Customer Churn Prediction Analysis using Ensemble Techniques

Thesis submitted in partial fulfilment

of the requirements of the degree of

Masters in Science with Specialization in

Artificial Intelligence

by

Ankit Javeri

Roll Number - 01

G.R. Number - 3511575

Under the Supervision of

Prof. Suraj

April 2023

Nagindas Khandwala College(Autonomous)
Malad, Mumbai 400064

CERTIFICATE

This is to certify that the dissertation entitled **"Customer Churn Prediction Analysis using Ensemble Techniques"** is a bonafide work of "**Ankit Javeri" (Roll No: 01 and G.R. No: 3511575)** submitted to the Nagindas Khandwala College(Autonomous),Mumbai in partial fulfillment of the requirement for the award of the degree of **"Masters in Science with Specialization in Artificial Intelligence"**.

Internal-Examiner

(Prof.Suraj)

External Examiner

Supervisor's Certificate

This is to certify that the dissertation entitled **"Customer Churn Prediction Analysis using Ensemble Techniques"** submitted by **Ankit Javeri, Roll No: 01** and **G.R. No: 3511575,** is a record of original work carried out by him/her under my supervision and guidance in partial fulfillment of the requirements of the degree of **Masters in Science with Specialization in Artificial Intelligence** at Nagindas Khandwala College(Autonomous), Mumbai 400064 . Neither this dissertation nor any part of it has been submitted earlier for any degree or diploma to any institute or university in India or abroad.

Internal Examiner

(Prof. Suraj)

Declaration of Originality

I, **Ankit Javeri, Roll No: 01** and **G.R. No: 3511575**, hereby declare that this dissertation entitled "**Customer Churn Prediction Analysis using Ensemble Techniques"** presents my original work carried out as a Master Student of Nagindas Khandwala College(Autonomous),Mumbai 400064. To the best of my knowledge, this dissertation contains no material previously published or written by another person, nor any material presented by me for the award of any degree or diploma of Nagindas Khandwala College(Autonomous), Mumbai or any other institution. Works of other authors cited in this dissertation have been duly acknowledged under the sections "Reference" or "Bibliography". I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission.

I am fully aware that in case of any non-compliance detected in future, the Academic Council of Nagindas Khandwala College(Autonomous), Mumbai may withdraw the degree awarded to me on the basis of the present dissertation.

Date:

Place:

Ankit Javeri

Abstract

To build a model such that it is able to predict the customer churn using different machine learning models.

Customer churn (or customer attrition) is a tendency of customers to abandon a brand and stop being a paying client of a particular business. The percentage of customers that discontinue using a company's products or services during a particular time period is called a customer churn (attrition) rate. One of the ways to calculate a churn rate is to divide the number of customers lost during a given time interval by the number of acquired customers, and then multiply that number by 100 percent. For example, if you got 150 customers and lost three last month, then your monthly churn rate is 2 percent.

Churn rate is a health indicator for businesses whose customers are subscribers and paying for services on a recurring basis, notes head of data analytics department at ScienceSoft Alex Bekker, "Customers [of subscription-driven businesses] opt for a product or a service for a particular period, which can be rather short – say, a month. Thus, a customer stays open for more interesting or advantageous offers. Plus, each time their current commitment ends, customers have a chance to reconsider and choose not to continue with the company. Of course, some natural churn is inevitable, and the figure differs from industry to industry. But having a higher churn figure than that is a definite sign that a business is doing something wrong."

There are many things brands may do wrong, from complicated onboarding when customers aren't given easy-to-understand information about product usage and its capabilities to poor communication, e.g. the lack of feedback or delayed answers to queries. Another situation: Longtime clients may feel unappreciated because they don't get as many bonuses as the new ones.

Table of Contents

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

Nobody in business enjoys seeing good customers go elsewhere. Obtaining new customers is typically the primary focus of a company in its early stages. Subsequently, the company expands its operations by providing existing customers with a broader range of products or working to increase the frequency with which they buy those products

If everything continues to go according to plan, there will come a time when the company will reach a size where it will need to decide on a slightly more defensive strategy and concentrate on maintaining relationships with its existing clientele. Despite providing the best user experience possible, there will always be a subset of customers who are dissatisfied and opt to stop using the service.

1.2 PROBLEM STATEMENT:

Bank has been observing a lot of customers closing their accounts or switching to competitor banks over the past couple of quarters. As such, this has caused a huge dent in the quarterly revenues and might drastically affect annual revenues for the ongoing financial year, causing stocks to plunge and market cap to reduce by X %. A team of business, product, engineering and data science folks have been put together to arrest this slide.

1.3 OBJECTIVE:

Can we build a model to predict, with a reasonable accuracy, the customers who are going to churn in the near future? Being able to accurately estimate when they are going to churn will be an added bonus

From a Biz team/Product Manager's perspective :

(1) Business goal: Arrest slide in revenues or loss of active bank customers

(2) Identify data source: Transactional systems, event-based logs, Data warehouse (MySQL DBs, Redshift/AWS), Data Lakes, NoSQL DBs

(3) Audit for data quality: De-duplication of events/transactions, Complete or partial absence of data for chunks of time in between, Obscuring PII (personal identifiable information) data

(4) Define business and data-related metrics: Tracking of these metrics over time, probably through some intuitive visualizations

(i) Business metrics: Churn rate (month-on-month, weekly/quarterly), Trend of avg. number of products per customer,

%age of dormant customers, Other such descriptive metrics

(ii) Data-related metrics : F1-score, Recall, Precision

Recall = TP/(TP + FN)

Precision = TP/(TP + FP)

F1-score = Harmonic mean of Recall and Precision

where, TP = True Positive, FP = False Positive and FN = False Negative

(5) Prediction model output format : Since this is not going to be an online model, it doesn't require deployment. Instead, periodic (monthly/quarterly) model runs could be made and the list of customers, along with their propensity to churn shared with the business (Sales/Marketing) or Product team

(6) Action to be taken based on model's output/insights : Based on the output obtained from Data Science team as above, various business interventions can be made to save the customer from getting churned. Customer-centric bank offers, getting in touch with customers to address grievances etc. Here, also Data Science team can help with basic EDA to highlight different customer groups/segments and the appropriate intervention to be applied against them

CHAPTER 2: LITERATURE SURVEY

Different customer dataset will have different customer churn predictions and based on the machine learning model being used it is able to predict the customer churn prediction.

Customer churn (or customer attrition) is a tendency of customers to abandon a brand and stop being a paying client of a particular business. The percentage of customers that discontinue using a company's products or services during a particular time period is called a customer churn (attrition) rate.

Recursive feature elimination (RFE) is the process of selecting features sequentially, in which features are removed one at a time, or a few at a time, iteration after iteration.

RFE initial steps:

Train a machine learning model

Derive feature importance

Remove least important feature(s)

Re-train the machine learning model on the remaining features

Impact of customer churn on businesses

A new customer than to retain an existing one, businesses with high churn rates will quickly find themselves in a financial hole as they have to devote more and more resources to new customer acquisition."

Many surveys focusing on customer acquisition and retention costs are available online. According to this one by Invesp, conversion rate optimization company, getting a new customer may cost up to five times more than retaining an existing customer.

Churn rates do correlate with lost revenue and increased acquisition spend. In addition, they play a more nuanced role in a company's growth potential, continues Michael,"Today's buyers aren't shy about sharing their experiences with vendors through channels like review sites and social media, as well as peer-to-peer networks. HubSpot Research found that 49 percent of buyers reported sharing an experience they had with a company on social media. In a world of eroding trust in businesses, word of mouth plays a more critical role in the buying process than ever before. From the same HubSpot

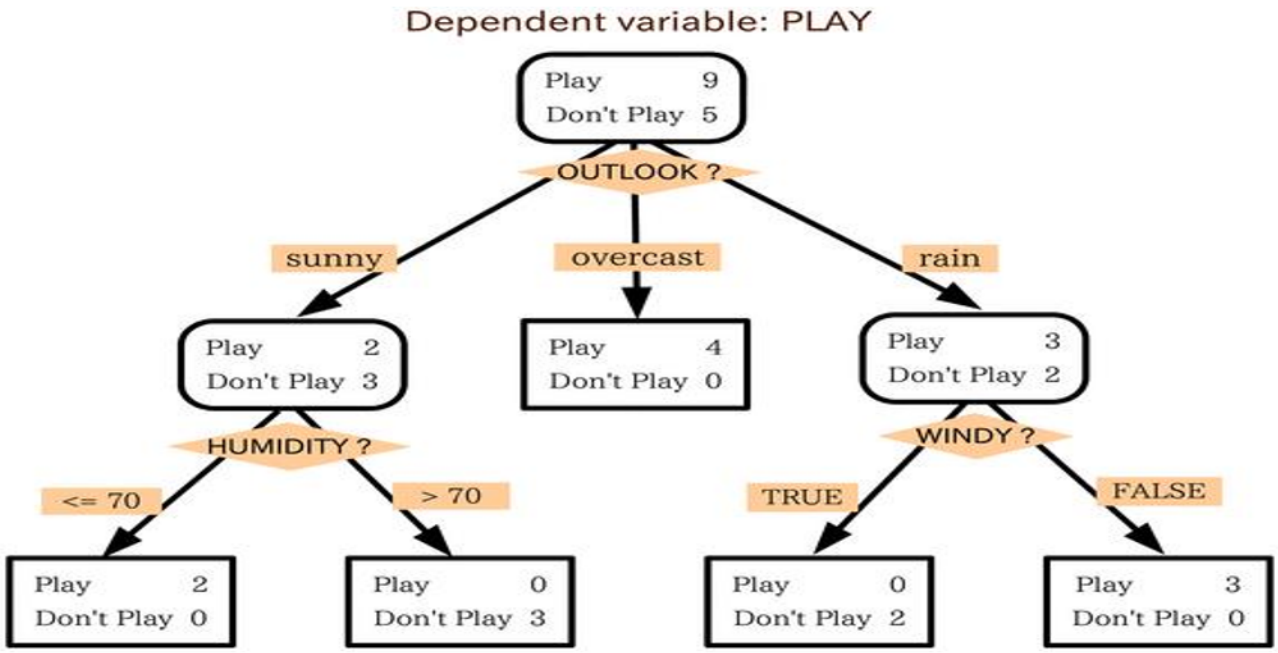
Music and video streaming services are probably the most commonly associated with the subscription business model (Netflix, YouTube, Apple Music, Google Play, Spotify, Hulu, Amazon Video, Deezer, etc.).

Media. Digital presence is a must among the press, so news companies offer readers digital subscriptions besides print ones (Bloomberg, The Guardian, Financial Times, The New York Times, Medium etc.).

Telecom companies (cable or wireless). These companies may provide a full range of products and services, including wireless network, internet, TV, cell phone, and home phone services (AT&T, Sprint, Verizon, Cox Communications, etc.). Some specialize in mobile telecommunications (China Mobile, Vodafone, T-Mobile, etc.).

Software as a service providers. The adoption of cloud-hosted software is growing. According to Gartner, the SaaS market remains the largest segment of the cloud market. Its revenue is expected to grow 17.8 percent and reach $85.1 billion in 2019. The product range of SaaS providers is extensive: graphic and video editing (Adobe Creative Cloud, Canva), accounting (Sage 50cloud, FreshBooks), eCommerce (BigCommerce, Shopify), email marketing (MailChimp, Zoho Campaigns), and many others.

Ensemble methods is a machine learning technique that combines several base models in order to produce one optimal predictive model. To better understand this definition lets take a step back into ultimate goal of machine learning and model building. This is going to make more sense as I dive into specific examples and why Ensemble methods are used.



CHAPTER 3 : METHEDOLOGY

3.1 INTRODUCTION

Bank has been observing a lot of customers closing their accounts or switching to competitor banks over the past couple of quarters. As such, this has caused a huge dent in the quarterly revenues and might drastically affect annual revenues for the ongoing financial year, causing stocks to plunge and market cap to reduce by X %. A team of business, product, engineering and data science folks have been put together to arrest this slide.

Objective : Can we build a model to predict, with a reasonable accuracy, the customers who are going to churn in the near future? Being able to accurately estimate when they are going to churn will be an added bonus

Definition of churn : A customer having closed all their active accounts with the bank is said to have churned. Churn can be defined in other ways as well, based on the context of the problem. A customer not transacting for 6 months or 1 year can also be defined as to have churned, based on the business requirements

Ensemble methods are techniques that create multiple models and then combine them to produce improved results. Ensemble methods usually produces more accurate solutions than a single model would

3.2 REQUIREMNETS

Hardware Used: -

1. Windows 10 (x64) Architecture / Linux OS(X64) / MAC OS
2. Processor AMD / Intel i3
3. Graphic Memory 1GB
4. RAM: 4GB

Software and Tools Used: -

1. Google Colab Online Tool
2. ML Libraries

3.3 METHEDOLOGY

Import Libraries:

```
!pip install ipython==7.22.0

!pip install joblib==1.0.1

!pip install lightgbm==3.3.1

!pip install matplotlib==3.3.4

!pip install numpy==1.20.1

!pip install pandas==1.3.5

!pip install scikit_learn==0.24.1

!pip install seaborn==0.11.1

!pip install shap==0.40.0

!pip install xgboost==1.5.1

!pip install scikit_learn==0.24.1

%matplotlib inline

## Import required libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

## Get multiple outputs in the same cell

from IPython.core.interactiveshell import InteractiveShell

InteractiveShell.ast_node_interactivity = "all"

## Ignore all warnings

import warnings

warnings.filterwarnings('ignore')
```

```
warnings.filterwarnings(action='ignore', category=DeprecationWarning)


## Display all rows and columns of a dataframe instead of a truncated version

from IPython.display import display

pd.set_option('display.max_columns', None)

pd.set_option('display.max_rows', None)

## Reading the dataset

from google.colab import drive

drive.mount('/content/drive')

df = pd.read_csv("/content/drive/MyDrive/churn/Churn_Modelling.csv")

df.shape

df.head(10).T
```

Basic EDA:

```
df.describe() # Describe all numerical columns

df.describe(include = ['O']) # Describe all non-numerical/categorical columns


## Checking number of unique customers in the dataset

df.shape[0], df.CustomerId.nunique()


df_t = df.groupby(['Surname']).agg({'RowNumber':'count', 'Exited':'mean'}

                                   ).reset_index().sort_values(by='RowNumber', ascending=False)


df_t.head()df.Geography.value_counts(normalize=True)
```

Discard row number

Discard CustomerID as well, since it doesn't convey any extra info. Each row pertains to a unique customer

Based on the above, columns/features can be segregated into non-essential, numerical, categorical and target variables

In general, CustomerID is a very useful feature on the basis of which we can calculate a lot of user-centric features. Here, the dataset is not sufficient to calculate any extra customer features

Separating out different columns into various categories as defined above

target_var = ['Exited']

cols_to_remove = ['RowNumber', 'CustomerId']

num_feats = ['CreditScore', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'EstimatedSalary']

cat_feats = ['Surname', 'Geography', 'Gender', 'HasCrCard', 'IsActiveMember']

Separating out target variable and removing the non-essential columns

y = df[target_var].values

df.drop(cols_to_remove, axis=1, inplace=True)

Separating out train-test-valid sets:

Since this is the only data available to us, we keep aside a holdout/test set to evaluate our model at the very end in order to estimate our chosen model's performance on unseen data / new data.

A validation set is also created which we'll use in our baseline models to evaluate and tune our models

from sklearn.model_selection import train_test_split

Keeping aside a test/holdout set

df_train_val, df_test, y_train_val, y_test = train_test_split(df, y.ravel(), test_size = 0.1, random_state = 42)

Splitting into train and validation set

df_train, df_val, y_train, y_val = train_test_split(df_train_val, y_train_val, test_size = 0.12, random_state = 42

df_train.shape, df_val.shape, df_test.shape, y_train.shape, y_val.shape, y_test.shape

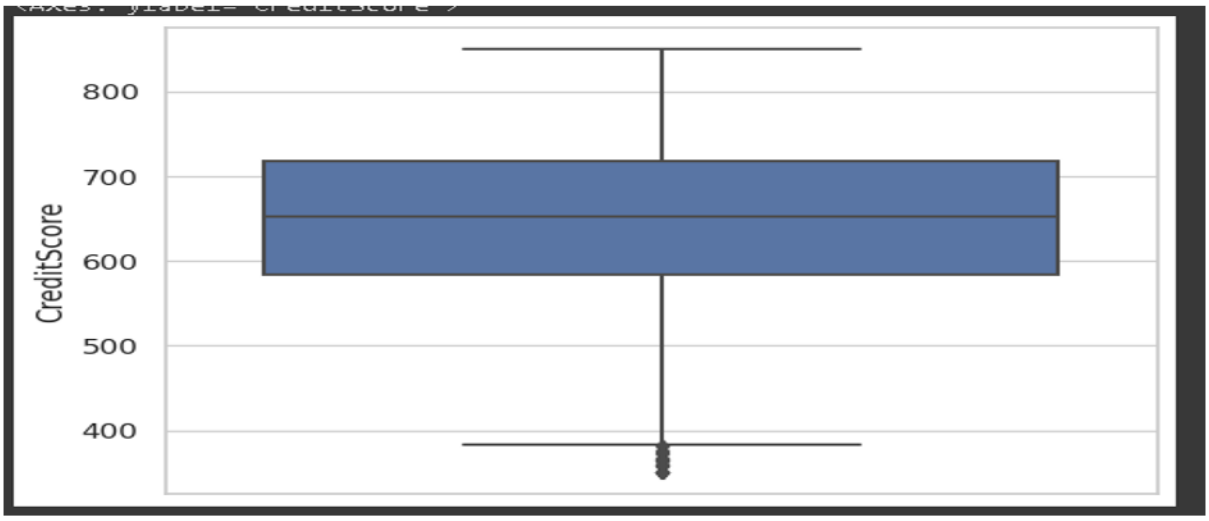
np.mean(y_train), np.mean(y_val), np.mean(y_test)

Univariate plots of numerical variables in training set:

CreditScore

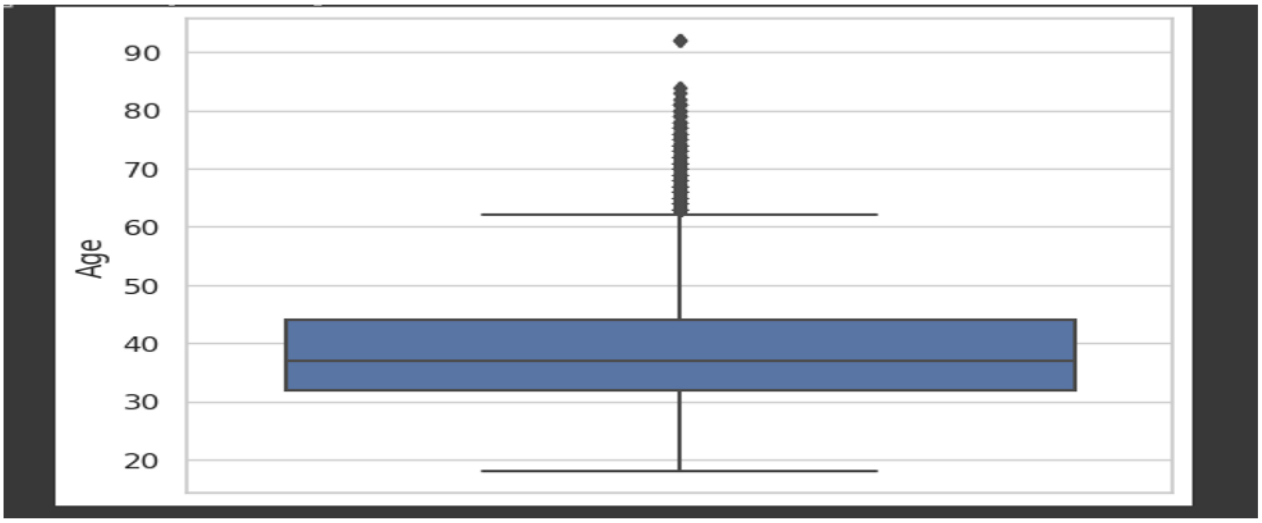
sns.set(style="whitegrid")

sns.boxplot(y = df_train['CreditScore'])



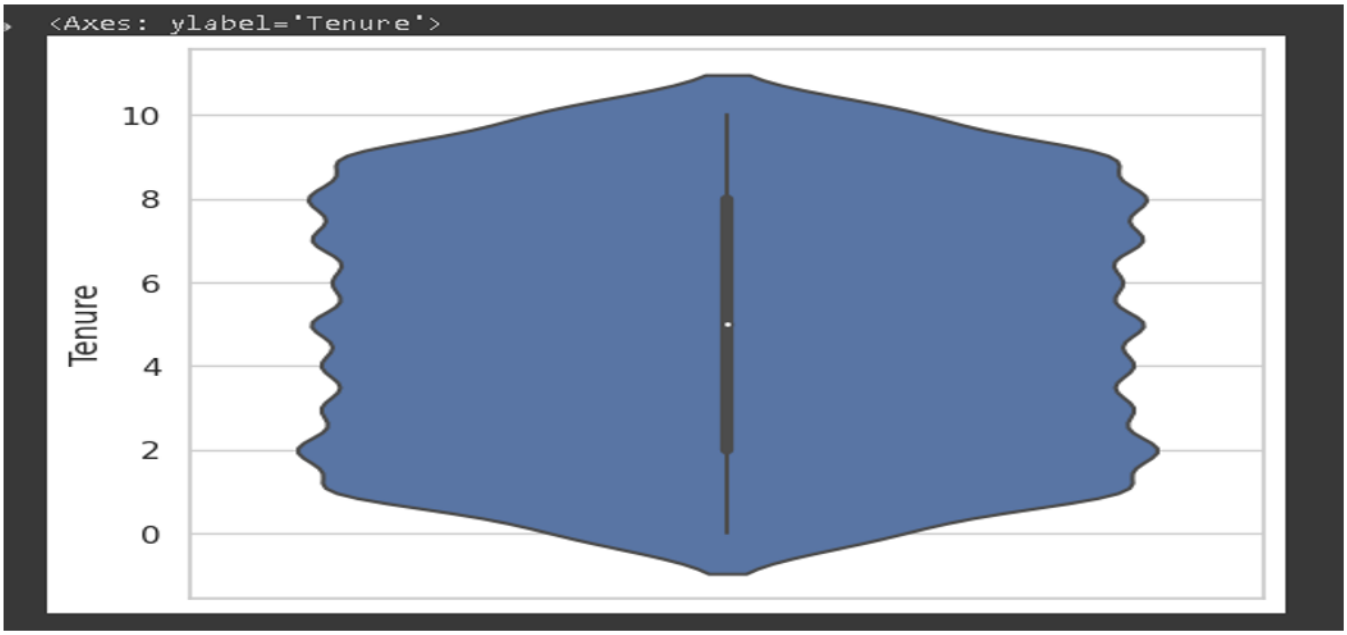
Age

sns.boxplot(y = df_train['Age'])



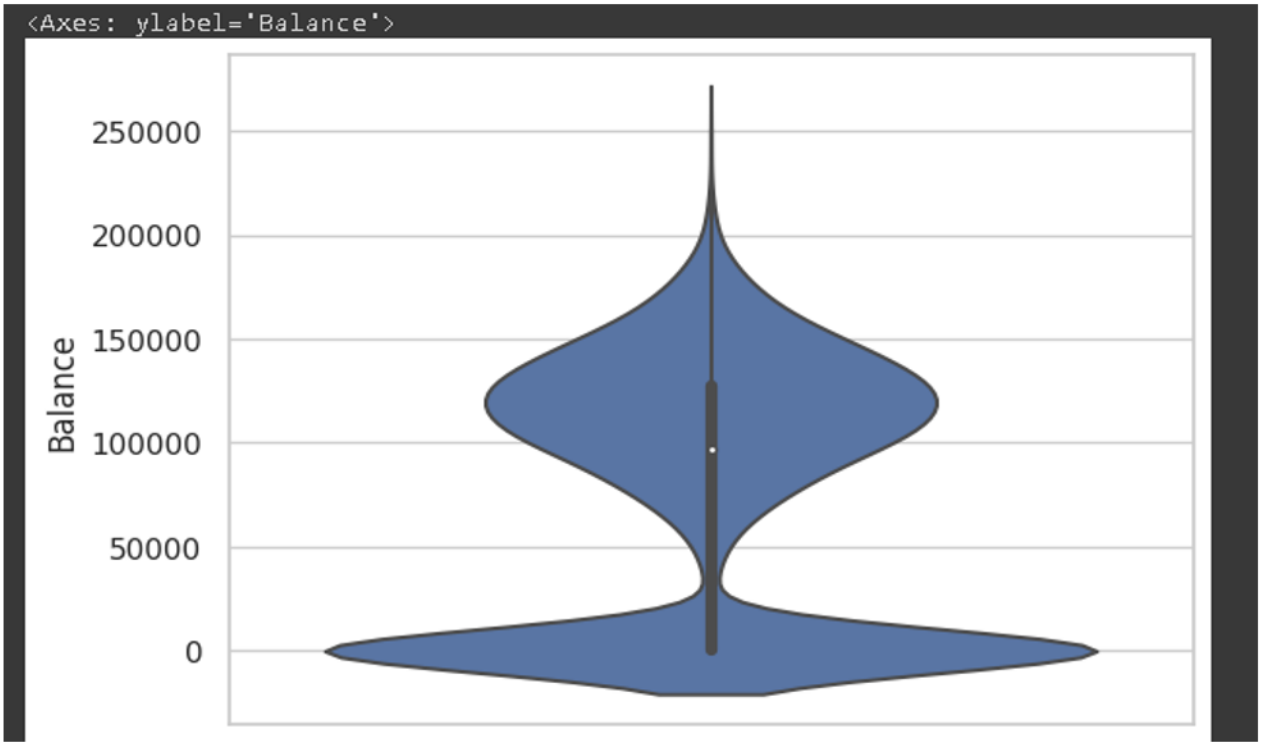
Tenure

sns.violinplot(y = df_train.Tenure)



Balance

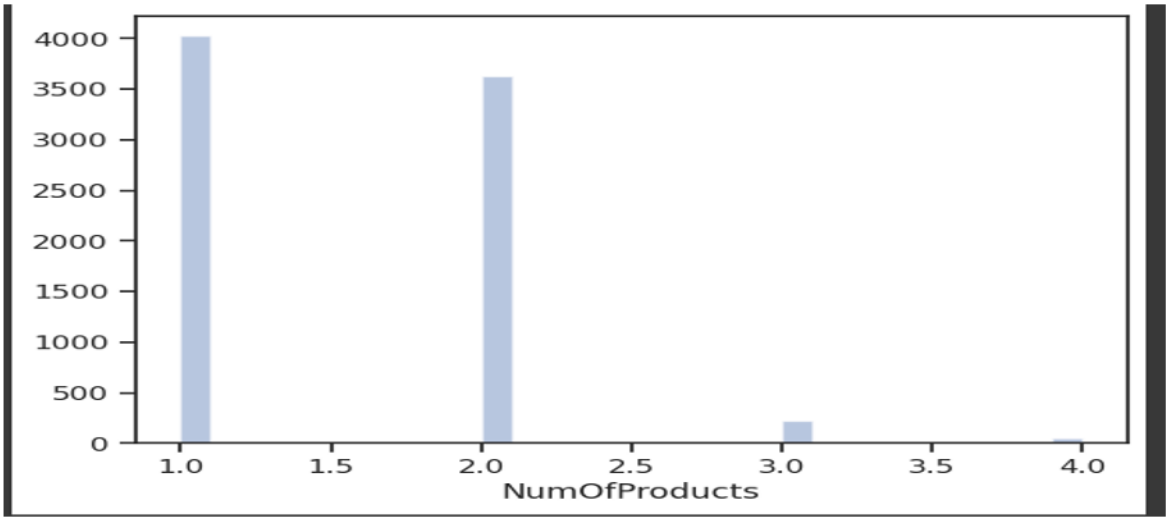
sns.violinplot(y = df_train['Balance'])



NumOfProducts

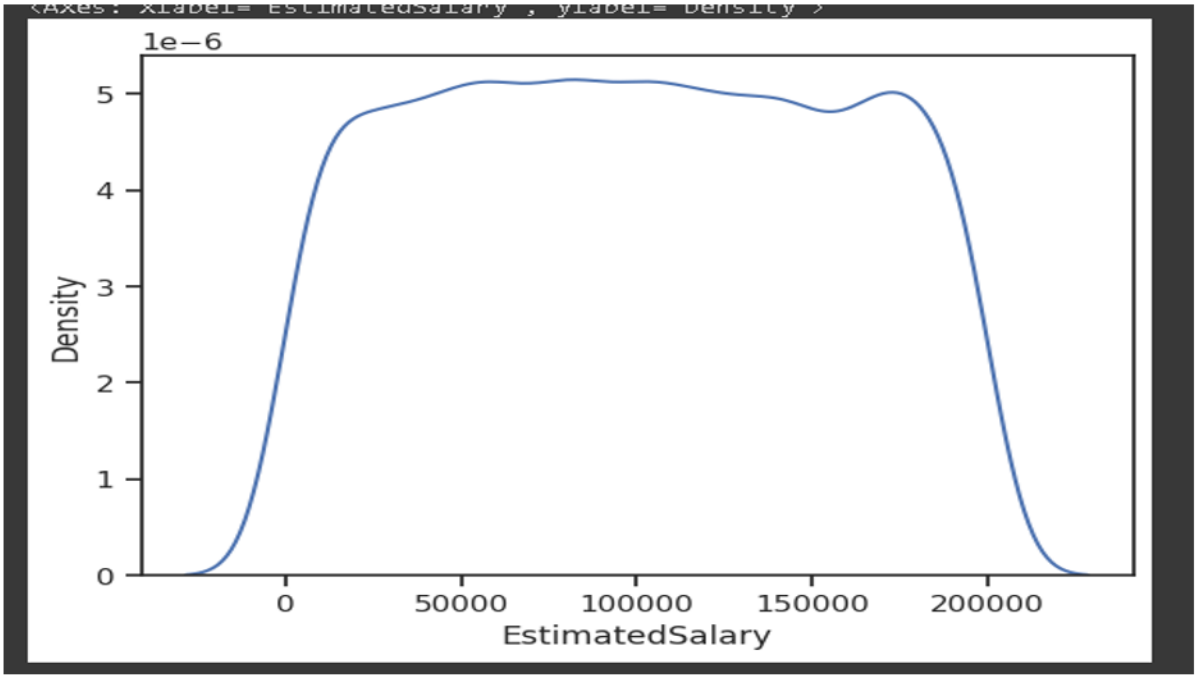
```
sns.set(style = 'ticks')

sns.distplot(df_train.NumOfProducts, hist=True, kde=False)
```



EstimatedSalary

```
sns.kdeplot(df_train.EstimatedSalary)
```



Missing values and outlier treatment:

No missing values!

```
df_train.isnull().sum()

## Making all changes in a temporary dataframe

df_missing = df_train.copy()

## Modify few records to add missing values/outliers

# Introducing 10% nulls in Age

na_idx = df_missing.sample(frac = 0.1).index

df_missing.loc[na_idx, 'Age'] = np.NaN

# Introducing 30% nulls in Geography

na_idx = df_missing.sample(frac = 0.3).index

df_missing.loc[na_idx, 'Geography'] = np.NaN

# Introducing 5% nulls in HasCrCard

na_idx = df_missing.sample(frac = 0.05).index

df_missing.loc[na_idx, 'HasCrCard'] = np.NaN

df_missing.isnull().sum()/df_missing.shape[0]
```

Calculating mean statistics

```
age_mean = df_missing.Age.mean()

age_mean

# Filling nulls in Age by mean value (numeric column)

#df_missing.Age.fillna(age_mean, inplace=True)

df_missing['Age'] = df_missing.Age.apply(lambda x: int(np.random.normal(age_mean,3)) if np.isnan(x) else x)


## Distribution of "Age" feature before data imputation

sns.distplot(df_train.Age)
```

Distribution of "Age" feature after data imputation

sns.distplot(df_missing.Age)

Filling nulls in Geography (categorical feature with a high %age of missing values)

geog_fill_value = 'UNK'

df_missing.Geography.fillna(geog_fill_value, inplace=True)

Filling nulls in HasCrCard (boolean feature) - 0 for few nulls, -1 for lots of nulls

df_missing.HasCrCard.fillna(0, inplace=True)

Categorical variable encoding:

As a rule of thumb, we can consider using :

Label Encoding ---> Binary categorical variables and Ordinal variables

One-Hot Encoding ---> Non-ordinal categorical variables with low to mid cardinality (< 5-10 levels)

Target encoding ---> Categorical variables with > 10 levels

HasCrCard and IsActiveMember are already label encoded

For Gender, a simple Label encoding should be fine.

For Geography, since there are 3 levels, OneHotEncoding should do the trick

For Surname, we'll try Target/Frequency Encoding

Evaluating the model - Metrics:

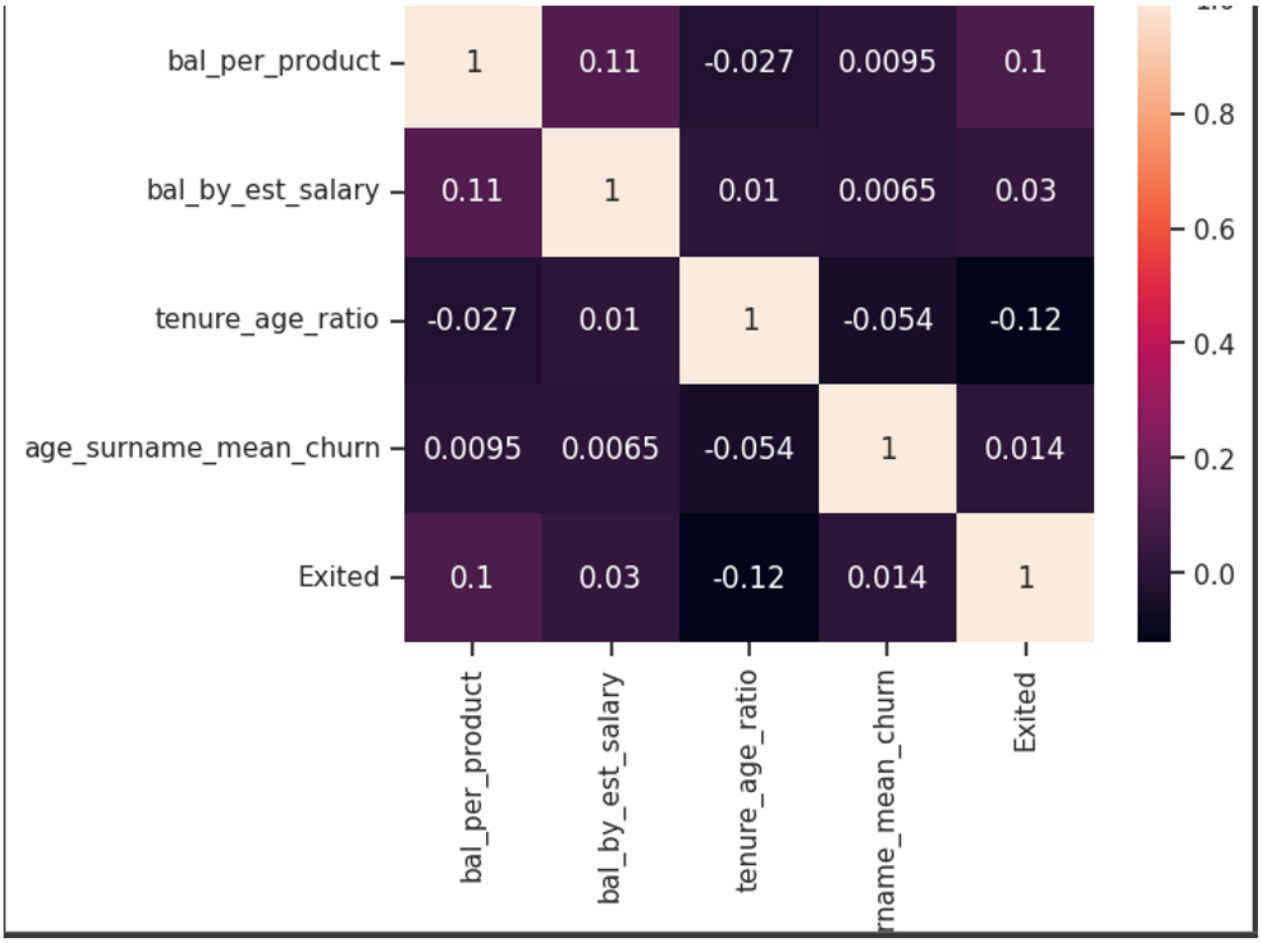
Training metrics

roc_auc_score(y_train, clf.predict(X_train))

recall_score(y_train, clf.predict(X_train))

confusion_matrix(y_train, clf.predict(X_train))

print(classification_report(y_train, clf.predict(X_train)))



Validation metrics

roc_auc_score(y_val, clf.predict(X_val))

recall_score(y_val, clf.predict(X_val))

confusion_matrix(y_val, clf.predict(X_val))

print(classification_report(y_val, clf.predict(X_val)))

Feature scaling and normalization:

Different methods :

Feature transformations - Using log, log10, sqrt, pow

MinMaxScaler - Brings all feature values between 0 and 1

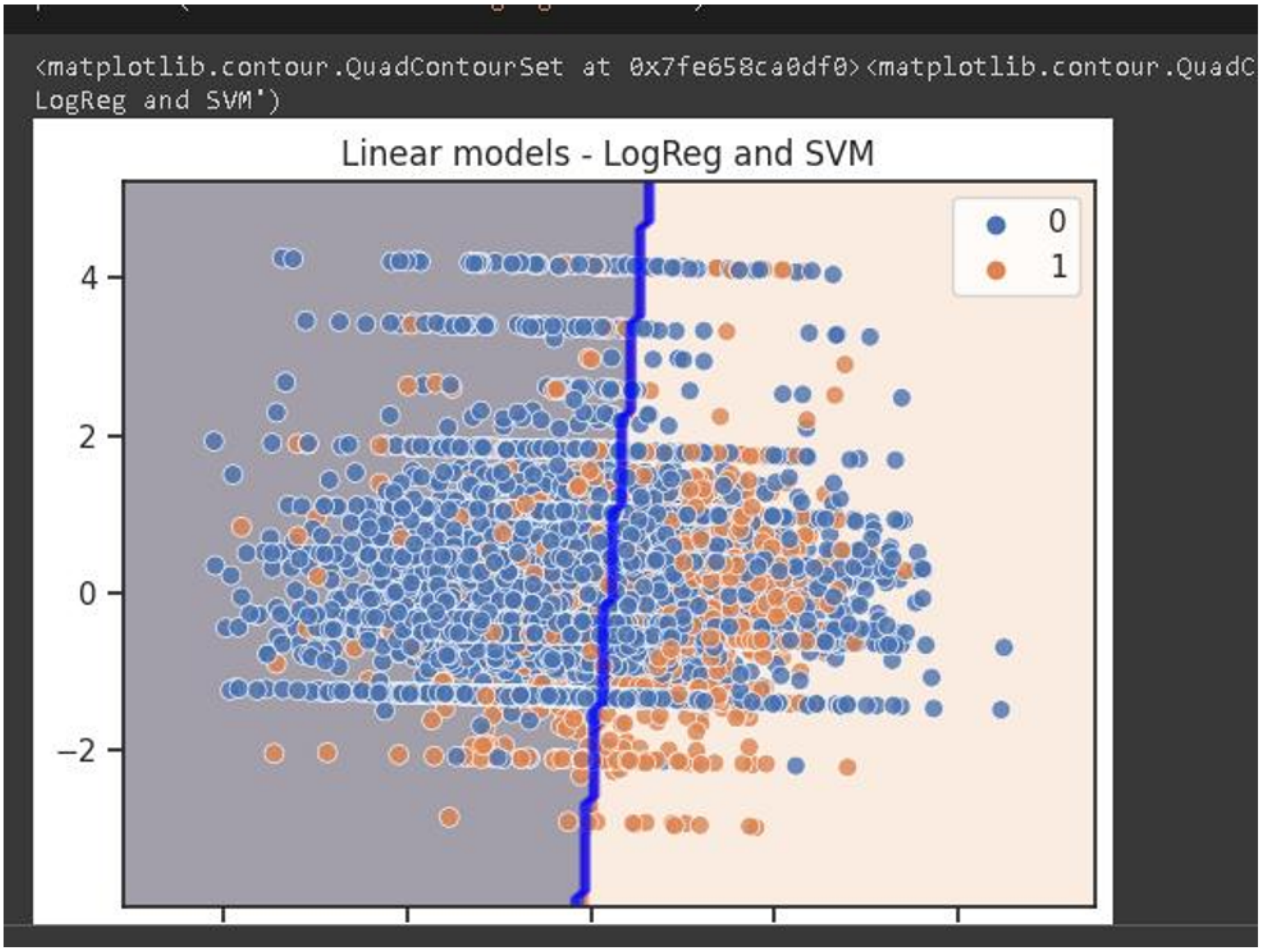
StandardScaler - Mean normalization. Feature values are an estimate of their z-score

Feature transformations:

Demo-ing feature transformations

sns.distplot(df_train.EstimatedSalary, hist=False)

Feature selection - RFE



Features shortlisted through EDA/manual inspection and bivariate analysis :

Age, Gender, Balance, NumOfProducts, IsActiveMember, the 3 country/Geography variables, bal per product, tenure age ratio.Now, let's see whether feature selection/elimination through RFE (Recursive Feature Elimination) gives us the same list of features, other extra features or lesser number of features.

To begin with, we'll feed all features to RFE + LogReg model.

Evaluating the model - Metrics:

Training metrics

roc_auc_score(y_train, clf.predict(X_train))

recall_score(y_train, clf.predict(X_train))

confusion_matrix(y_train, clf.predict(X_train))

print(classification_report(y_train, clf.predict(X_train)))

Validation metrics

roc_auc_score(y_val, clf.predict(X_val))

recall_score(y_val, clf.predict(X_val))

confusion_matrix(y_val, clf.predict(X_val))

print(classification_report(y_val, clf.predict(X_val)))

Hyperparameter tuning:

RandomSearchCV vs GridSearchCV

Random Search is more suitable for large datasets, with a large number of parameter settings

Grid Search results in a more precise hyperparameter tuning, thus resulting in better model performance. Intelligent tuning mechanism can also help reduce the time taken in GridSearch by a large factor

Will optimize on F1 metric. We could easily reach 75% Recall from the default parameters as seen earlier

from sklearn.pipeline import Pipeline

from sklearn.model_selection import GridSearchCV, RandomizedSearchCV

```
Recall metric
Model rf_21: mean = 0.7499816180027477, std_dev = 0.02250666830570937
Model lgb_21: mean = 0.7866856291480427, std_dev = 0.015745566437193475
Model xgb_21: mean = 0.7506085408564075, std_dev = 0.01096611280139578
Model et_21: mean = 0.7307153499351793, std_dev = 0.006637230639703262
Model rf_1001: mean = 0.7518430370929355, std_dev = 0.02662435713135778
Model lgb_1001: mean = 0.6884232116251622, std_dev = 0.014573973874519829
Model xgb_1001: mean = 0.6753719935759757, std_dev = 0.01756702999772903
Model et_1001: mean = 0.7338267448385286, std_dev = 0.0065527730614234865
Model knn_3: mean = 0.32214933921557243, std_dev = 0.021051639994704833
Model knn_5: mean = 0.2879356049612043, std_dev = 0.006396680440459953
Model knn_11: mean = 0.23568622898163735, std_dev = 0.023099705052575383
Model gauss_nb: mean = 0.0360906329211896, std_dev = 0.0151162576177723
Model multi_nb: mean = 0.5404191095373541, std_dev = 0.022285871235774777
Model compl_nb: mean = 0.5404191095373541, std_dev = 0.022285871235774777
Model bern_nb: mean = 0.31030552814380524, std_dev = 0.022201596952259223
F1-score metric
Model rf_21: mean = 0.6301402837422649, std_dev = 0.014786649708942829
Model lgb_21: mean = 0.6445713376921776, std_dev = 0.010347896896123705
Model xgb_21: mean = 0.6130509823329311, std_dev = 0.00848890204896738
Model et_21: mean = 0.5913244194622826, std_dev = 0.010752644551251507
Model rf_1001: mean = 0.6282845853832282, std_dev = 0.01600543760644489
Model lgb_1001: mean = 0.677231392541388, std_dev = 0.009841732603586511
Model xgb_1001: mean = 0.683463280904695, std_dev = 0.014982910608582397
```

```
from lightgbm import LGBMClassifier

## Preparing data and a few common model parameters

# Unscaled features will be used since it's a tree model

X_train = df_train.drop(columns = ['Exited'], axis = 1)

X_val = df_val.drop(columns = ['Exited'], axis = 1)

X_train.shape, y_train.shape

X_val.shape, y_val.shape

lgb = LGBMClassifier(boosting_type = 'dart', min_child_samples = 20, n_jobs = - 1, importance_type =
'gain', num_leaves = 31)

model = Pipeline(steps = [('categorical_encoding', CategoricalEncoder()),

                  ('add_new_features', AddFeatures()),

                  ('classifier', lgb)

                 ]
```

CHAPTER 4 : CONCLUSION

Different Ensemble techniques were used to analyse and determined whether and how much the accurate the customer prediction churn was being performed and in future we can also use this data to analyse loan based prediction eligibility for future customer.

FUTURE SCOPE:

This will be used to link with analysis of Debit Card Fraud Analysis, Fraud Online Transaction Analysis. It also has application in field of Risk Management, Market Analysis

CHAPTER 5 : REFERENCES

1. https://www.qualtrics.com/experience-management/customer/customer-churn/
2. https://www.google.com
3. https://arxiv.org/abs/2206.01523
4. https://www.kdnuggets.com/
5. https://www.kaggle.com/