PREDICTING CUSTOMER CHURN IN BANKS

A PROJECT REPORT

for

SOFT COMPUTING (SWE1011)

in

M.Tech (Software Engineering)

By

FALL SEMESTER 2022-23

PREDICTING CUSTOMER CHURN IN BANKS

SHYAM SUNDAR S (1), KARTHIKEYAN S (2)

Department of Information Technology, VIT University, Vellore, Tamil Nadu, India

Abstract

In this era, ANN can understand human activities and their meanings. We can utilize this ability of ANN in various fields or applications. One specific field of interest is a prediction of churning customers in any industry. Prediction of churning customers is the state of art approach which predicts which customer is near to leave the services of the specific bank. We can use this approach in any big organization that is very conscious about their customers. However, this study aims to develop a model that offers a meaningful churn prediction for the banking industry. For this purpose, we develop a customer churn prediction approach with the three intelligent models' random forest (RF), AdaBoost. Furthermore, the experimental results show that RF yielded good results for the full feature-selected datasets

Keywords –Artificial Neural Network, PREDICTING CUSTOMER CHURN IN BANKS

I. INTRODUCTION

Customer churn has become a big issue in many banks because it costs a lot more to acquire a new customer than retaining existing ones. With the use of a customer churn prediction model possible churners in a bank can be identified, and as a result the bank can take some action to prevent them from leaving. Churn in the banking sector is a major problem today. Losing the customers can be very expensive as it costs to acquire a new customer. In this project, we have made a solution for the churn problem in banking sector using data mining technique. As predicting churn is more important for a bank, we have used Classification to yield a better overall classification rate. As we know, it is much more expensive to sign in a new client than keeping an existing one. It is advantageous for banks to know what leads a client towards the decision to leave the company. Churn Prediction allows companies to develop loyalty programs and retention campaigns to keep as many customers as possible. An artificial neural network (ANN) is the piece of a computing system designed to simulate the way the human brain analyzes and processes information. It is the

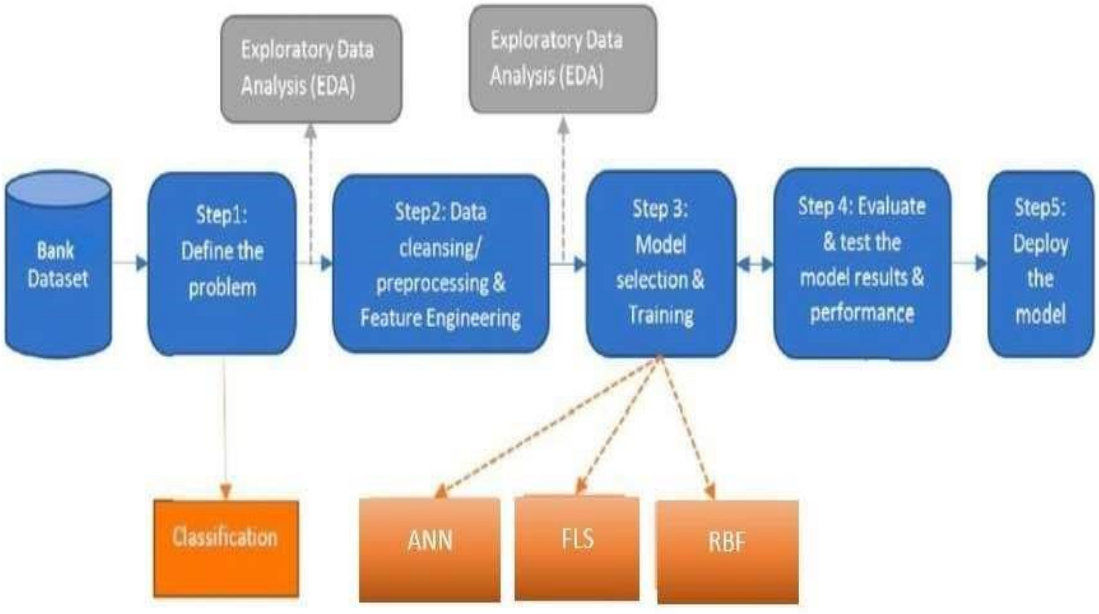
foundation of artificial intelligence (AI) and solves problems that would prove impossible or difficult by human or statistical standards.

II. BACKGROUND

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. Its uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

- Here we downloaded more than 1500 Python/R data science packages,
- Managed libraries, dependencies, and environments with conda
- Build and train ML and deep learning models with scikit-learn, TensorFlow and Theano
- Use Dask, NumPy, Pandas and Numba to analyze data scalably and fast
- Perform visualization with Matplotlib, Bokeh, Datashader, and Holoviews

General architecture:



Review on various schemes

IMAGE PRE-PROCESSING:

This section pre-processed the data before introducing it to our proposed model. In the first instance, we modified the values of our class variable (Attrition Flag). This column contains two values. The "Attrition Customer" value is changed from "1" to "0" while the "Existing Customer" value remains unchanged. The gender column is then modified. Female is replaced with 1, and male is replaced with 0. Finally, there are some Unknown values in Education Level, Income Category, and Marital Status. These values have been eliminated from our dataset.

ROI from NON-ROI:

Customer churn prediction ROI means determining which customers can opt-out of a product or subscription to a service, depending on how they use it. This is an important forecast for many businesses because attracting new customers is much more expensive than retaining existing ones. Therefore, using advanced artificial intelligence techniques such as machine learning (ML), you will be able to predict potential outflows that are going to abandon your services.

Churn prediction software and solutions are used in many industries, such as ecommerce, mobile gaming, telecom, fin-tech (finance), healthcare, insurtech (insurance), fitness, retail, banking and many more businesses. While calculating investment on customer churn prediction costs you should understand all other financial and business benefit aspects. In addition, statistics show that it is 5 times cheaper to retain existing customers rather than finding and acquiring a new ones. Churn prediction allows you to not only keep your clients active and loyal but also give you additional opportunity for up-sell and cross-sell on retained customers. So, all of these factors give you an amazing opportunity to grow your business faster.

Literature survey:

Authors	Methodology or Techniques used	Advantages	Issues	Dataset used	Metrics used
Kamorudeen A. AMUDA, Adesesan B. ADEYEMO2 2019	Customers Churn Prediction in Financial Institution Using Artificial Neural Network.	In this study a predictive model using Multi-layer Perceptron of Artificial Neural Network architecture was developed to predict customer churn in financial institution. Previous researches have used supervised machine learning classifiers such as Logistic Regression, Decision Tree, Support Vector Machine, K-Nearest Neighbors, and Random Forest.	The information obtained from the predictive model can be used for decision making in customer retention management system. For future work other ANN architectures such as (CNN), RNN) and Long Short-Term Memory (LSTM).	The dataset was extracted from the bank's customer relationship management database and transaction warehouse from a major Nigeria bank.	This research was designed for the prediction of customers churn using the data from a financial institution in Nigeria. The data was extracted from the bank database and divided into three sets: training set, test set, and validation set.
Ibrahim M.M.Mitkees, Asist. Prof. Sherif M Badr, Dr. Ahmed Ibrahim Bahgat ElSeddawy 2017	Customer Churn Prediction Model using Data Mining techniques	The results are compared to find an appropriate model with higher precision and predictability. As a result, the use of the Random Forest model after oversampling is better compared to other models in terms of accuracy.	The Churn Factor is used in many functions to describe the various areas or scenarios when the churn rate is high. The study proposes that there is a huge deviation in the graph of churners when customer service calls are considered.	using a real-world churn dataset, they compared different cost insensitive and cost-sensitive classification algorithms and measured their usefulness, based on their predictive power and the cost optimization	As for the clustering technique, it has proved that Dbscan clustering is suitable and effective in clustering the data set into two categories together with giving high percentage of non-churners over churners at the cluster-0. As regards the association rule angle Aperiori and FP-Growth

Authors	Methodology or Techniques used	Advantages	Issues	Dataset used	Metrics used
Amgad Muneer, Rao Faizan Ali, Amal Alghamdi, Shakirah Mohd Taib, Ahmed Almaghthawi, Ebrahim Abdulwasea Abdullah Ghaleb 2022	Predicting customers churning in banking industry: A machine learning approach	In this era, machines can understand human activities and their meanings. We can utilize this ability of machines in various fields or applications. One specific field of interest is a prediction of churning customers in any industry.	Churning can be measured in terms of the number of customers lost, the ratio of customers lost, or the percentage of customers lost compared to the total number of customers in the bank. This churning can be measured quarterly or annually. An accurate forecast provides insight into the future, which allows for developing a strategy.	The dataset used for the prediction process task is publicly available on the Kaggle website	The proposed study conducted the most comprehensive investigation of the credit card churn prediction problem in banks using machine learning techniques. We proposed a customer churn prediction system with Random Forest, AdaBoost, and SVM intelligent models.
Teemu Mutanen, Jussi Ahola, and Sami Nousiainen	Customer churn prediction - a case study in retail banking	This paper will present a customer churn analysis in consumer retail banking sector. The focus on customer churn is to determinate the customers who are at risk of leaving and if possible on the analysis whether those customers are worth retaining.	The customer churn analysis in this study might not be interesting if the customers are valued based on the customer lifetime value. The churn definition in this study was based on the current account. But if the churn definition was based on for example loyalty program account or active use of the internet service	The dataset used for the prediction process task is publicly available on the Kaggle website	The findings of this study indicate that, in case of logistic regression model, the user should update the model to be able to produce predictions with high accuracy since the independent variables of the models varies.

Authors	Methodology or Techniques used	Advantages	Issues	Dataset used	Metrics used
Xiaohua Hu	"A Data Mining Approach for Retailing Bank Customer Attrition Analysis"	The advantage of this approach is that the rules are easy to understand, and they are frequently useful for discovering underlying business processes.	The accuracy of the learning stage of the ANN approach, not detailed in this paper.	Dataset from Kaggle Churn Modelling Classification dataset.	This research Work demonstrates The effectiveness and efficiency of data mining in attrition analysis for retailing bank
Abbas Keramati, Hajar Ghaneei & Seyed Mohammad Mirmohammadi	"Decision tree Technique for Churn prediction"	Being based on existing information technologies which allow one to collect data from organizations' databases, data mining introduces a powerful tool for the extraction of knowledge from huge amounts of data.	In addition, due to the large volume of data stored in the database and the associated privacy issues, it was time-consuming to extract all the data. Future research will further investigate the implementation results	Transaction data through electronic banking portals, the length of the customer association, and customer complaints were extracted from the bank's database.The DT method was applied for the modeling of this dataset.	Data mining by evolutionary learning (DMEL) could show the reason or probability of a churning phenomenon; DT, however, could only show the reason.
Weiyun Ying; Xiu Li; Yaya Xie; Ellis Johnson	"Preventing customer churn by using random forests modeling"	The random forests method, introduced by Breiman, adds an additional layer of randomness to bootstrap aggregating ("bagging") and is found to perform very well compared with many other classifiers. It is robust against overfitting and very userfriendly.	Volume is the major issue in this paper. Future research will further investigate the implementation results	Dataset were extracted from the bank's database	In this paper, Continuing research aimed at improving the effectiveness and generalization ability. IBRF employs internal variables to determine the distribution of samples.

Authors	Methodology or Techniques used	Advantages	Issues	Dataset used	Metrics used
Jing Zhao; Xing-Hua Dang	"Bank Customer Churn Prediction Based on Support Vector Machine: Taking a Commercial Bank's VIP Customer Churn as the Example"	The method was compared with artificial neural network, decision tree, logistic regression and naïve Bayesian classifier regarding customer churn prediction for commercial bank's VIP customers. It is found that the method has the best accuracy rate, hit rate, covering rate and lift coefficient, and provides an effective measurement for bank's customer churn prediction.	Predicting the tendency of customer churn according to SVM will provide less guide for the customer marketing of the bank.In future the more guidance should engage.	Utilized VIP customers' dataset provided by a domestic branch of China Construction Bank (CCB).	In this paper, they applied SVM to the prediction of bank Customer churn. By comparing with (ANN), decision tree (C4.5), logistic regression and naive Bayesian classifier as a benchmark. The results demonstrated that, from methodological perspective, SVM had the characteristics of simple classification surface, high generalizationperformance and high fitting accuracy, etc.
Indranil Bose &Xi Chen	"Hybrid Models Using Unsupervised clustering for Prediction of Customer Churn"	Algorithms are applied on the calibration data. The result included two cluster labels. One indicated the identity of the segment obtained using information on services usage and the other indicated the identity of the segment obtained using information on revenue contribution.	The attributes for customers of mobile services were grouped under services usage and revenue contribution because this categorize seemed intuitive. Different types of grouping of attributes and use of other attributes related to customers can be investigated in future.	SOM helped generate the best hybrid model for current data, BIRCH helped generate the best hybrid model for future data, and KM helped generate the most number of models that beat the benchmark model for the two data sets	In this paper, Clustering is used as the first stage in the hybrid method and the second stage is conducted using decision trees.
Author links open overlay panelParag C.Pendharkar	Hybrid Neural Network, ANN, Data mining Techniques	The (ANN+ANN) technique of the two hybrid models performs the data reduction task by filtering out unrepresentative training data. Then, the outputs as representative data are used to create the prediction model based on the (SOM+ANN) technique.	Population is growing in exponential way. The only solution to control this is to predict theheart disease and medicate it before it gone worse.Our hybrid approach gives higher accuracy rate of 97% of diseasedetection than earlier proposed method.	Dataset from Kaggle	To account for uneven class distribution in training and test datasets, two additional performance metrics called sensitivity and specificity are used in the literature

Authors	Methodology or Techniques used	Advantages	Issues	Dataset used	Metrics used
ChihFongTsai Yu-HsinLu (2018)	"Hybrid Neural Network, ANN, Data mining Techniques"	The (ANN+ANN) technique of the two hybrid models performs the data reduction task by filtering out unrepresentative training data. Then, the outputs as representative data are used to create the prediction model based on the (SOM+ANN) technique.	For future work, several issues can be considered. First, as the pre-processing stage in data mining is a very important step for the final prediction performances.	we consider a CRM dataset 1 provided by American telecom companies.	In this paper, three different kinds of testing sets are considered. They are the general testing set and two fuzzy testing sets based on the filtered-out data by the first technique of the two hybrid models, i.e. ANN and SOM, respectively.
Tiwari, A., Hadden, J & Turner, C. (2020)	"New Neural Network Based Customer Profiling Methodology"	The methodology for customer churn prediction describes a predictive approach for the identification of customers who are most likely to churn in the future.	Future research could include the further study of customer behaviour, web based communication systems and even stock markets.	Dataset from Kaggle Churn Modelling Classification dataset.	The accuracy of the learning stage of the NN approach, detailed in this paper, could be enhanced by its combination with an evolutionary optimization technique such as genetic algorithms.
ChiunSinLin GwoHshiung Tzeng YangChiehCh in (2018)	"Combined rough set theory and flow network graph to predict customer churn in credit card accounts"	In RST, discretization can convert continuous attributes into discrete attributes while removing redundant and irrelative attributes	Since limited studies exist on credit card customer churn, numerous possible research avenues remain. First, the present study considers only 1-month period data, so future studies could use longer period data for more accurate results.Compare this combined model with other approaches or modify the model to be even more effective in predicting customer churn.	Studies have not yet adequately introduced rules based on customer characteristics and churn forms of original data.	In this paper, RST is used to discover hidden information in data and to exploring the rules and characteristics of customer churn. The decision rules can be transferred into a flow network graph to represent the connections of pathways and the degrees of their interdependency.

Authors	Methodology or Techniques used	Advantages	Issues	Dataset used	Metrics used
Manjit Kaur, Dr. Kawaljeet Singh, Dr. Neeraj Sharma (2016)	"Customer Churn predictive using data mining"	The customer churn is a common measure of lost customers. By minimizing customer churn a company can maximize its profits.	The modelling of collective systems could be possible with such an enhanced system outlined here. Future research could include the further study of customer behavior, web based communication systems and even markets.	The total prediction accuracy has been calculated by adding the correct number of classified non-churn with the correct number of classified churn, dividing the number by the total dataset size of 8409 and multiplying by 100.	In this paper, In the model construction phase, we build a classification/prediction model that predicts the potential behavior of customers in the near future.
Özden Gür AliaUmutArıt Ürk (2014)	"Dynamic churn prediction framework with more effective use of rare event data"	Improves accuracy significantly even vs. balanced data, across prediction horizons. Independently trained binary classifiers approach outperforms survival analysis.	Future research can explore the improvement due to MPTD as a 1142 function of the relative and absolute rarity in the data, and compare 1143 It with other methods of addressing the rare event problem. This will 1144 help establish guidelines for the circumstances under which MPTD 1145 offers a significant improvement in accuracy.	We used the 12 months from April 2009 to 710 March 2010 for constructing the training datasets.	The proposed approach to dynamic churn prediction involves a set of independently trained horizon specific binary classifiers that use the proposed dataset generation framework.
Abbas Keramati, Hajar Ghaneei & Seyed Mohammad Mirmohamma di (2017)	"Decision tree Technique for churn prediction"	Being based on existing information technologies which allow one to collect data from organizations' databases, data mining introduces a powerful tool for the extraction of knowledge from huge amounts of data.	Future research could include the further study of customer behaviour, web based communication systems and even stock markets.	Churn Modelling Classification dataset from Kaggle.	Data mining by evolutionary learning (DMEL) could show the reason or probability of a churning phenomenon; DT, however, could only show the reason.

Authors	Methodology or Techniques used	Advantages	Issues	Dataset used	Metrics used
Mohammad Ridwan Ismail, Mohd Khalid Awang, M Nordin A Rahman and Mokhairi Makhtar	"Multi-Layer Perceptron Approach"	The model consists of the Interconnection of neurons via the respective weight for each connection. The neural network produced an output based on experience during the training process	The results are compared against the most popular churn prediction techniques such as Multiple Regression Analysis and Logistic Regression Analysis.	Customer Churn datasets.	Regression analysis is known as a popular statistical tool for the prediction of customers. In this paper, the analysis provides the relationship between the independent and dependent variables which apply the input features and the result of churn or non-churn in this application.
Farid Shirazia Mahbob Mohammadi	"A big data analytics model for customer Churn prediction in the retiree segment"	Sustaining a competitive advantage and maintaining the Point of Differentiation (POD), in order to remain in clients' financial paths, is considered one of the highest priorities in strategic planning related to client attraction and more importantly, retention within the retail banking sector.	The online usage within the retiree segment did not show any correlation with the rate of attrition, due to low usage of online banking and information seeking channels by these particular clients.	Dataset were extracted from the bank's database.	The main purpose of the conducted research supports the claim that offering the right product at the right moment can minimize churning. This study is also attempting to verify some sub-claims such as the number of visits to external websites is an indication of churn decisions and the longer a client has a history with a bank, lower the likelihood of churn.
Hemlata Dalmia, Ch V S S Nikil, Sandeep Kumar	"Customer churn Prediction using Supervised Learning, Algebraic Fault Analyzer"	Machine solver is a block which has an input equation and process to solve it by using the mathematical formula automatically by software, and output is applied as an input into the sat solver.	Results in less sensitive rule-sets, but allows to include domain knowledge, and results in comprehensible rule-sets which are much smaller than the rule-sets induced with C4.5.	This paper presents the application of AntMiner+ and ALBA on a publicly available churn prediction dataset.	After surveying we found that, the encryption standard is effective encryption system, but it cannot run in the low resource system below block size of 128 bit. The LBC provide security that works in the low resource system like wearable devices, RFID, sensor nodes, etc.

Authors	Methodology or Techniques used	Advantages	Issues	Dataset used	Metrics used
ArnoDe Caigny KristofCous sement Koen W.De Bock StefanLessma nn 2020	"Customer churn Prediction using Convolutional neural network"	Embedding these terms in a lower dimensional space offers 2 advantages for deep ANNs. First, ANNs often have computational difficulties with very high dimension, sparse vectors, which are resolved by using term embeddings & Second, this process improves the model's generalization power.	This does not indicate what type of information the textual data actually provide. This information could be valuable for detecting specific churn drivers, which could help managers to set up better-targeted retention campaigns and reduce churn rates in the long run. Therefore, further research on the interpretability of deep ANNs will be useful.	The empirical results demonstrate that textual information is an important source of data for CCP models.	This survey says that the TDL is a valuable performance metric from a managerial perspective because it focuses on those customers who are most at risk of leaving the company and therefore are interesting customers to consider for a retention campaign.
Wouter Verbeke, David Martens, Christophe Mues, Bart Baesens 2019	"Building comprehensible customer churn prediction models with advanced rule induction techniques"	Customer churn prediction models aim to detect customers with a high propensity to attrite. Predictive accuracy, comprehensibility, and justifiability are three key aspects of a churn prediction model.	AntMiner+ results in less sensitive rule-sets, but allows to include domain knowledge, and results in comprehensible rule-sets which are much smaller than the rule-sets induced with C4.5.	This paper presents the application of AntMiner+ and ALBA on a publicly available churn prediction dataset.	This paper provides an extended overview of the literature on the use of data mining in customer churn prediction modeling. It is shown that only limited attention has been paid to the comprehensibility and the intuitiveness of churn prediction models. Therefore, two novel data mining techniques such as C4.5 and RIPPER.

PROPOSED ALGORITHM

KNN

K nearest neighbor (KNN): It is also known as a lazy learning algorithm. KNN is used for both regression and classification predictive problems. The optimal k value is used for better results that can be found using the elbow method.

Random forest

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

Logistic regression

Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of certain classes based on some dependent variables. In short, the logistic regression model computes a sum of the input features (in most cases, there is a bias term), and calculates the logistic of the result.

Gradient boosting

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.

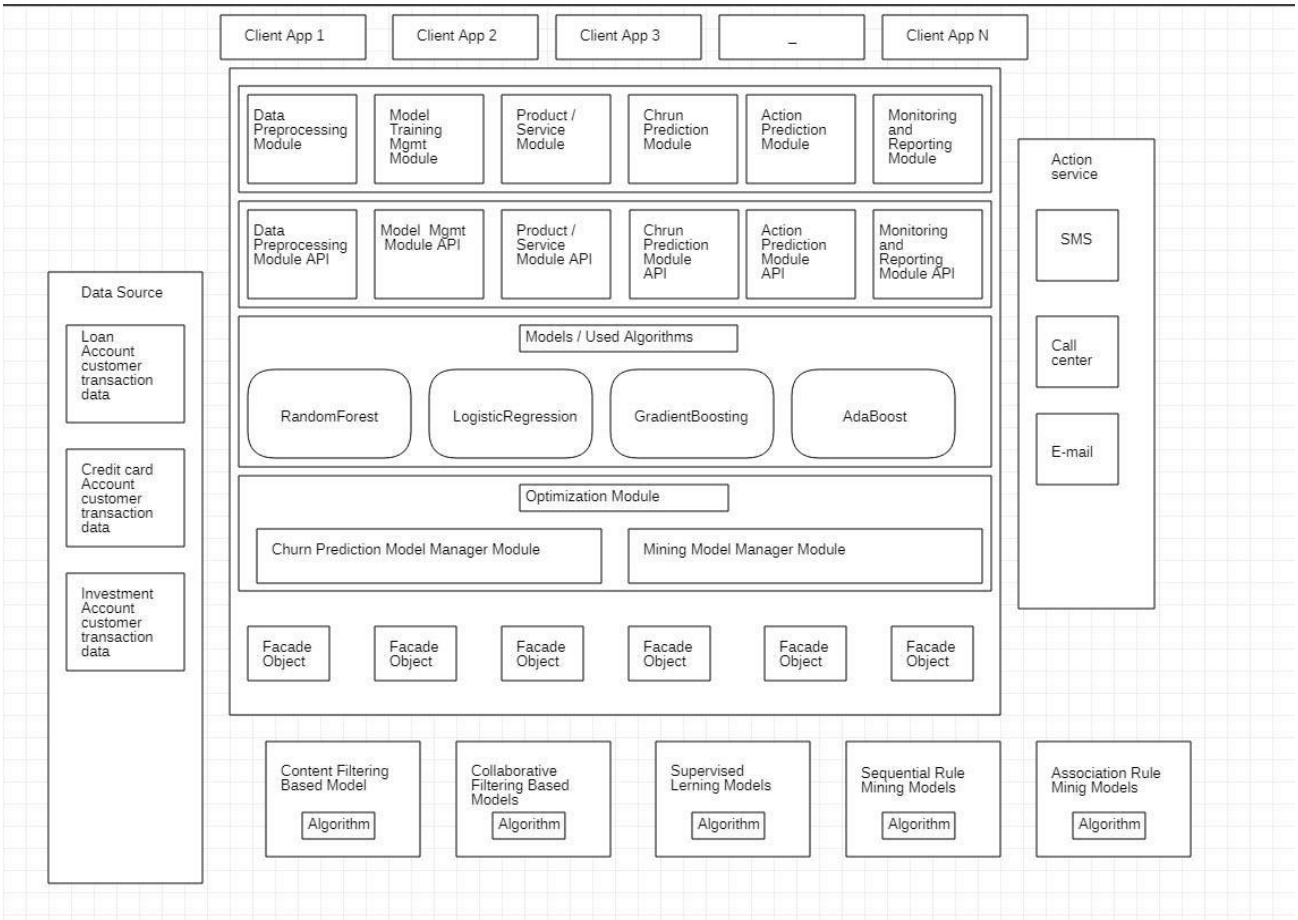
AdaBoost Algorithm

AdaBoost Algorithm is also known as Adaptive Boosting is an Ensemble modelling technique used in Machine Learning to find the best model.

SVC algorithm

SVC is a nonparametric clustering algorithm that does not make any assumption on the number or shape of the clusters in the data. In our experience it works best for low-dimensional data, so if your data is high-dimensional,a preprocessing step, e.g., using principal component analysis, is usually required

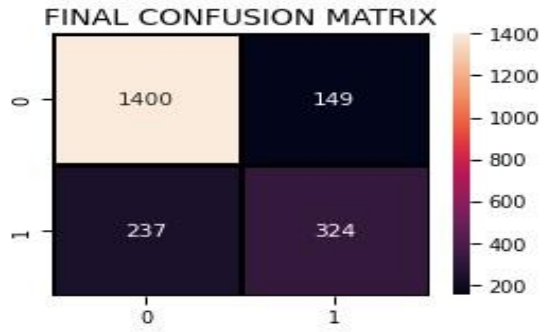
Proposed Architecture



III. EXPERIMENTS RESULTS

Final Result:

```
In [72]: plt.figure(figsize=(4,3))
         sns.heatmap(confusion_matrix(y_test, predictions),
                     annot=True,fmt = "d",linecolor="k",linewidths=3)

         plt.title("FINAL CONFUSION MATRIX",fontsize=14)
         plt.show()
```




```
In [73]: print("Thank You...")
         Thank You...
```

IV. COMPARATIVE STUDY

Initially, while using Artificial neural network where we used multiple classifiers for the accuracy and also, we have done confusion matrix for all the classifier

	Training accuracy
Basic and Previous method	72.20%
Our method	82.30%

V. CONCLUSION AND FUTURE WORK

From the confusion matrix we can see that:

• There are total 1400+149=1549 actual non-churn values and the algorithm predicts 1400 of them as non-churn and 149 of them as churn. While there are 237+324=561 actual churn values and the algorithm predicts 237 of them as non-churn values and 324 of them as churn values.

Customer churn is definitely bad to a firm ’s profitability. Various strategies can be implemented to eliminate customer churn. The best way

to avoid customer churn is for a company to truly know its customers. This includes identifying customers who are at risk of churning and working to improve their satisfaction. Improving customer service is, of course, at the top of the priority for tackling this issue. Building customer loyalty through relevant experiences and specialized service is another strategy to reduce customer churn. Some firms survey customers who have already churned to understand their reasons for leaving in order to adopt a proactive approach to avoiding future customer churn.

VI. REFERENCES

https://www.researchgate.net/publication/299336397_Predicting_Customer_Churn_in_Banking_Industry_using_Neural_Networks

http://dx.doi.org/10.1016/j.eswa.2005.09.080

https://www.google.com/search?q=introduction+for+PREDICTING+CUSTOMER+CHURN+IN+BANKS&client=ms-android-oppo-rev1&sxsrf=ALiCzsbZjEpeoGzP0tIu-zLFx0S03YiNg%3A1663165421140&ei=7eMhY9KGCPSm8QPfg6SIDg&oq=introduction+for+PREDICTING+CUSTOMER+CHURN+IN+BANKS&gs_lcp=ChNtb2JpbGUtZ3dzLXdpei1zZXJwEAMyBQgAEKIEMgUIABCiBDoHCAAQRxCwAzoHCCMQ6gIQJzoHCC4Q6gIQJzoGCAAQHhAWOgUIABCGAzoFCCEQoAE6BAghEApKBAhBGABQtBBYy0ZgrUloAnACeACAAfEBiAH5E5IBBjAuMTEuM5gBAKABAbABD8gBCMABAQ&sclient=mobile-gwswiz-serp

https://scholar.google.co.in/scholar_url?url=https://hrcak.srce.hr/file/227601&hl=en&sa=X&ei=NkQnY4feKILgyAS_u7_IDg&scisig=AAGBfm2yU12wg6ePnhQ_lFu4Q_Xr4QyPQ&oi=scholarr

https://jfin-swufe.springeropen.com/articles/10.1186/s40854-016-0029-6

https://ideas.repec.org/a/zna/indecs/v14y2016i2p116-124.html

https://link.springer.com/article/10.1007/s00521-022-07067-x

https://www.semanticscholar.org/paper/Customer-Churn-Analysis-in-Banking-Sector-Khine-Myo/854462e493544f666f3e5d0b06671aff76270e4f

https://thesai.org/Publications/ViewPaper?Volume=9&Issue=11&Code=IJACSA&SerialNo=96

https://www.slideshare.net/BU_Research_Methods/customer-churn-prediction-inbanking

http://dx.doi.org/10.1016/S0377-2217(03)00069-9

http://dx.doi.org/10.1145/1541880.1541883

http://eprints.qut.edu.au/66229/1/Andrew_Ashwood_Thesis.pdf

VII. CODING AND IMPLEMENTATION

Loading libraries and data:

```
In [2]: import pandas as pd
        import numpy as np
        import missingno as msno
        import matplotlib.pyplot as plt
        import seaborn as sns
        import plotly.express as px
        import plotly.graph_objects as go
        from plotly.subplots import make_subplots
        import warnings
        warnings.filterwarnings('ignore')
```

```
In [3]: from sklearn.preprocessing import StandardScaler
        from sklearn.preprocessing import LabelEncoder

        from sklearn.tree import DecisionTreeClassifier
        from sklearn.ensemble import RandomForestClassifier
        from sklearn.naive_bayes import GaussianNB
        from sklearn.neighbors import KNeighborsClassifier
        from sklearn.svm import SVC
        from sklearn.neural_network import MLPClassifier
        from sklearn.ensemble import AdaBoostClassifier
        from sklearn.ensemble import GradientBoostingClassifier
        from sklearn.ensemble import ExtraTreesClassifier
        from sklearn.linear_model import LogisticRegression
        from sklearn.model_selection import train_test_split
        from sklearn.metrics import accuracy_score
        from xgboost import XGBClassifier
        from catboost import CatBoostClassifier
        from sklearn import metrics
        from sklearn.metrics import roc_curve
        from sklearn.metrics import recall_score, confusion_matrix, precision_score, f1_score, accuracy_score, classification_report
```

Understanding the data:

Each row represents a customer, each column contains customer's attributes described on the column Metadata.

```
In [8]: df.head()
```

Out[8]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	Tech
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	

5 rows × 21 columns

The data set includes information about:

- Customers who left within the last month – the column is called Churn
- Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies

- Customer account information - how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges
- Demographic info about customers – gender, age range, and if they have partners and dependents
- The target the we will use to guide the exploration is Churn

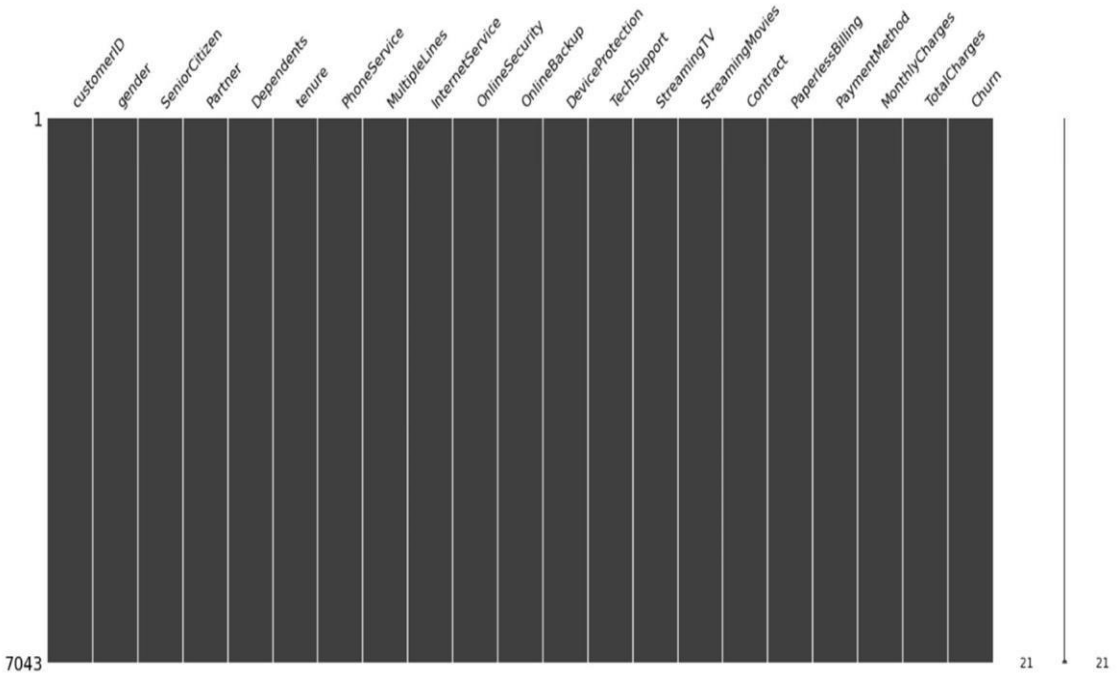
```
In [9]: df.shape
Out[9]: (7043, 21)

In [10]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object
 20  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

```
In [11]: df.columns.values

Out[11]: array(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
       'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
       'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',
       'TotalCharges', 'Churn'], dtype=object)
```

```
In [12]: df.dtypes

Out[12]: customerID          object
         gender              object
         SeniorCitizen        int64
         Partner             object
         Dependents          object
         tenure               int64
         PhoneService        object
         MultipleLines       object
         InternetService     object
         OnlineSecurity      object
         OnlineBackup        object
         DeviceProtection    object
         TechSupport         object
         StreamingTV         object
         StreamingMovies     object
         Contract            object
         PaperlessBilling    object
         PaymentMethod       object
         MonthlyCharges     float64
         TotalCharges        object
         Churn               object
         dtype: object
```

Visualize missing values

```
In [13]: # Visualize missing values as a matrix
         msno.matrix(df);
```



Using this matrix, we can very quickly find the pattern of missingness in the dataset.

From the above visualization we can observe that it has no peculiar pattern that stands out.

In fact, there is no missing data.

Data Manipulation

```
In [14]: df = df.drop(['customerID'], axis = 1)
         df.head()
```

Out[14]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechS
0	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes	No	
1	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes	
2	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	
3	Male	0	No	No	45	No	No phone service	DSL	Yes	No	Yes	
4	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No	

- On deep analysis, we can find some indirect missingness in our data (which can be in form of blank spaces). Let's see that!

```
In [15]: df['TotalCharges'] = pd.to_numeric(df.TotalCharges, errors='coerce')
         df.isnull().sum()
```

Out[15]:

```
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges       11
Churn               0
dtype: int64
```

- Here we see that the Total Charges has 11 missing values. Let's check this data.

```
In [16]: df[np.isnan(df['TotalCharges'])]
```

Out[16]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	Te
488	Female	0	Yes	Yes	0	No	No phone service	DSL	Yes	No	Yes	
753	Male	0	No	Yes	0	Yes	No	No	No internet service	No internet service	No internet service	
936	Female	0	Yes	Yes	0	Yes	No	DSL	Yes	Yes	Yes	
1082	Male	0	Yes	Yes	0	Yes	Yes	No	No internet service	No internet service	No internet service	
1340	Female	0	Yes	Yes	0	No	No phone service	DSL	Yes	Yes	Yes	
3331	Male	0	Yes	Yes	0	Yes	No	No	No internet service	No internet service	No internet service	
3826	Male	0	Yes	Yes	0	Yes	Yes	No	No internet service	No internet service	No internet service	
4380	Female	0	Yes	Yes	0	Yes	No	No	No internet service	No internet service	No internet service	
5218	Male	0	Yes	Yes	0	Yes	No	No	No internet service	No internet service	No internet service	
6670	Female	0	Yes	Yes	0	Yes	Yes	DSL	No	Yes	Yes	
6754	Male	0	No	Yes	0	Yes	Yes	DSL	Yes	Yes	No	

- It can also be noted that the Tenure column is 0 for these entries even though the Monthly Charges column is not empty.

Let's see if there are any other 0 values in the tenure column.

```
In [17]: df[df['tenure'] == 0].index
```

```
Out[17]: Int64Index([488, 753, 936, 1082, 1340, 3331, 3826, 4380, 5218, 6670, 6754], dtype='int64')
```

```
In [18]: df.drop(labels=df[df['tenure'] == 0].index, axis=0, inplace=True)
         df[df['tenure'] == 0].index
```

```
Out[18]: Int64Index([], dtype='int64')
```

- There are no additional missing values in the Tenure column.

Let's delete the rows with missing values in Tenure columns since there are only 11 rows and deleting them will not affect the data.

- To solve the problem of missing values in Total Charges column, I decided to fill it with the mean of Total Charges values.

In [19]:
```
df.fillna(df["TotalCharges"].mean())
```

Out[19]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	Te
0	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes	No	
1	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes	
2	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	
3	Male	0	No	No	45	No	No phone service	DSL	Yes	No	Yes	
4	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No	
...	...	...	...	...	...	...	...	...	...	...	...	...
7038	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	No	Yes	
7039	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	Yes	Yes	
7040	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	No	No	
7041	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No	No	No	
7042	Male	0	No	No	66	Yes	No	Fiber optic	Yes	No	Yes	

7032 rows × 20 columns

In [20]:
```
df.isnull().sum()
```

Out[20]:
```
gender             0
SeniorCitizen      0
Partner            0
Dependents         0
tenure             0
PhoneService       0
MultipleLines      0
InternetService    0
OnlineSecurity     0
OnlineBackup       0
DeviceProtection   0
TechSupport        0
StreamingTV        0
StreamingMovies    0
Contract           0
PaperlessBilling   0
PaymentMethod      0
MonthlyCharges     0
TotalCharges       0
Churn              0
dtype: int64
```

In [22]:
```
df["SeniorCitizen"]= df["SeniorCitizen"].map({0: "No", 1: "Yes"})
df.head()
```

Out[22]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechS
0	Female	NaN	Yes	No	1	No	No phone service	DSL	No	Yes	No	
1	Male	NaN	No	No	34	Yes	No	DSL	Yes	No	Yes	
2	Male	NaN	No	No	2	Yes	No	DSL	Yes	Yes	No	
3	Male	NaN	No	No	45	No	No phone service	DSL	Yes	No	Yes	
4	Female	NaN	No	No	2	Yes	No	Fiber optic	No	No	No	

In [23]:
```
df["InternetService"].describe(include=['object', 'bool'])
```

Out[23]:
```
count            7032
unique              3
top       Fiber optic
freq             3096
Name: InternetService, dtype: object
```

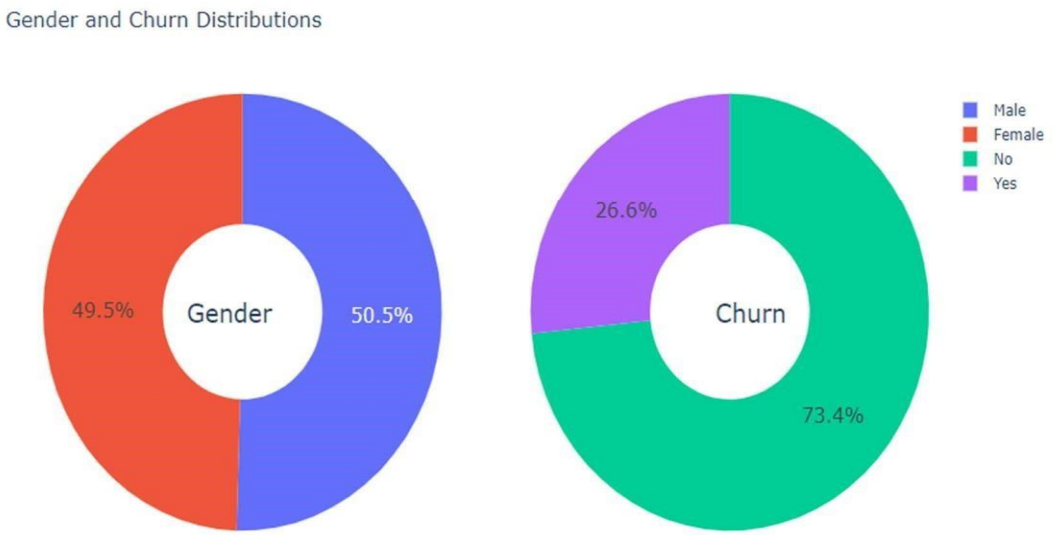
In [24]:
```
numerical_cols = ['tenure', 'MonthlyCharges', 'TotalCharges']
df[numerical_cols].describe()
```

Out[24]:

	tenure	MonthlyCharges	TotalCharges
count	7032.000000	7032.000000	7032.000000
mean	32.421786	64.798208	2283.300441
std	24.545260	30.085974	2266.771362
min	1.000000	18.250000	18.800000
25%	9.000000	35.587500	401.450000
50%	29.000000	70.350000	1397.475000
75%	55.000000	89.862500	3794.737500
max	72.000000	118.750000	8684.800000

Data Visualization

In [25]:
```
g_labels = ['Male', 'Female']
c_labels = ['No', 'Yes']
# Create subplots: use 'domain' type for Pie subplot
fig = make_subplots(rows=1, cols=2, specs=[[{'type':'domain'}, {'type':'domain'}]])
fig.add_trace(go.Pie(labels=g_labels, values=df['gender'].value_counts(), name="Gender"),
              1, 1)
fig.add_trace(go.Pie(labels=c_labels, values=df['Churn'].value_counts(), name="Churn"),
              1, 2)

# Use `hole` to create a donut-like pie chart
fig.update_traces(hole=.4, hoverinfo="label+percent+name", textfont_size=16)

fig.update_layout(
    title_text="Gender and Churn Distributions",
    # Add annotations in the center of the donut pies.
    annotations=[dict(text='Gender', x=0.16, y=0.5, font_size=20, showarrow=False),
                 dict(text='Churn', x=0.84, y=0.5, font_size=20, showarrow=False)])
fig.show()
```



- 26.6 % of customers switched to another firm.
- Customers are 49.5 % female and 50.5 % male.

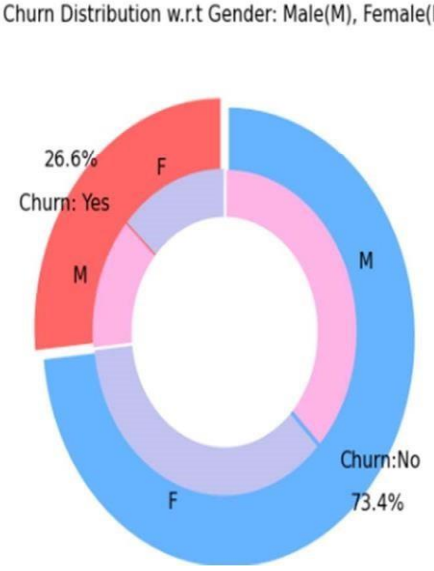
In [26]:
```
df["Churn"][df["Churn"]=="No"].groupby(by=df["gender"]).count()
```

Out[26]:
```
gender
Female    2544
Male      2619
Name: Churn, dtype: int64
```

In [27]:
```
df["Churn"][df["Churn"]=="Yes"].groupby(by=df["gender"]).count()
```

Out[27]:
```
gender
Female    939
Male      930
Name: Churn, dtype: int64
```

In [28]:
```
plt.figure(figsize=(6, 6))
labels =["Churn: Yes","Churn:No"]
values = [1869,5163]
labels_gender = ["F","M","F","M"]
sizes_gender = [939,930 , 2544,2619]
colors = ['#ff6666', '#66b3ff']
colors_gender = ['#c2c2f0','#ffb3e6', '#c2c2f0','#ffb3e6']
explode = (0.3,0.3)
explode_gender = (0.1,0.1,0.1,0.1)
textprops = {"fontsize":15}
#Plot
plt.pie(values, labels=labels,autopct='%1.1f%%',pctdistance=1.08, labeldistance=0.8,colors=colors, startangle=90,frame=True,
plt.pie(sizes_gender,labels=labels_gender,colors=colors_gender,startangle=90, explode=explode_gender,radius=7, textprops =tex
#Draw circle
centre_circle = plt.Circle((0,0),5,color='black', fc='white',linewidth=0)
fig = plt.gcf()
fig.gca().add_artist(centre_circle)

plt.title('Churn Distribution w.r.t Gender: Male(M), Female(F)', fontsize=15, y=1.1)

# show plot

plt.axis('equal')
plt.tight_layout()
plt.show()
```

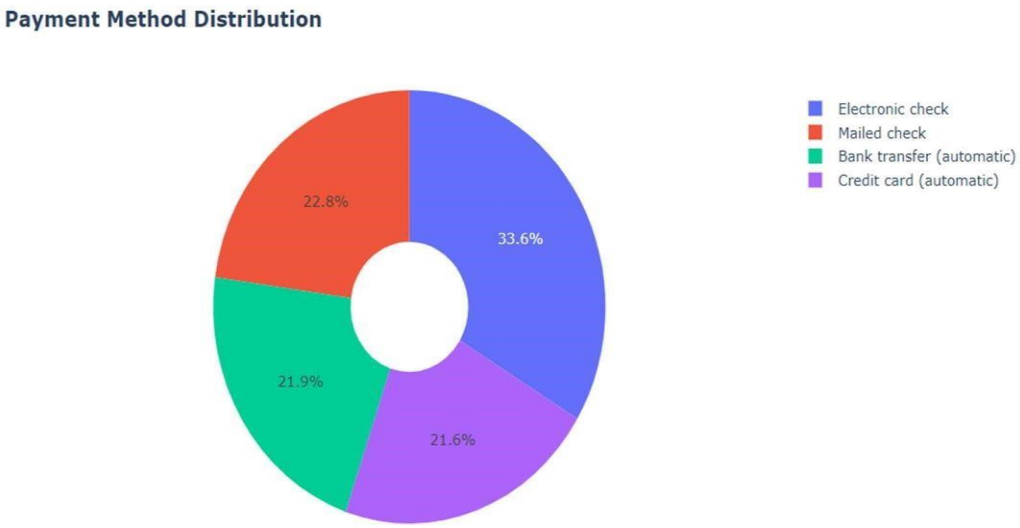


- There is negligible difference in customer percentage/ count who changed the service provider. Both genders behaved in similar fashion when it comes to migrating to another service provider/firm.

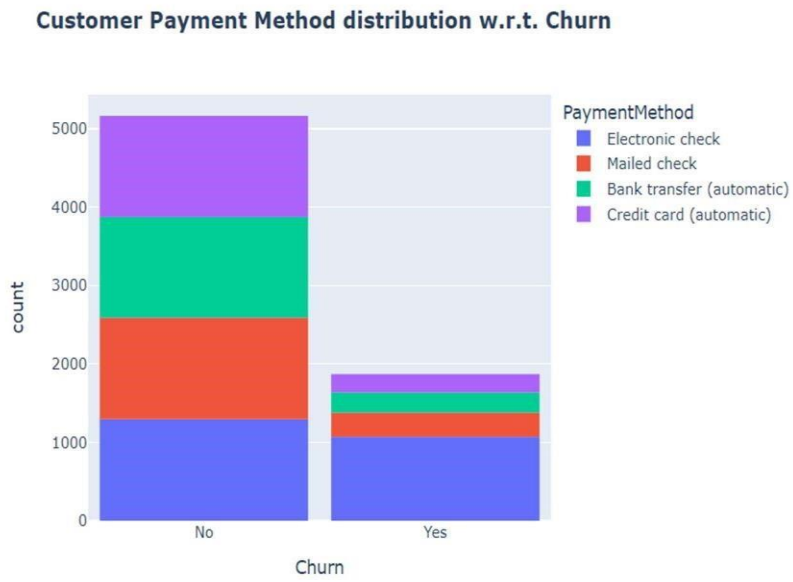
In [29]:
```
fig = px.histogram(df, x="Churn", color="Contract", barmode="group", title="<b>Customer contract distribution<b>")
fig.update_layout(width=700, height=500, bargap=0.1)
fig.show()
```

- About 75% of customer with Month-to-Month Contract opted to move out as compared to 13% of customers with One Year Contract and 3% with Two Year Contract

In [30]:
```
labels = df['PaymentMethod'].unique()
values = df['PaymentMethod'].value_counts()

fig = go.Figure(data=[go.Pie(labels=labels, values=values, hole=.3)])
fig.update_layout(title_text="<b>Payment Method Distribution</b>")
fig.show()
```



```
In [31]: fig = px.histogram(df, x="Churn", color="PaymentMethod", title="<b>Customer Payment Method distribution w.r.t. Churn</b>")
         fig.update_layout(width=700, height=500, bargap=0.1)
         fig.show()
```

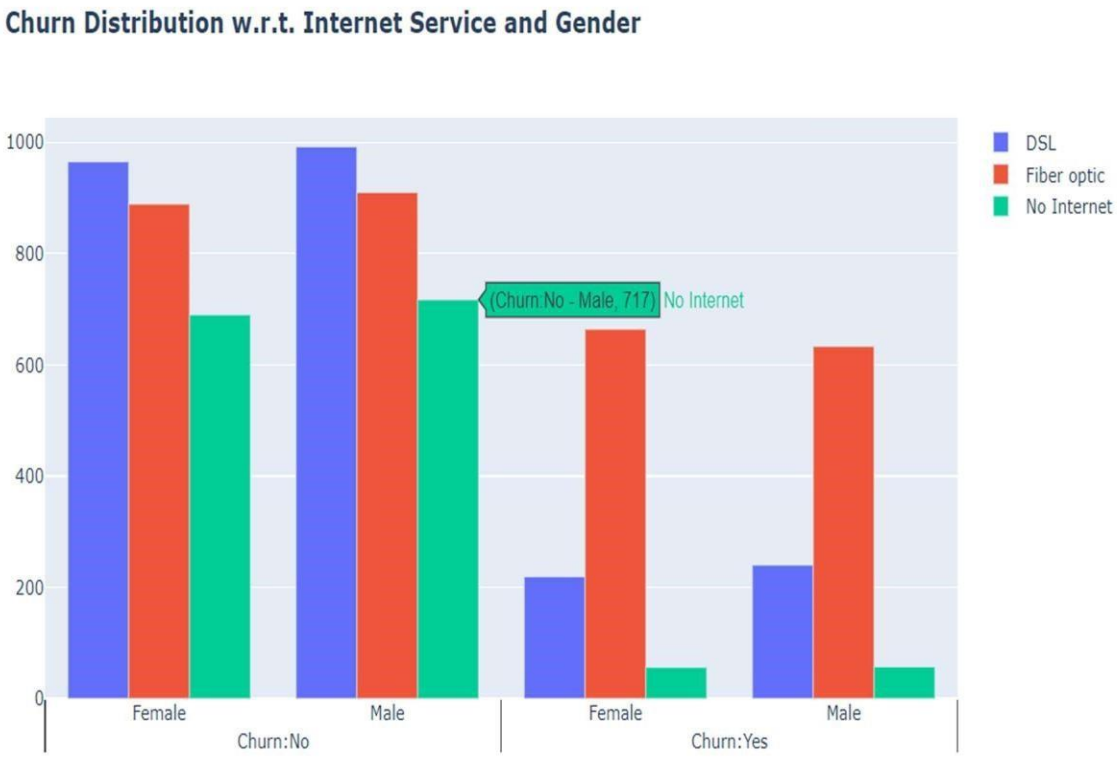


- Major customers who moved out were having Electronic Check as Payment Method.
- Customers who opted for Credit-Card automatic transfer or Bank Automatic

Transfer and Mailed Check as Payment Method were less likely to move out.

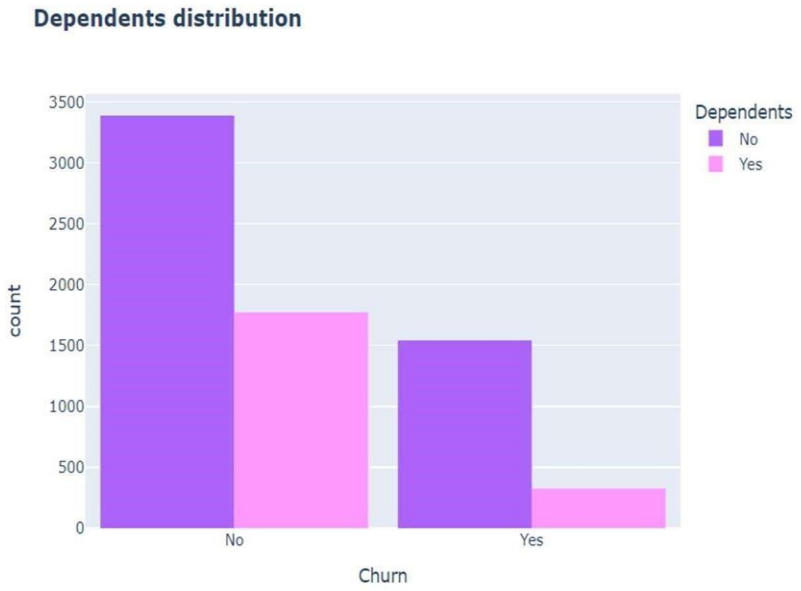
```
In [32]: df["InternetService"].unique()

Out[32]: array(['DSL', 'Fiber optic', 'No'], dtype=object)

In [33]: df[df["gender"]=="Male"][["InternetService", "Churn"]].value_counts()

Out[33]: InternetService  Churn
         DSL              No       992
         Fiber optic      No       910
         No               No       717
         Fiber optic      Yes      633
         DSL              Yes      240
         No               Yes       57
         dtype: int64

In [34]: df[df["gender"]=="Female"][["InternetService", "Churn"]].value_counts()

Out[34]: InternetService  Churn
         DSL              No       965
         Fiber optic      No       889
         No               No       690
         Fiber optic      Yes      664
         DSL              Yes      219
         No               Yes       56
         dtype: int64
```

In [35]:
```
fig = go.Figure()

fig.add_trace(go.Bar(
  x = [['Churn:No', 'Churn:No', 'Churn:Yes', 'Churn:Yes'],
       ["Female", "Male", "Female", "Male"]],
  y = [965, 992, 219, 240],
  name = 'DSL',
))

fig.add_trace(go.Bar(
  x = [['Churn:No', 'Churn:No', 'Churn:Yes', 'Churn:Yes'],
       ["Female", "Male", "Female", "Male"]],
  y = [889, 910, 664, 633],
  name = 'Fiber optic',
))

fig.add_trace(go.Bar(
  x = [['Churn:No', 'Churn:No', 'Churn:Yes', 'Churn:Yes'],
       ["Female", "Male", "Female", "Male"]],
  y = [690, 717, 56, 57],
  name = 'No Internet',
))

fig.update_layout(title_text="<b>Churn Distribution w.r.t. Internet Service and Gender</b>")

fig.show()
```



- A lot of customers choose the Fiber optic service and it's also evident that the customers who use Fiber optic have high churn rate, this might suggest a dissatisfaction with this type of internet service.
- Customers having DSL service are majority in number and have less churn rate compared to Fiber optic service.

In [36]:
```
color_map = {"Yes": "#FF97FF", "No": "#AB63FA"}
fig = px.histogram(df, x="Churn", color="Dependents", barmode="group", title="<b>Dependents distribution</b>", color_discrete
fig.update_layout(width=700, height=500, bargap=0.1)
fig.show()
```



- Customers without dependents are more likely to churn

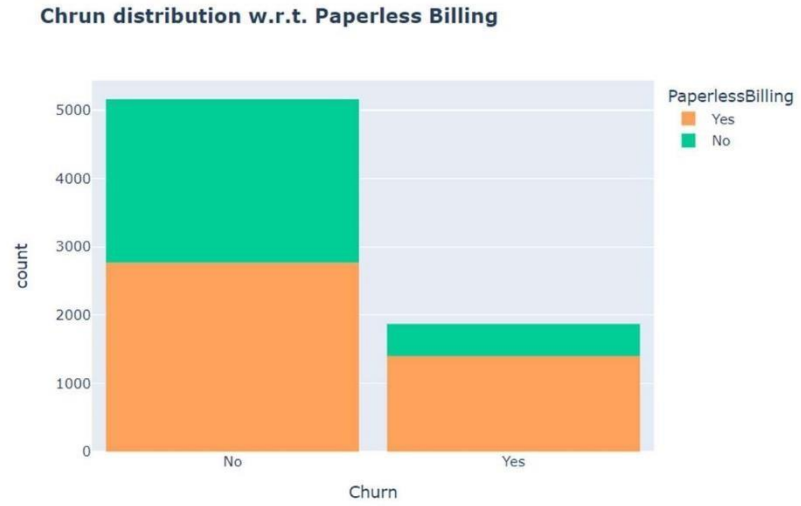
In [37]:
```
color_map = {"Yes": '#FFA15A', "No": '#00CC96'}
fig = px.histogram(df, x="Churn", color="Partner", barmode="group", title="<b>Chrun distribution w.r.t. Partners</b>", color_
fig.update_layout(width=700, height=500, bargap=0.1)
fig.show()
```

- Customers that don’t have partners are more likely to churn

In [41]:
```
color_map = {"Yes": "#FF97FF", "No": "#AB63FA"}
fig = px.histogram(df, x="Churn", color="OnlineSecurity", barmode="group", title="<b>Churn w.r.t Online Security</b>", color_
fig.update_layout(width=700, height=500, bargap=0.1)
fig.show()
```

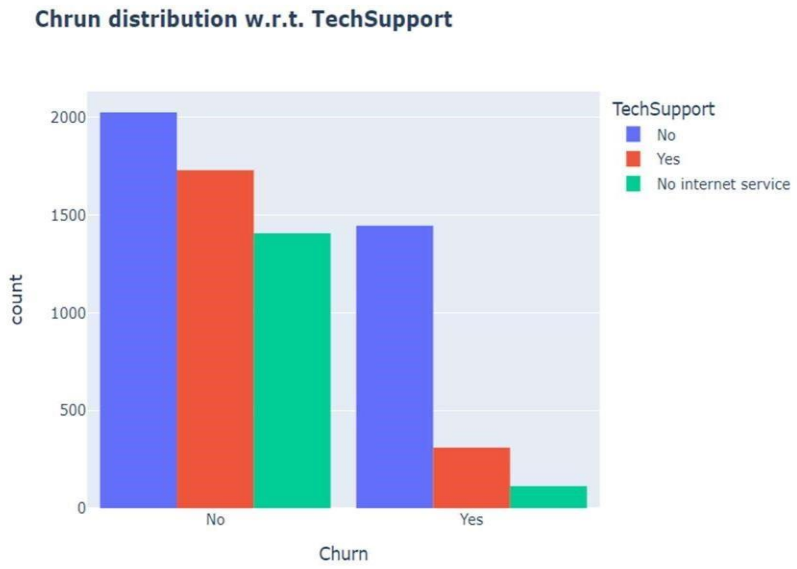
- Most customers churn in the absence of online security,

In [42]:
```
color_map = {"Yes": '#FFA15A', "No": '#00CC96'}
fig = px.histogram(df, x="Churn", color="PaperlessBilling",  title="<b>Chrun distribution w.r.t. Paperless Billing</b>", colo
fig.update_layout(width=700, height=500, bargap=0.1)
fig.show()
```



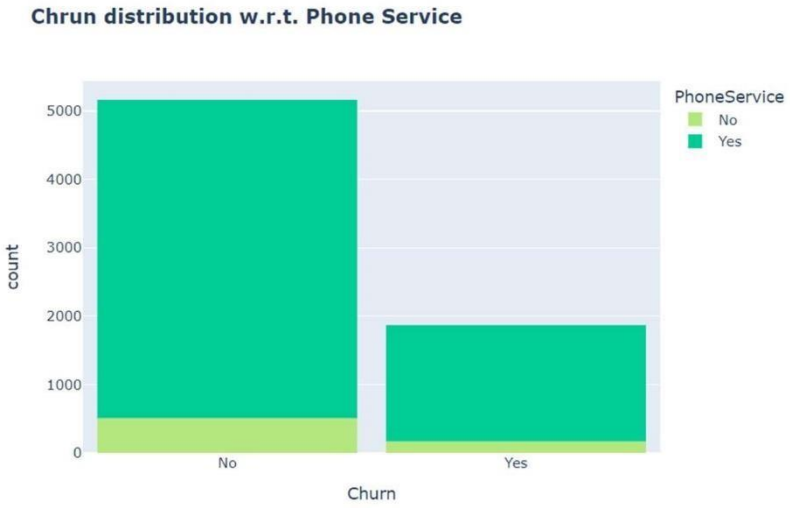
- Customers with Paperless Billing are most likely to churn.

```
In [43]: fig = px.histogram(df, x="Churn", color="TechSupport",barmode="group",  title="<b>Chrun distribution w.r.t. TechSupport</b>")
         fig.update_layout(width=700, height=500, bargap=0.1)
         fig.show()
```



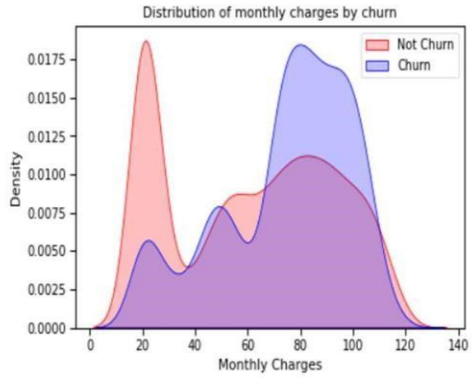
- Customers with no TechSupport are most likely to migrate to another service provider.

```
In [44]: color_map = {"Yes": '#00CC96', "No": '#B6E880'}
         fig = px.histogram(df, x="Churn", color="PhoneService", title="<b>Chrun distribution w.r.t. Phone Service</b>", color_discret
         fig.update_layout(width=700, height=500, bargap=0.1)
         fig.show()
```



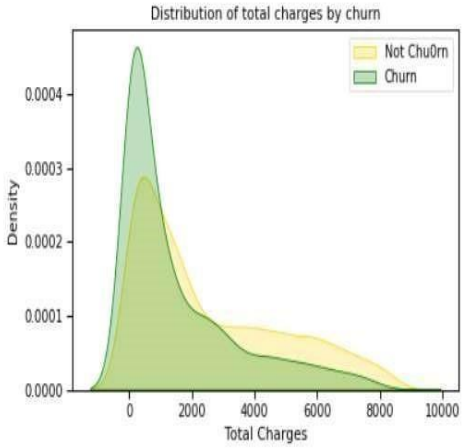
- Very small fraction of customers doesn't have a phone service and out of that, 1/3rd Customers are more likely to churn.

In [45]:
```
sns.set_context("paper",font_scale=1.1)
ax = sns.kdeplot(df.MonthlyCharges[(df["Churn"] == 'No') ],
                 color="Red", shade = True);
ax = sns.kdeplot(df.MonthlyCharges[(df["Churn"] == 'Yes') ],
                 ax =ax, color="Blue", shade= True);
ax.legend(["Not Churn","Churn"],loc='upper right');
ax.set_ylabel('Density');
ax.set_xlabel('Monthly Charges');
ax.set_title('Distribution of monthly charges by churn');
```



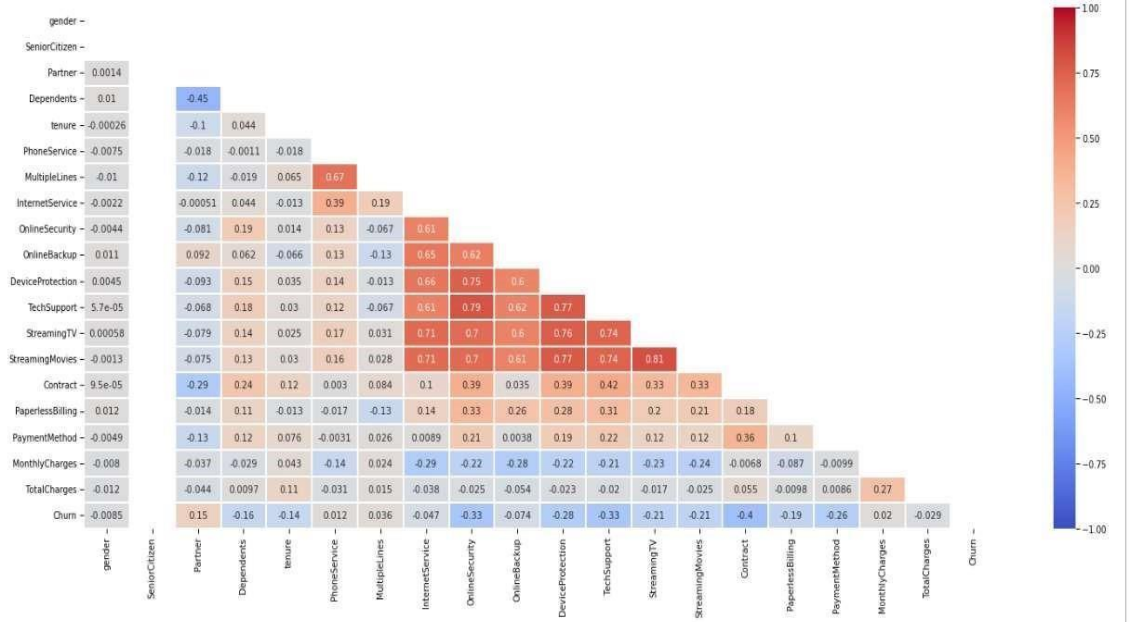
- Customers with higher Monthly Charges are also more likely to churn

In [46]:
```
ax = sns.kdeplot(df.TotalCharges[(df["Churn"] == 'No') ],
                 color="Gold", shade = True);
ax = sns.kdeplot(df.TotalCharges[(df["Churn"] == 'Yes') ],
                 ax =ax, color="Green", shade= True);
ax.legend(["Not Chu0rn","Churn"],loc='upper right');
ax.set_ylabel('Density');
ax.set_xlabel('Total Charges');
ax.set_title('Distribution of total charges by churn');
```



- New customers are more likely to churn

In [48]:
```
plt.figure(figsize=(25, 10))

corr = df.apply(lambda x: pd.factorize(x)[0]).corr()

mask = np.triu(np.ones_like(corr, dtype=bool))

ax = sns.heatmap(corr, mask=mask, xticklabels=corr.columns, yticklabels=corr.columns, annot=True, linewidths=.2, cmap='coolwa
```



Data Preprocessing

Splitting the data into train and test sets:

In [49]:
```
def object_to_int(dataframe_series):
    if dataframe_series.dtype=='object':
        dataframe_series = LabelEncoder().fit_transform(dataframe_series)
    return dataframe_series
```

In [50]:
```
df = df.apply(lambda x: object_to_int(x))
df.head()
```

Out[50]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechS
0	0	0	1	0	1	0	1	0	0	2	0	
1	1	0	0	0	34	1	0	0	2	0	2	
2	1	0	0	0	2	1	0	0	2	2	0	
3	1	0	0	0	45	0	1	0	2	0	2	
4	0	0	0	0	2	1	0	1	0	0	0	

In [51]:
```
plt.figure(figsize=(14,7))
df.corr()['Churn'].sort_values(ascending = False)
```

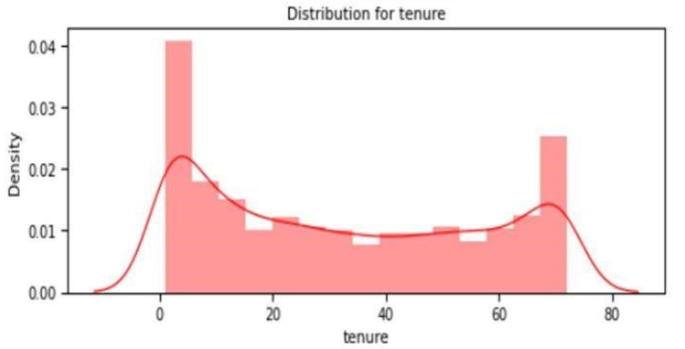
Out[51]:
```
Churn              1.000000
MonthlyCharges     0.192858
PaperlessBilling   0.191454
PaymentMethod      0.107852
MultipleLines      0.038043
PhoneService       0.011691
gender            -0.008545
StreamingTV       -0.036303
StreamingMovies   -0.038802
InternetService   -0.047097
Partner           -0.149982
Dependents        -0.163128
DeviceProtection  -0.177883
OnlineBackup      -0.195290
TotalCharges      -0.199484
TechSupport       -0.282232
OnlineSecurity    -0.289050
tenure            -0.354049
Contract          -0.396150
SeniorCitizen           NaN
Name: Churn, dtype: float64

<Figure size 1008x504 with 0 Axes>
```

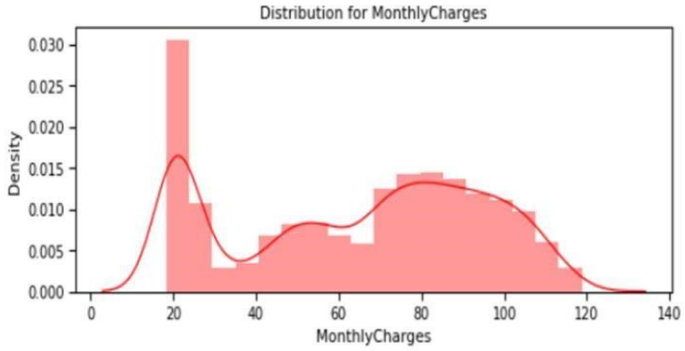
In [52]:
```
X = df.drop(columns = ['Churn'])
y = df['Churn'].values
```

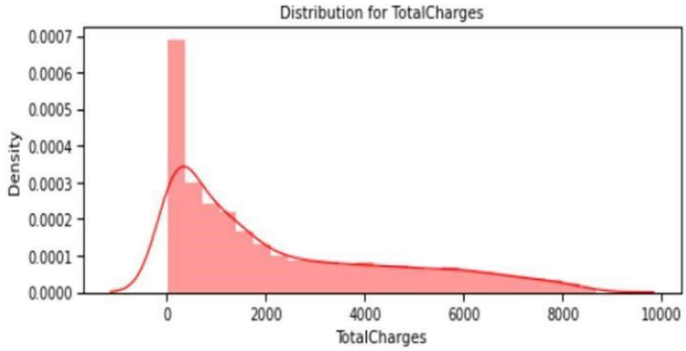
In [53]:
```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = 0.30, random_state = 40, stratify=y)
```

In [54]:
```
def distplot(feature, frame, color='r'):
    plt.figure(figsize=(8,3))
    plt.title("Distribution for {}".format(feature))
    ax = sns.distplot(frame[feature], color= color)
```

In [55]:
```
num_cols = ["tenure", 'MonthlyCharges', 'TotalCharges']
for feat in num_cols: distplot(feat, df)
```



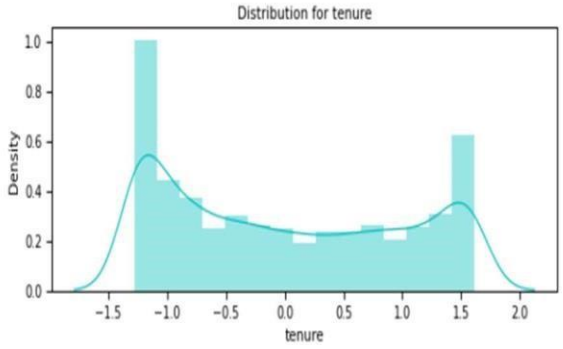


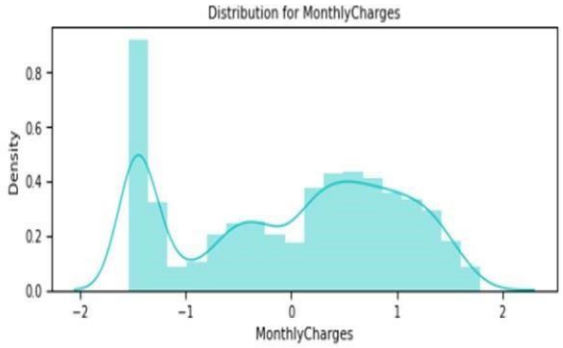


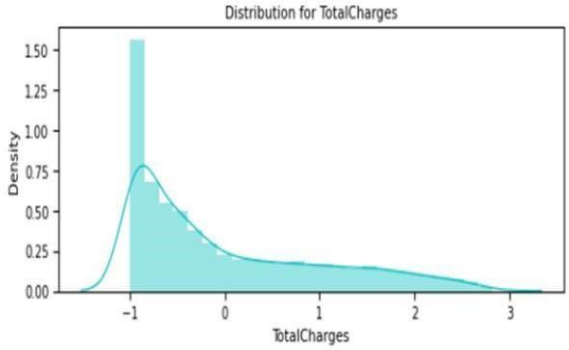
Since the numerical features are distributed over different value ranges, I will use standard scalar to scale them down to the same range.

Standardizing numeric attributes:

In [56]:
```
df_std = pd.DataFrame(StandardScaler().fit_transform(df[num_cols].astype('float64')),
                      columns=num_cols)
for feat in numerical_cols: distplot(feat, df_std, color='c')
```







In [57]:
```
# Divide the columns into 3 categories, one ofor standardisation, one for label encoding and one for one hot encoding

cat_cols_ohe =['PaymentMethod', 'Contract', 'InternetService'] # those that need one-hot encoding
cat_cols_le = list(set(X_train.columns)- set(num_cols) - set(cat_cols_ohe)) #those that need label encoding
```

In [58]:
```
scaler= StandardScaler()

X_train[num_cols] = scaler.fit_transform(X_train[num_cols])
X_test[num_cols] = scaler.transform(X_test[num_cols])
```

Machine Learning Model Evaluations and Predictions

KNN

In [59]:
```
knn_model = KNeighborsClassifier(n_neighbors = 11)
knn_model.fit(X_train,y_train)
predicted_y = knn_model.predict(X_test)
accuracy_knn = knn_model.score(X_test,y_test)
print("KNN accuracy:",accuracy_knn)
```

```
KNN accuracy: 0.7729857819905214
```

In [60]:
```
print(classification_report(y_test, predicted_y))
```

```
              precision    recall  f1-score   support

           0       0.83      0.86      0.85      1549
           1       0.58      0.52      0.55       561

    accuracy                           0.77      2110
   macro avg       0.71      0.69      0.70      2110
weighted avg       0.77      0.77      0.77      2110
```

SVC

In [61]:
```
svc_model = SVC(random_state = 1)
svc_model.fit(X_train,y_train)
predict_y = svc_model.predict(X_test)
accuracy_svc = svc_model.score(X_test,y_test)
print("SVM accuracy is :",accuracy_svc)
```

```
SVM accuracy is : 0.8080568720379147
```

In [62]:
```
print(classification_report(y_test, predict_y))
```

```
              precision    recall  f1-score   support

           0       0.83      0.92      0.88      1549
           1       0.69      0.50      0.58       561

    accuracy                           0.81      2110
   macro avg       0.76      0.71      0.73      2110
weighted avg       0.80      0.81      0.80      2110
```

Random Forest

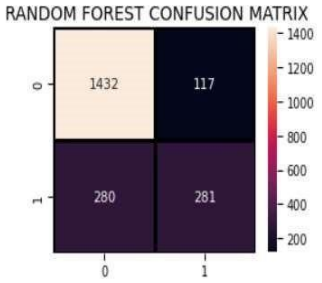
In [63]:
```
model_rf = RandomForestClassifier(n_estimators=500 , oob_score = True, n_jobs = -1,
                                  random_state =50, max_features = "auto",
                                  max_leaf_nodes = 30)
model_rf.fit(X_train, y_train)

# Make predictions
prediction_test = model_rf.predict(X_test)
print (metrics.accuracy_score(y_test, prediction_test))
```

```
0.8118483412322275
```

In [64]:
```
print(classification_report(y_test, prediction_test))
```

```
              precision    recall  f1-score   support

           0       0.84      0.92      0.88      1549
           1       0.71      0.50      0.59       561

    accuracy                           0.81      2110
   macro avg       0.77      0.71      0.73      2110
weighted avg       0.80      0.81      0.80      2110
```

In [65]:
```
plt.figure(figsize=(4,3))
sns.heatmap(confusion_matrix(y_test, prediction_test),
                annot=True,fmt = "d",linecolor="k",linewidths=3)

plt.title(" RANDOM FOREST CONFUSION MATRIX",fontsize=14)
plt.show()
```



Logistic Regression

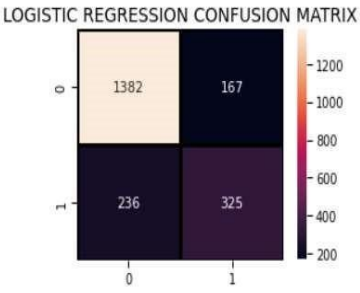
In [66]:
```
lr_model = LogisticRegression()
lr_model.fit(X_train,y_train)
accuracy_lr = lr_model.score(X_test,y_test)
print("Logistic Regression accuracy is :",accuracy_lr)
```

```
Logistic Regression accuracy is : 0.8090047393364929
```

In [67]:
```
lr_pred= lr_model.predict(X_test)
report = classification_report(y_test,lr_pred)
print(report)
```

```
              precision    recall  f1-score   support

           0       0.85      0.89      0.87      1549
           1       0.66      0.58      0.62       561

    accuracy                           0.81      2110
   macro avg       0.76      0.74      0.75      2110
weighted avg       0.80      0.81      0.80      2110
```

In [68]:
```
plt.figure(figsize=(4,3))
sns.heatmap(confusion_matrix(y_test, lr_pred),
                annot=True,fmt = "d",linecolor="k",linewidths=3)

plt.title("LOGISTIC REGRESSION CONFUSION MATRIX",fontsize=14)
plt.show()
```



Decision Tree Classifier

In [69]:
```
dt_model = DecisionTreeClassifier()
dt_model.fit(X_train,y_train)
predictdt_y = dt_model.predict(X_test)
accuracy_dt = dt_model.score(X_test,y_test)
print("Decision Tree accuracy is :",accuracy_dt)
```

```
Decision Tree accuracy is : 0.7255924170616114
```

In [70]:
```
print(classification_report(y_test, predictdt_y))
```

```
              precision    recall  f1-score   support

           0       0.82      0.80      0.81      1549
           1       0.48      0.52      0.50       561

    accuracy                           0.73      2110
   macro avg       0.65      0.66      0.66      2110
weighted avg       0.73      0.73      0.73      2110
```

AdaBoost Classifier

In [71]:
```
a_model = AdaBoostClassifier()
a_model.fit(X_train,y_train)
a_preds = a_model.predict(X_test)
print("AdaBoost Classifier accuracy")
metrics.accuracy_score(y_test, a_preds)
```

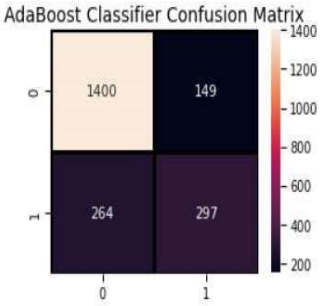
```
AdaBoost Classifier accuracy
```

Out[71]:
```
0.8042654028436019
```

In [72]:
```
print(classification_report(y_test, a_preds))
```

```
              precision    recall  f1-score   support

           0       0.84      0.90      0.87      1549
           1       0.67      0.53      0.59       561

    accuracy                           0.80      2110
   macro avg       0.75      0.72      0.73      2110
weighted avg       0.79      0.80      0.80      2110
```

In [73]:
```
plt.figure(figsize=(4,3))
sns.heatmap(confusion_matrix(y_test, a_preds),
                annot=True,fmt = "d",linecolor="k",linewidths=3)

plt.title("AdaBoost Classifier Confusion Matrix",fontsize=14)
plt.show()
```



Gradient Boosting Classifier

In [74]:
```
gb = GradientBoostingClassifier()
gb.fit(X_train, y_train)
gb_pred = gb.predict(X_test)
print("Gradient Boosting Classifier", accuracy_score(y_test, gb_pred))
```

```
Gradient Boosting Classifier 0.8080568720379147
```

In [75]:
```
print(classification_report(y_test, gb_pred))
```

```
              precision    recall  f1-score   support

           0       0.85      0.90      0.87      1549
           1       0.67      0.55      0.60       561

    accuracy                           0.81      2110
   macro avg       0.76      0.73      0.74      2110
weighted avg       0.80      0.81      0.80      2110
```

In [76]:
```
plt.figure(figsize=(4,3))
sns.heatmap(confusion_matrix(y_test, gb_pred),
                annot=True,fmt = "d",linecolor="k",linewidths=3)

plt.title("Gradient Boosting Classifier Confusion Matrix",fontsize=14)
plt.show()
```

Voting Classifier

Let's now predict the final model based on the highest majority of voting and check its score.

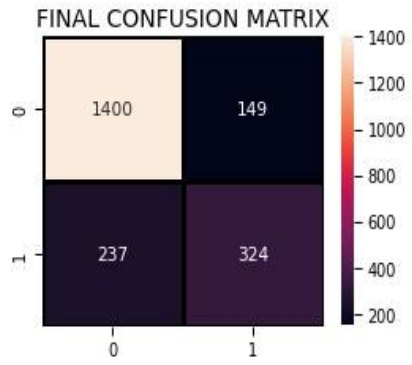
```
from sklearn.ensemble import VotingClassifier
clf1 = GradientBoostingClassifier()
clf2 = LogisticRegression()
clf3 = AdaBoostClassifier()
eclf1 = VotingClassifier(estimators=[('gbc', clf1), ('lr', clf2), ('abc', clf3)], voting='soft')
eclf1.fit(X_train, y_train)
predictions = eclf1.predict(X_test)
print("Final Accuracy Score ")
print(accuracy_score(y_test, predictions)*100)
```

```
Final Accuracy Score
81.70616113744076
```

```
print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           0       0.86      0.90      0.88      1549
           1       0.68      0.58      0.63       561

    accuracy                           0.82      2110
   macro avg       0.77      0.74      0.75      2110
weighted avg       0.81      0.82      0.81      2110
```

```
plt.figure(figsize=(4,3))
sns.heatmap(confusion_matrix(y_test, predictions),
                annot=True,fmt = "d",linecolor="k",linewidths=3)

plt.title("FINAL CONFUSION MATRIX",fontsize=14)
plt.show()
```



****************************THANK_YOU**