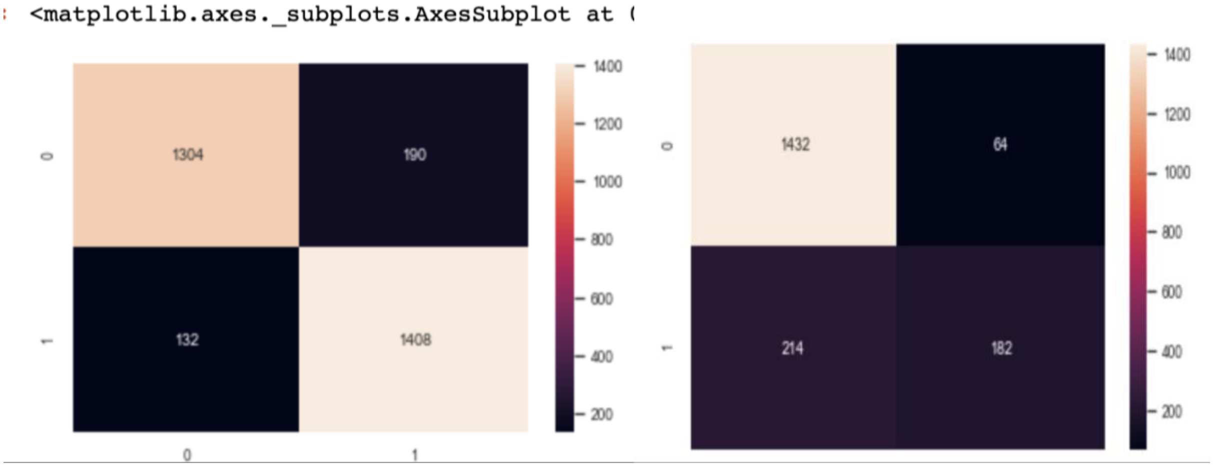
```
plt.title('Feature Importance Graph')
plt.bar(range(X_train.shape[1]), importances[indices], color = "red", align = "center")
plt.xticks(range(X_train.shape[1]), features_label, rotation = 90)
plt.show()
```



What is the meaning of your result? How to explain your result based on domain knowledge?

The final results for the best model using two different methods i.e, with sampling and without sampling, are displayed below

Model	Method/Technique	Metrics	Values
RandomForest	Without Sampling	Accuracy	85.41
		AUC	0.70
		Sensitivity	0.44
RandomForest	With Sampling	Accuracy	89.39
		AUC	0.89
		Sensitivity	0.91



```
: <matplotlib.axes._subplots.AxesSubplot at (
```

Fig: Confusion Matrices for (i) With sampling and (ii) Without Sampling

The confusion matrices represent the data in the following way as demonstrated in the below table

	Predicted Values	
Actual Values	**Non-Churn(N)**	**Churn(P)**
Non-Churn(N)	TN	FP
Churn(P)	FN	TP

Interpreting the results from the two confusion matrices, we can clearly see that both the models have decent accuracies. However, it is not the accuracy that we consider to determine the performance of the model. The second row in both the confusion matrices represent data about the customers who will churn and this was the project goal from the beginning.

However, observing the confusion matrix on the right, the one without sampling, it is clearly evident that 214 churn customers are misclassified as not-churn customers as False Negatives. Misclassifying churn customers into non-churn is not acceptable as the bank will be losing customers in this way. However, misclassifying non-churn customers as churn is not a problem because in that scenario, the bank might provide attention towards these customers, which anyways increase customer satisfaction.

Understanding the problem domain and reducing the number of False Negatives is the key to achieving a better performing model. The confusion matrix obtained without sampling has high False Negative(FN) values, whereas the confusion matrix obtained with sampling has fewer FN values. For this reason, considered TPR or Sensitivity as the evaluation metric for this Bank customer churn problem.