# Topic: Bank Customer Churn Prediction

- **Churn prediction means detecting which customers are likely to leave a service or to cancel a subscription to a service. It is a critical prediction for many businesses because acquiring new clients often costs more than retaining existing ones.**
- **Dataset** : https://www.kaggle.com/datasets/shantanudhakadd/bank-customer-churn-prediction
- **About the Dataset :**
- *It is the dataset of a U.S. bank customer for getting the information that, this particular customer will leave bank or not. The dataset contains 10,000 rows and 14 columns.*

```
df.head()
```

| | RowNumber | CustomerId | Surname | CreditScore | Geography | Gender | Age | Tenure | Balance | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 15634602 | Hargrave | 619 | France | Female | 42 | 2 | 0.00 | 1 | 1 | 1 | 101348.88 | 1 |
| 1 | 2 | 15647311 | Hill | 608 | Spain | Female | 41 | 1 | 83807.86 | 1 | 0 | 1 | 112542.58 | 0 |
| 2 | 3 | 15619304 | Onio | 502 | France | Female | 42 | 8 | 159660.80 | 3 | 1 | 0 | 113931.57 | 1 |
| 3 | 4 | 15701354 | Boni | 699 | France | Female | 39 | 1 | 0.00 | 2 | 0 | 0 | 93826.63 | 0 |
| 4 | 5 | 15737888 | Mitchell | 850 | Spain | Female | 43 | 2 | 125510.82 | 1 | 1 | 1 | 79084.10 | 0 |

```
df.info()
```

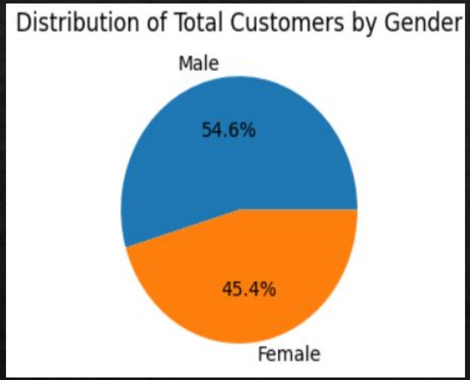
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   RowNumber        10000 non-null  int64
 1   CustomerId       10000 non-null  int64
 2   Surname          10000 non-null  object
 3   CreditScore      10000 non-null  int64
 4   Geography        10000 non-null  object
 5   Gender           10000 non-null  object
 6   Age              10000 non-null  int64
 7   Tenure           10000 non-null  int64
 8   Balance          10000 non-null  float64
 9   NumOfProducts    10000 non-null  int64
 10  HasCrCard        10000 non-null  int64
 11  IsActiveMember   10000 non-null  int64
 12  EstimatedSalary  10000 non-null  float64
 13  Exited           10000 non-null  int64
dtypes: float64(2), int64(9), object(3)
```

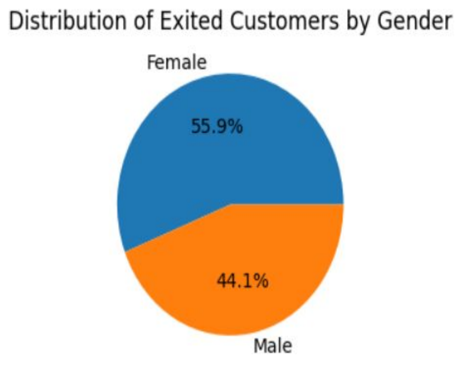
```
df.describe()
```

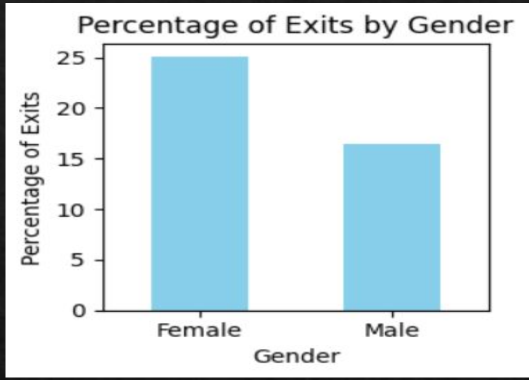
| | RowNumber | CustomerId | CreditScore | Age | Tenure | Balance | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited |
|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 10000.00000 | 1.000000e+04 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.00000 | 10000.000000 | 10000.000000 | 10000.000000 |
| mean | 5000.50000 | 1.569094e+07 | 650.528800 | 38.921800 | 5.012800 | 76485.889288 | 1.530200 | 0.70550 | 0.515100 | 100090.239881 | 0.203700 |
| std | 2886.89568 | 7.193619e+04 | 96.653299 | 10.487806 | 2.892174 | 62397.405202 | 0.581654 | 0.45584 | 0.499797 | 57510.492818 | 0.402769 |
| min | 1.00000 | 1.556570e+07 | 350.000000 | 18.000000 | 0.000000 | 0.000000 | 1.000000 | 0.00000 | 0.000000 | 11.580000 | 0.000000 |
| 25% | 2500.75000 | 1.562853e+07 | 584.000000 | 32.000000 | 3.000000 | 0.000000 | 1.000000 | 0.00000 | 0.000000 | 51002.110000 | 0.000000 |
| 50% | 5000.50000 | 1.569074e+07 | 652.000000 | 37.000000 | 5.000000 | 97198.540000 | 1.000000 | 1.00000 | 1.000000 | 100193.915000 | 0.000000 |
| 75% | 7500.25000 | 1.575323e+07 | 718.000000 | 44.000000 | 7.000000 | 127644.240000 | 2.000000 | 1.00000 | 1.000000 | 149388.247500 | 0.000000 |
| max | 10000.00000 | 1.581569e+07 | 850.000000 | 92.000000 | 10.000000 | 250898.090000 | 4.000000 | 1.00000 | 1.000000 | 199992.480000 | 1.000000 |

# Data Visualization

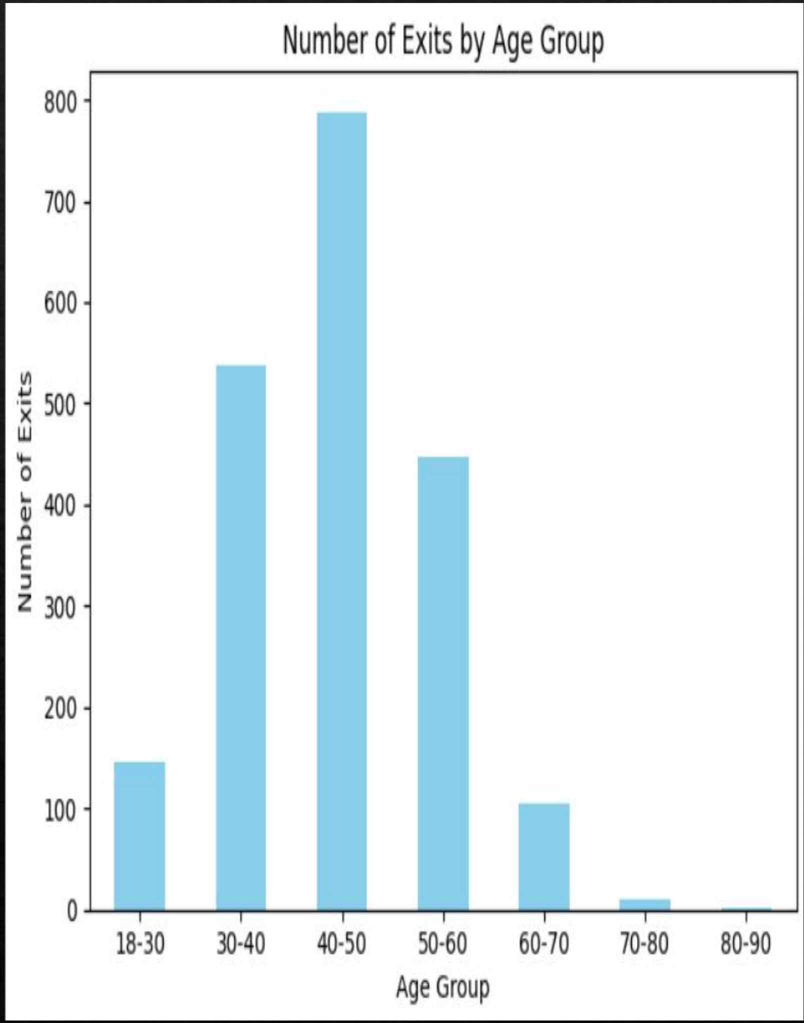
## Customer Churn based on Gender



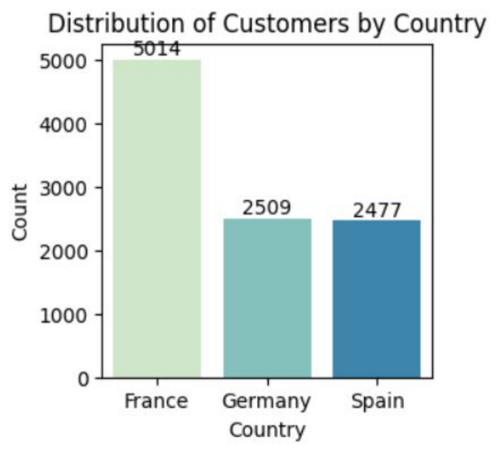



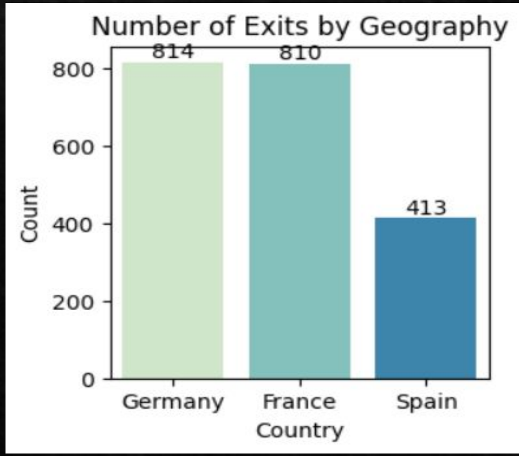



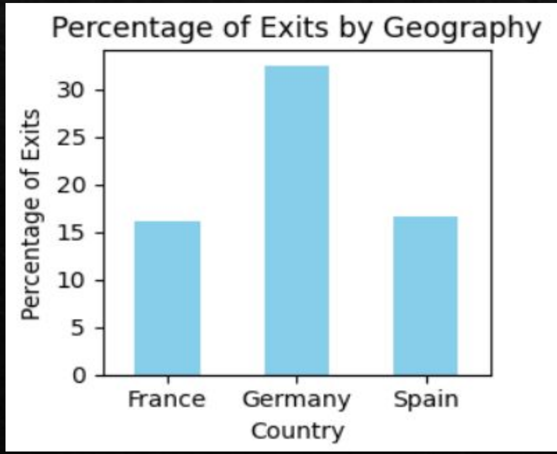

## Customer Churn Based on Age

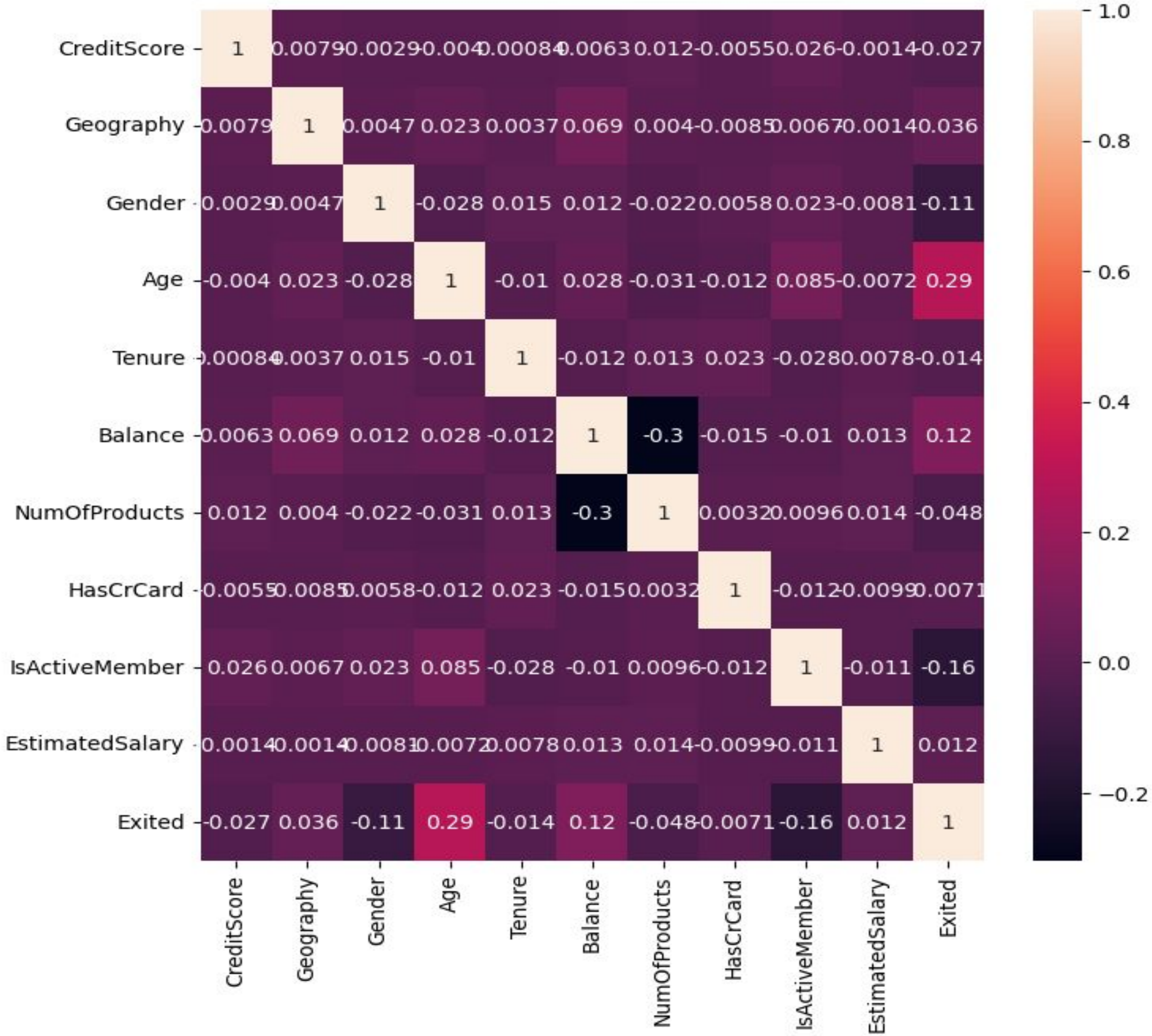



## Customer Churn based on Geography






# Correlation Heatmap

# Data Cleaning

- Removed ‘RowNumber’, ‘CustomerID’ and ‘Surname’ columns from the dataset.
- Data does not contain any missing values
- Encoded categorical columns : ‘Gender’ and ‘Geography’

Removing unnecessary columns

```
[ ] clean_data = df.drop(['RowNumber', 'CustomerId','Surname'], axis=1)
```

Checking Null Values

```
clean_data.isna().sum()
```

```
CreditScore        0
Geography          0
Gender             0
Age                0
Tenure             0
Balance            0
NumOfProducts      0
HasCrCard          0
IsActiveMember     0
EstimatedSalary    0
Exited             0
dtype: int64
```

Label Encoding Categorical Data

```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
clean_data['Geography'] = encoder.fit_transform(clean_data['Geography'])
clean_data['Gender'] = encoder.fit_transform(clean_data['Gender'])
```

```
x = clean_data.drop('Exited', axis=1)
y = clean_data['Exited']
```

```
x.head()
```

| | CreditScore | Geography | Gender | Age | Tenure | Balance | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 619 | 0 | 0 | 42 | 2 | 0.00 | 1 | 1 | 1 | 101348.88 |
| 1 | 608 | 2 | 0 | 41 | 1 | 83807.86 | 1 | 0 | 1 | 112542.58 |
| 2 | 502 | 0 | 0 | 42 | 8 | 159660.80 | 3 | 1 | 0 | 113931.57 |
| 3 | 699 | 0 | 0 | 39 | 1 | 0.00 | 2 | 0 | 0 | 93826.63 |
| 4 | 850 | 2 | 0 | 43 | 2 | 125510.82 | 1 | 1 | 1 | 79084.10 |

**train-test split**

```
[ ] from sklearn.model_selection import train_test_split
    x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.2, cv=5)
```

# Models Used

## K Nearest Neighbours:

Mean cross validation score: 0.7963

Recall: 0.0

## Logistic Regression:

mean cross validation score: 0.7904

Recall: 0.063

## Support Vector Machine:

Mean cross validation score: 0.7963

Recall: 0.0

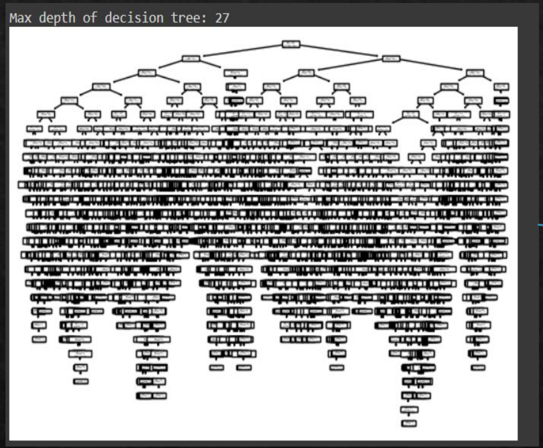
## Decision Tree:

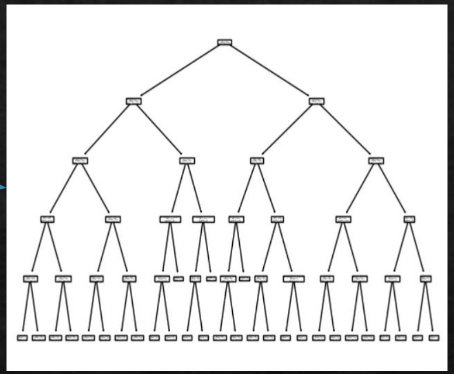
Mean cross validation score: 0.854

Recall: 0.4370

**Hyper-parameter tuning:**

On reducing max_depth of tree from 27 to 5 accuracy on testing data increased from 0.79 to 0.86.



## Random Forest:

Mean cross validation score: 0.7804

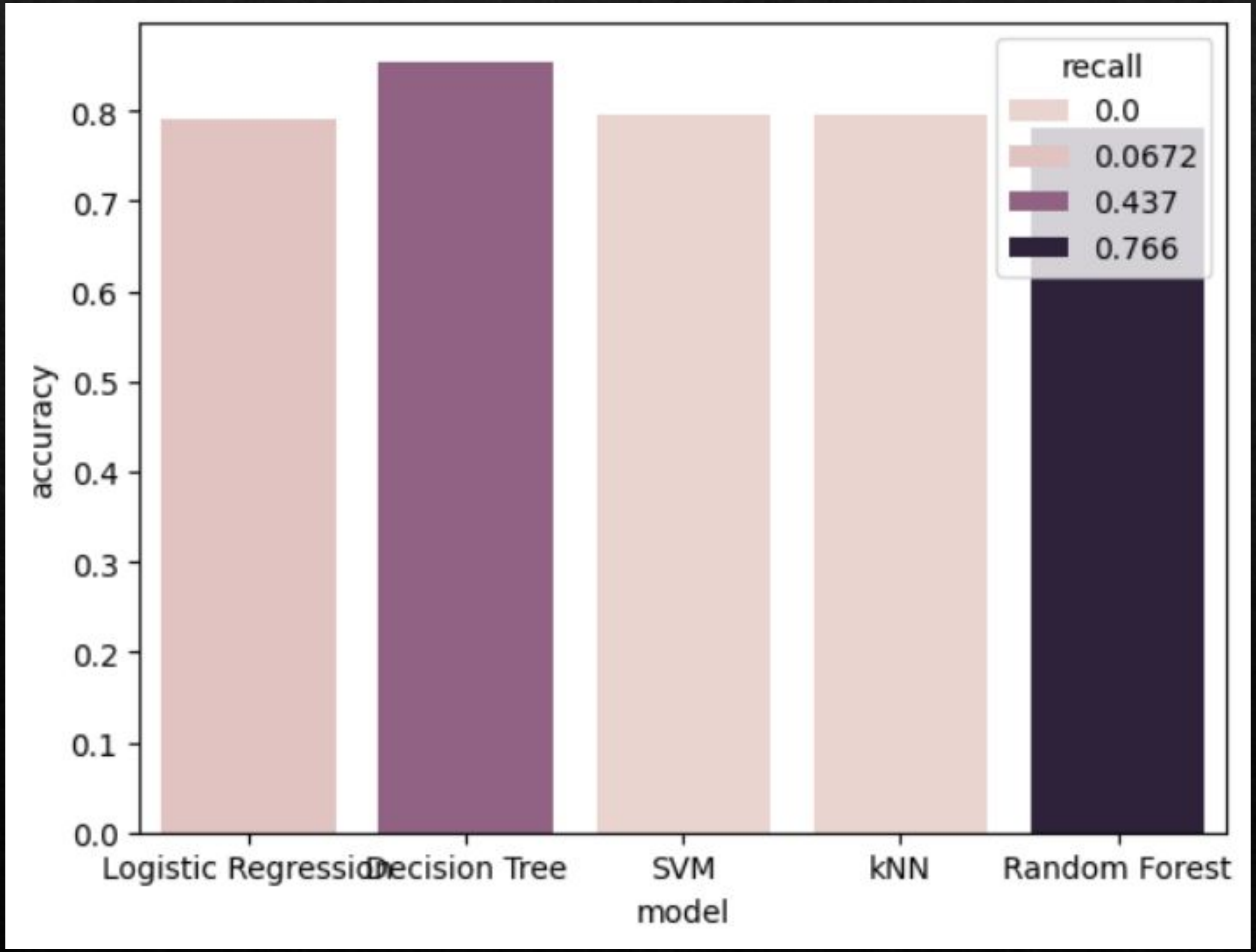
Recall: 0.766

**Hyper-parameter tuning:**

Improved recall from 0.41 to 0.766 by assigning **class weights 0.1 to exited and 0.9 to not exited classes** since the dataset contained greater proportion of not-exited class.

Also max_depth = 10 and n_iterations = 18 gave greater accuracy.

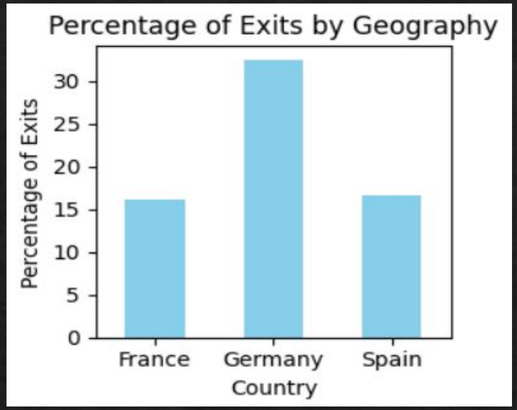
# Comparison Between Models

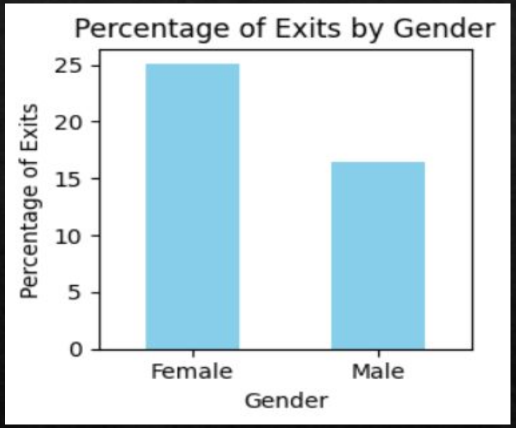


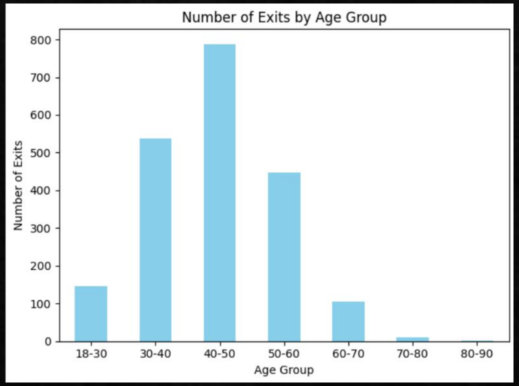

Recall and accuracy have been used to evaluate the models. Since predicting exiting customers accurately is more important than predicting not exiting customers, we would select the model that provides greater recall and satisfactory accuracy. As Random Forest provides greater recall i.e. 0.766, it predicts customer churn more accurately than other models.

# Managerial Implications

- Churn prediction means detecting which customers are likely to leave a service or to cancel a subscription to a service. It is a critical prediction for many businesses because acquiring new clients often costs more than retaining existing ones. The model helps predict exiting customers accurately with a greater recall.
- It has been observed that German customers, women and people in age group 40-50 have a higher percentage of exiting customers. Actions can be taken to increase their retention.



# Novelty

- Many churn models prioritize accuracy, but in this case, finding as many at-risk customers as possible (high recall) is crucial. Therefore more importance has been given to higher recall along with obtaining a satisfactory accuracy.

# References


References:
1) https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/
2) https://medium.com/@rithpansanga/improving-precision-and-recall-in-machine-learning-tips-and-techniques-acb5a5fd27a6#:~:text=Implementing%20class%20weights%20