VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

COURSE NAME

RISK AND FRAUD ANALYTICS

(MGT3013)

SLOT

D1 + TD1

FACULTY

Dr BHUVANESH C

PROJECT TITLE

CUSTOMER CHURN ANALYSIS IN TELECOM INDUSTRY

TEAM MEMBERS

ALAGARSAMY N | 19MIA1082

NIRANJAN J | 19MIA1003

VIGNESH N | 19MIA1093

ROSHAN SRINIVAAS. S | 19MIA1001

T.S.S. ABINANDHAN KUMAR | 19MIA1062

FALL SEMESTER 2022-2023

INTRODUCTION:

Customer churn analysis and prediction in telecom sector is an issue now a days because it's very important for telecommunication industries to analyse behaviours of various customer to predict which customers are about to leave the from telecom company, So data mining techniques and algorithm plays an important role for companies in today's commercial conditions because gaining a new customer's cost is more than retaining the existing ones. We have planned to focus on various machine learning techniques for predicting customer churn through which we can build the classification models and also compare the performance of these models. Finding the causes of client loss, gauging customer loyalty, and winning back customers have all become crucial topics for many businesses. Instead of recruiting new clients, businesses do a variety of studies and initiatives to keep the ones they already have. The telecommunications industry collects enormous amounts of data as a result of fast developing technology, an increase in subscribers, and value-added services. This industry's unchecked and rapid expansion result in growing losses due to risk, fraud and technological challenges. Therefore, the creation of novel analysis techniques has become essential.

LITERATURE REVIEW:

[1] M.A.H. Farquad, proposed a hybrid approach to overcome the drawbacks of general SVM model which generates a black box model (i.e., it does not reveal the knowledge gained during training in human understandable form).

[2] introduced the hybrid neural networks techniques to predict the customer churners in a CRM dataset provided by American telecom companies. Here, they built two hybrid models by combining two different neural network International Journal of Computer Applications (0975 – 8887).

[3] Wouter Verbeke, proposed the application of Ant-Miner+ and ALBA algorithms on a publicly available churn prediction dataset in order to build accurate as well as comprehensible classification rule-sets churn prediction models.

[4] Ning Lu, proposed the use of boosting algorithms to enhance a customer churn prediction model in which customers are separated into two clusters based on the weight assigned by the

boosting algorithm. As a result, a high risky customer cluster has been found. Logistic regression is used as a basis learner, and a churn prediction model is built on each cluster, respectively. The experimental results showed that boosting algorithm provides a good separation of churn data when compared with a single logistic regression model.

[5] Benlan He, suggested a customer churn prediction methodology based on SVM model, and used random sampling method to improve SVM model by considering the imbalance characteristics of customer data sets. A support vector machine constructs a hyper-plane in a high- or infinite dimensional space, which can be used for classification. Random sampling method can be used to change the distribution of data in order to reduce the imbalance of the dataset. Imbalance in dataset is caused due to the low proportion of churners.

[6] Ssu-Han Chen, used a novel mechanism based on the gamma Cumulative SUM (CUSUM) chart in which the gamma CUSUM chart monitors individual customer's Inter Arrival Time (IAT) by introducing a finite mixture model to design the reference value and decision interval of the chart and used a hierarchical Bayesian model to capture the heterogeneity of customers. Recency, another time interval variable which is complementary to IAT, is combined into the model and tracks the recent status of the login behavior. In addition, benefits from the basic nature of control charts, the graphical interface for each customer is an additional advantage of the proposed method. The results showed that the accuracy rate (ACC) for gamma CUSUM chart is 5.2% higher than exponential CUSUM and the Average Time to Signal (ATS) is about two days longer than required for exponential CUSUM.

[7] Koen W. De Bock [10] proposed two rotation-based ensemble classifiers namely Rotation Forest and Rotboost as modeling techniques for customer churn prediction. An ensemble classifier is a combination of several member classifier models into one aggregated model, including the fusion rule to combine member classifiers outputs. In Rotation Forests, feature extraction is applied to feature subsets in order to turn the input data for training base classifiers, while RotBoost combines Rotation Forest with AdaBoost. Four data sets from

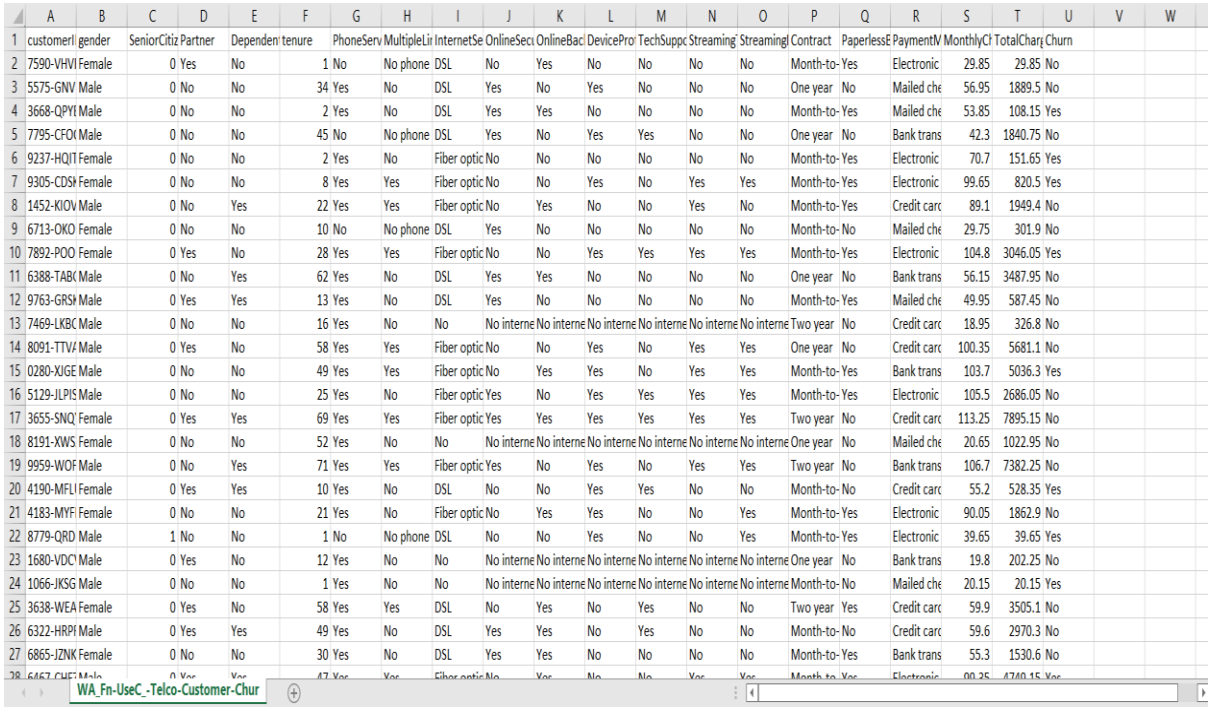
real-life customer churn prediction projects are used here. The results showed that Rotation Forests outperform RotBoost in terms of area under the curve (AUC) and top-decile lift, while RotBoost demonstrates higher accuracy than Rotation Forests. They also compared three alternative feature extraction algorithms namely: Principal Component Analysis (PCA), Independent Component Analysis (ICA) and Sparse Random Projections (SRP) on classification performance of both RotBoost and Rotation Forest.

[8] Ver-braken et al. [14] proposed a new performance measure called the expected maximum profit criterion, which is aligned with the main objectives of the end users. The proposed framework not only assists the companies with selecting the classifier that maximizes the profit, but also provides information about the fraction of the customer base to be included in the retention campaign.

[9] P.C.Pendharkar [15] suggested two Genetic Algorithm(GA) based neural network (NN) models to predict the customer churn. The first GA-based NN model used a cross entropy based criteria to predict customer churn, and the second GA based NN model made some efforts to directly increase the prediction accuracy of customer churn. Using real-world customer dataset and three various sizes of NNs, they compared the two GA-based NN models with a statistical zscore model using model evaluation criterion like prediction accuracy, top 10% docile lift and area under Receiver Operating Characteristics (ROC) curve. The results of experiments indicated that both GA-based NN models outperform the statistical z-score model on all performance criteria.

[10] Y.Xie et al., [16] used an improved balance random forest (IBFR) model which is a combination of balanced random forests and weighted random forests in order to overcome the data distribution problem. The nature of IBRF is that the best features are iteratively learned by altering the class distribution and by putting higher penalties on misclassification of the minority class. The experiments are carried out with Chinese bank dataset which showed that IBRF is better than artificial neural network, decision tree and support vector machines in terms of accuracy.

DATASET DESCRIPTION:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	customerI	gender	SeniorCitiz	Partner	Dependen	tenure	PhoneServ	MultipleLir	InternetSe	OnlineSecu	OnlineBac	DevicePro	TechSuppc	Streaming	StreamingI	Contract	PaperlessB	PaymentM	MonthlyCh	TotalChar	Churn
2	7590-VHVI	Female	0	Yes	No	1	No	No phone	DSL	No	Yes	No	No	No	No	Month-to-	Yes	Electronic	29.85	29.85	No
3	5575-GNV	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes	No	No	No	One year	No	Mailed che	56.95	1889.5	No
4	3668-QPY	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	No	No	No	Month-to-	Yes	Mailed che	53.85	108.15	Yes
5	7795-CFO	Male	0	No	No	45	No	No phone	DSL	Yes	No	Yes	Yes	No	No	One year	No	Bank trans	42.3	1840.75	No
6	9237-HQIT	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No	No	No	No	Month-to-	Yes	Electronic	70.7	151.65	Yes
7	9305-CDS	Female	0	No	No	8	Yes	Yes	Fiber optic	No	No	Yes	No	Yes	Yes	Month-to-	Yes	Electronic	99.65	820.5	Yes
8	1452-KIOV	Male	0	No	Yes	22	Yes	Yes	Fiber optic	No	Yes	No	No	Yes	No	Month-to-	Yes	Credit carc	89.1	1949.4	No
9	6713-OKO	Female	0	No	No	10	No	No phone	DSL	Yes	No	No	No	No	No	Month-to-	No	Mailed che	29.75	301.9	No
10	7892-POO	Female	0	Yes	No	28	Yes	Yes	Fiber optic	No	No	Yes	Yes	Yes	Yes	Month-to-	Yes	Electronic	104.8	3046.05	Yes
11	6388-TABC	Male	0	No	Yes	62	Yes	No	DSL	Yes	Yes	No	No	No	No	One year	No	Bank trans	56.15	3487.95	No
12	9763-GRS	Male	0	Yes	Yes	13	Yes	No	DSL	Yes	No	No	No	No	No	Month-to-	Yes	Mailed che	49.95	587.45	No
13	7469-LKBC	Male	0	No	No	16	Yes	No	No	No interne	No interne	No interne	No interne	No interne	No interne	Two year	No	Credit carc	18.95	326.8	No
14	8091-TTVA	Male	0	Yes	No	58	Yes	Yes	Fiber optic	No	No	Yes	No	Yes	Yes	One year	No	Credit carc	100.35	5681.1	No
15	0280-XJGE	Male	0	No	No	49	Yes	Yes	Fiber optic	No	Yes	Yes	No	Yes	Yes	Month-to-	Yes	Bank trans	103.7	5036.3	Yes
16	5129-JLPIS	Male	0	No	No	25	Yes	No	Fiber optic	Yes	No	Yes	Yes	Yes	Yes	Month-to-	Yes	Electronic	105.5	2686.05	No
17	3655-SNQ	Female	0	Yes	Yes	69	Yes	Yes	Fiber optic	Yes	Yes	Yes	Yes	Yes	Yes	Two year	No	Credit carc	113.25	7895.15	No
18	8191-XWS	Female	0	No	No	52	Yes	No	No	No interne	No interne	No interne	No interne	No interne	No interne	One year	No	Mailed che	20.65	1022.95	No
19	9959-WOF	Male	0	No	Yes	71	Yes	Yes	Fiber optic	Yes	No	Yes	No	Yes	Yes	Two year	No	Bank trans	106.7	7382.25	No
20	4190-MFLI	Female	0	Yes	Yes	10	Yes	No	DSL	No	No	Yes	Yes	No	No	Month-to-	No	Credit carc	55.2	528.35	Yes
21	4183-MYF	Female	0	No	No	21	Yes	No	Fiber optic	No	Yes	Yes	No	No	Yes	Month-to-	Yes	Electronic	90.05	1862.9	No
22	8779-QRD	Male	1	No	No	1	No	No phone	DSL	No	No	Yes	No	No	Yes	Month-to-	Yes	Electronic	39.65	39.65	Yes
23	1680-VDC	Male	0	Yes	No	12	Yes	No	No	No interne	No interne	No interne	No interne	No interne	No interne	One year	No	Bank trans	19.8	202.25	No
24	1066-JKSG	Male	0	No	No	1	Yes	No	No	No interne	No interne	No interne	No interne	No interne	No interne	Month-to-	No	Mailed che	20.15	20.15	Yes
25	3638-WEA	Female	0	Yes	No	58	Yes	Yes	DSL	No	Yes	No	Yes	No	No	Two year	Yes	Credit carc	59.9	3505.1	No
26	6322-HRP	Male	0	Yes	Yes	49	Yes	No	DSL	Yes	Yes	No	Yes	No	No	Month-to-	No	Credit carc	59.6	2970.3	No
27	6865-JZNK	Female	0	No	No	30	Yes	No	DSL	Yes	Yes	No	No	No	No	Month-to-	Yes	Bank trans	55.3	1530.6	No
28	6467-CHF	Male	0	Yes	Yes	47	Yes	Yes	Fiber optic	No	Yes	No	No	Yes	Yes	Month-to-	Yes	Electronic	99.35	4749.15	Yes

WA_Fn-UseC_-Telco-Customer-Chur

AIM:

"Predict behaviour to retain customers. You can analyse all relevant customer data and develop focused customer retention programs."

ABOUT DATA:

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

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

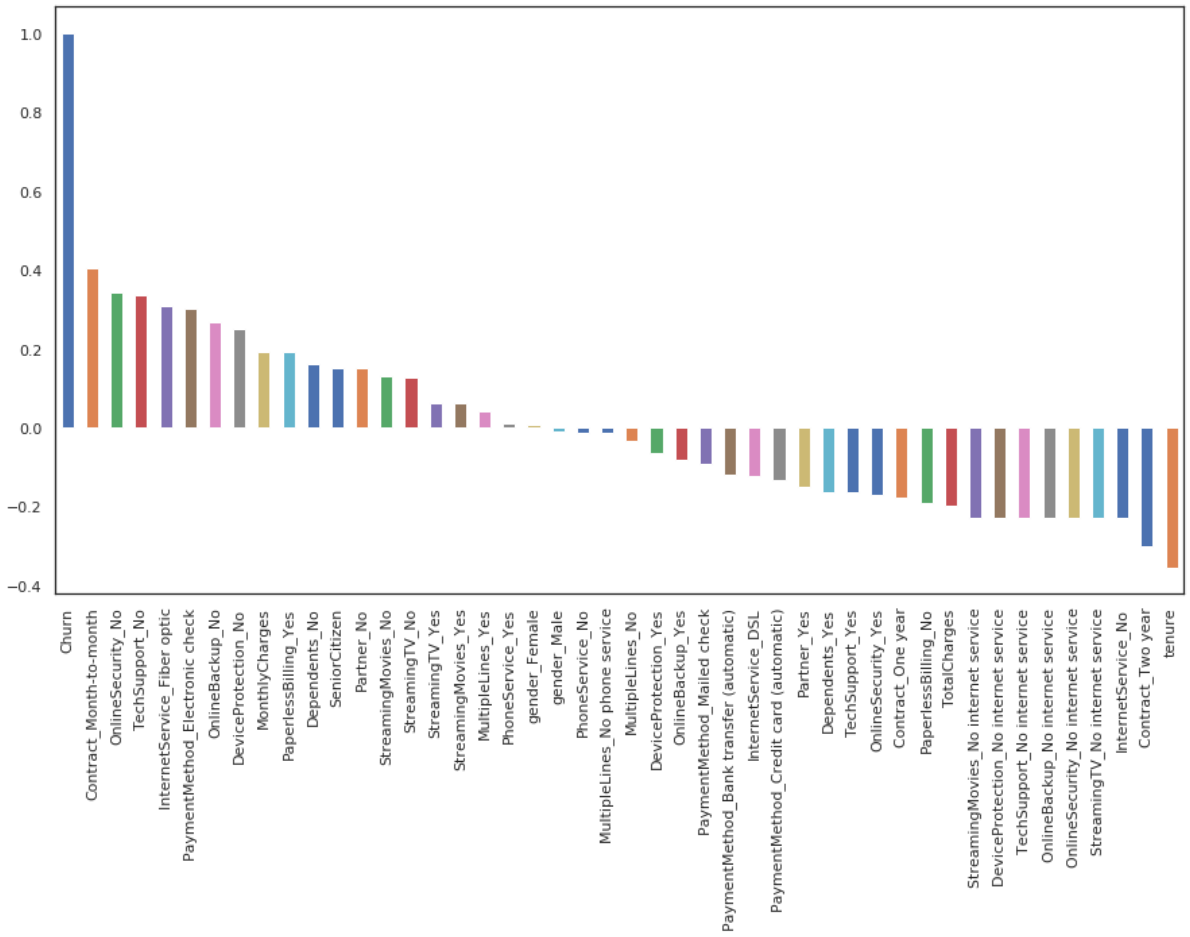
METHODOLOGY:

- Initially we have explored and Visualised various plots on the data to understand the various relations between the predictor and other dependent variables in the dataset.
- To forecast the churn factor, five machine learning techniques were used: logistic regression, XG Boost, Random Forest, SVM, and ADA Boost. From these three models, we discovered that XG Boost's performance is excellent in comparison to other models, increasing the accuracy of the test results.

RESULTS AND FINDINGS:

Initially we performed data cleaning which gives overall productivity and allow for the highest quality information in your decision-making. One of the main issues facing the telecom sector is churn. According to studies, the top 4 wireless carriers in the US have an average monthly churn rate between 1.9% and 2%.

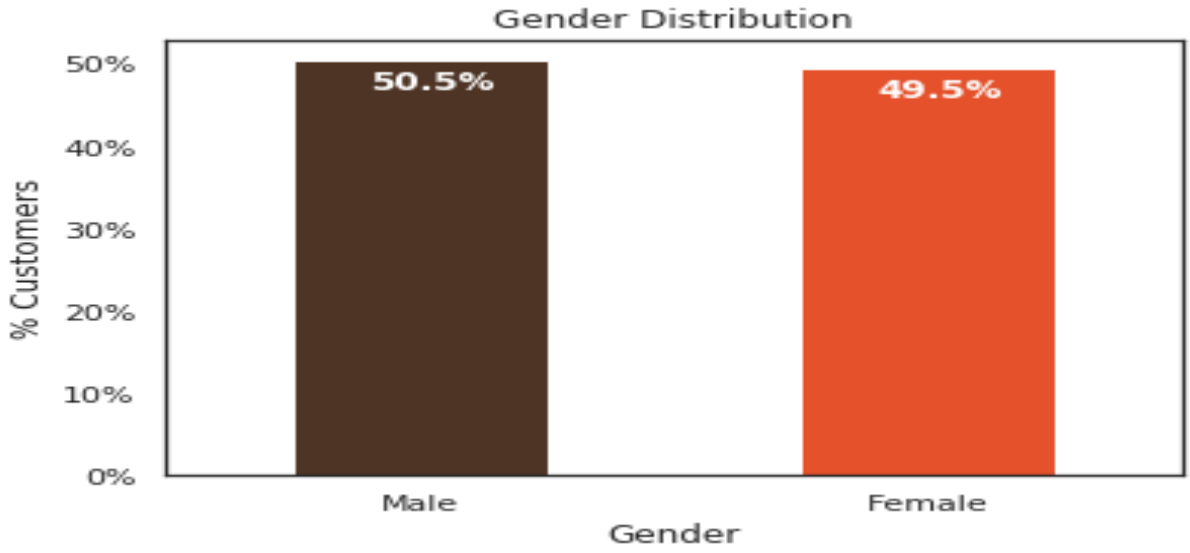
Next we saw the null values in the dataset there are 11 missing values for Total Charges. Let us replace remove these 11 rows from our data set, and found the correlation between predictor variable(churn) and other variables.



Contracts that are month-to-month, a lack of online security, and tech support appear to have a favourable correlation with customer churn. While tenure appears to be positively connected with churn, two-year contracts do not. It's interesting to note that services like tech assistance, streaming TV, online backup, online security, etc. that don't require an internet connection appear to be adversely correlated with turnover. Before we dive into modelling and identifying the key variables, we will first examine the patterns for the aforementioned correlations.

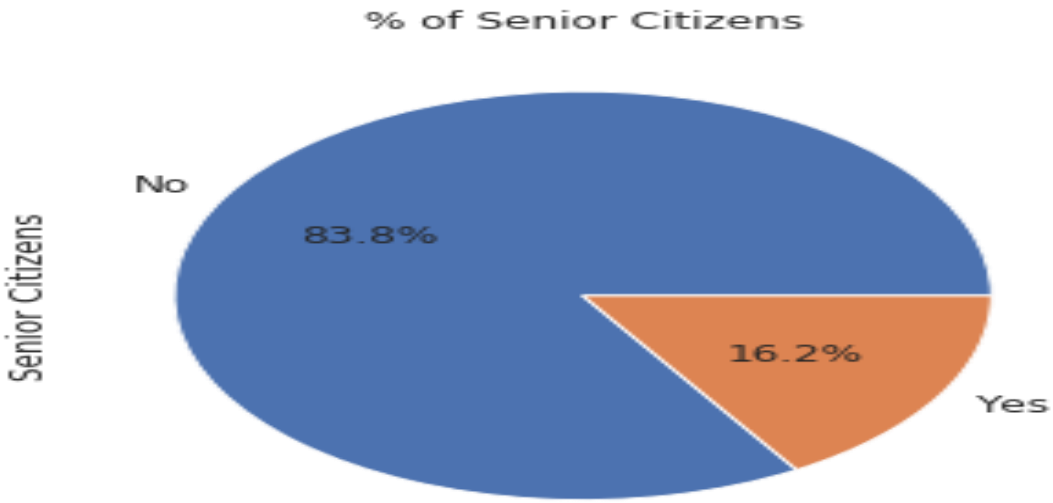
Next, we performed data exploration steps based on the demographics exploration, based on the customer account information like tenure, contract and other services used by the customers.

[1]



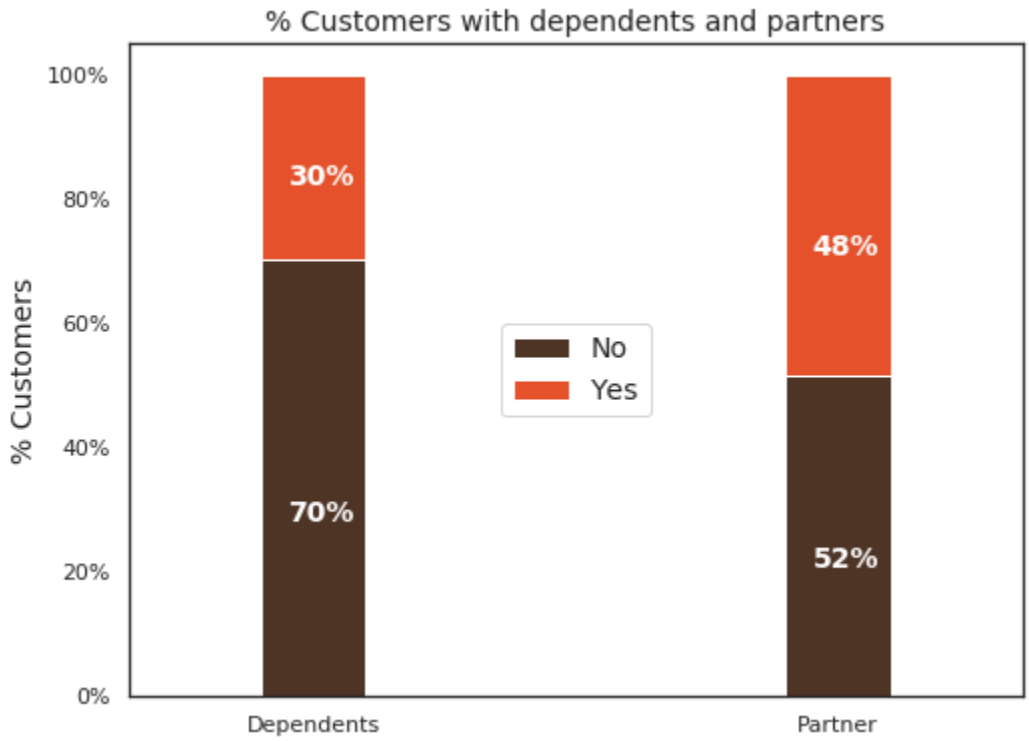
Only 16% of the clients are seniors, according to the statistics. Thus most of our customers in the data are younger people.

[2] Partner and dependent status



About 50% of the customers have a partner, while only 30% of the total customers have dependents.

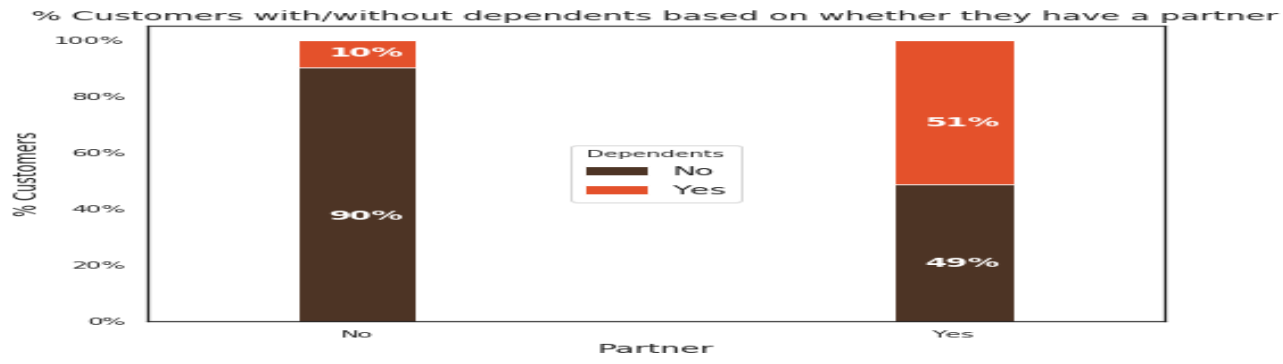
[3]



What would be interesting is to look at the % of customers, who have partners, also have dependents. We will explore this next.

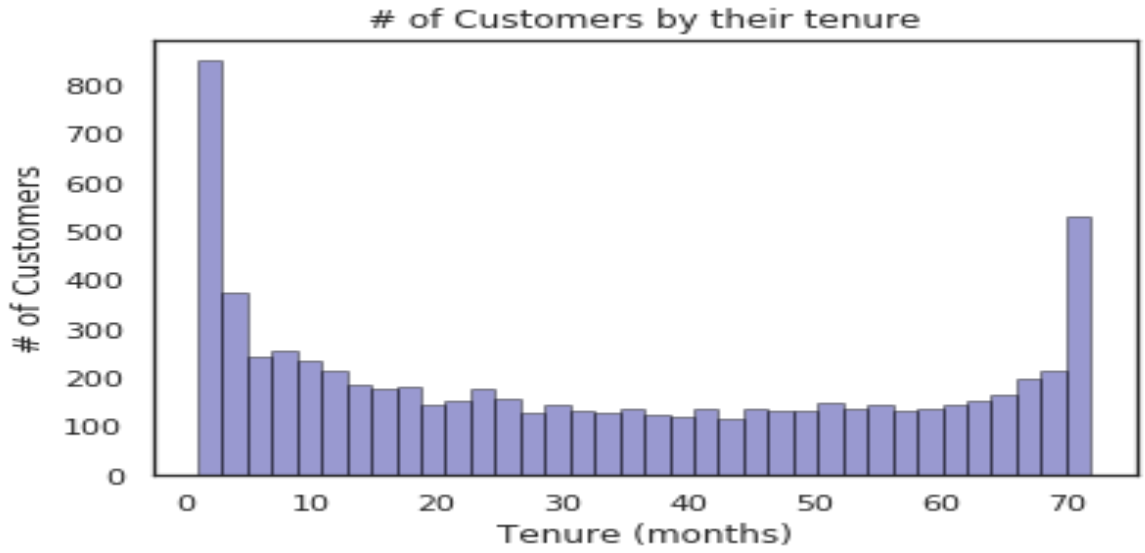
Interestingly, among the customers who have a partner, only about half of them also have a dependent, while other half do not have any independents. Additionally, as expected, among the customers who do not have any partner, a majority (80%) of them do not have any dependents.

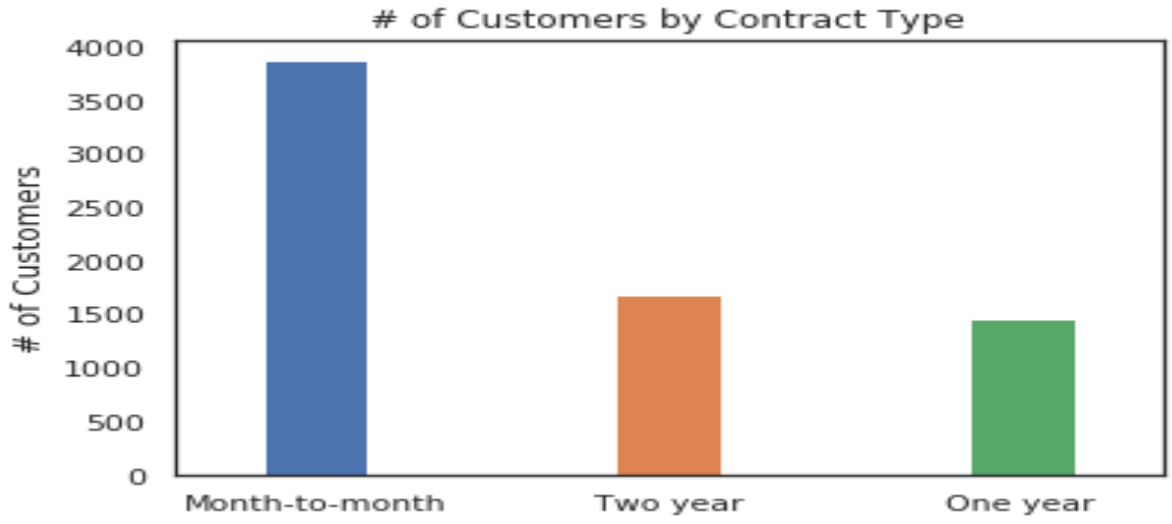
[4]



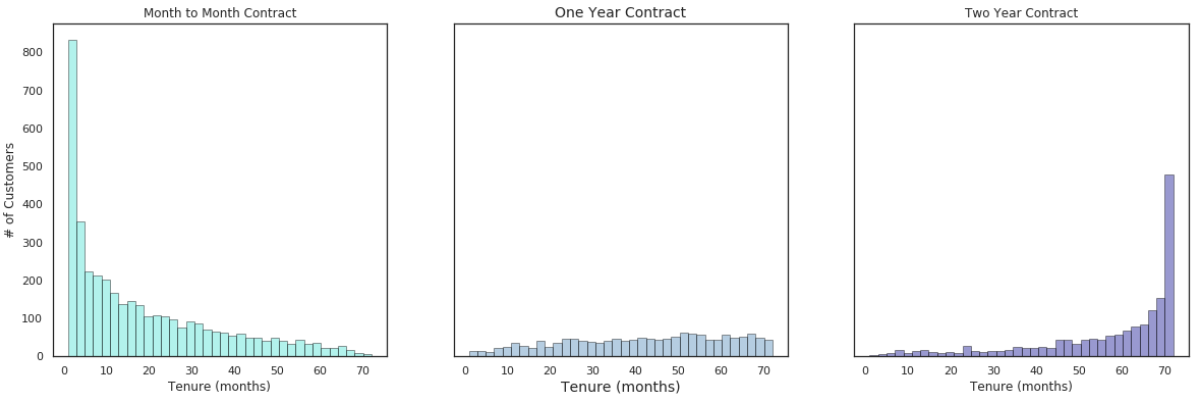
We also examined any gender-based disparities in the percentage of clients having dependents and partners. The distribution of them by gender is the same. Furthermore, there is no distinction in senior citizen status according to gender.

[4]



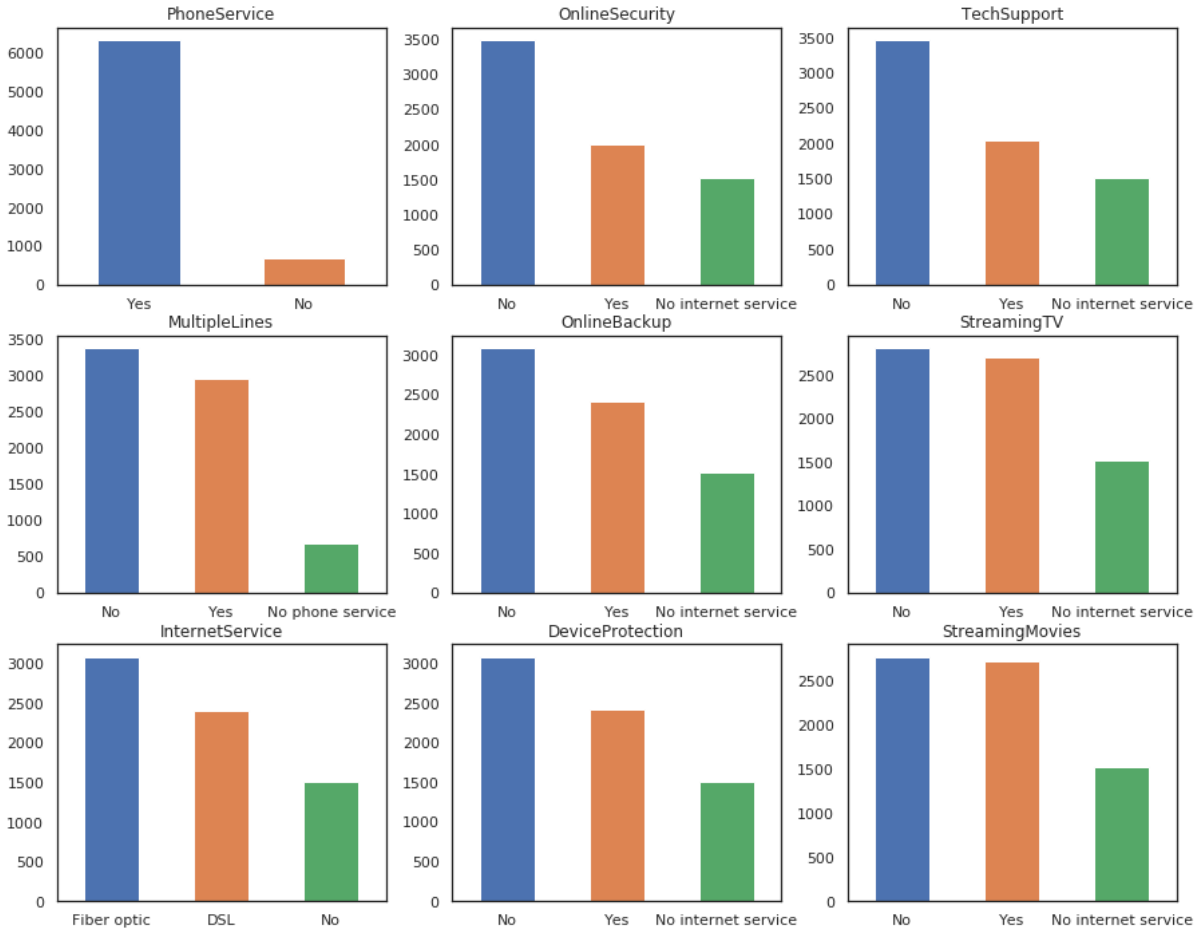


The majority of the consumers are under month-to-month contracts, as this graph shows. The number of customers in the 1 year and 2 year contracts is equal.

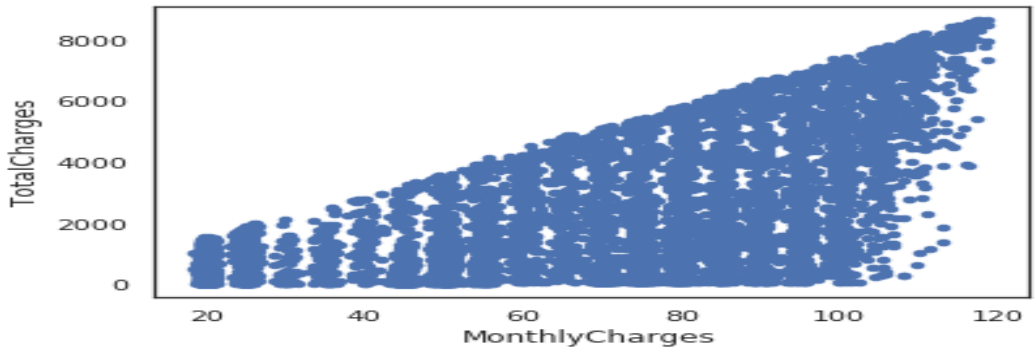


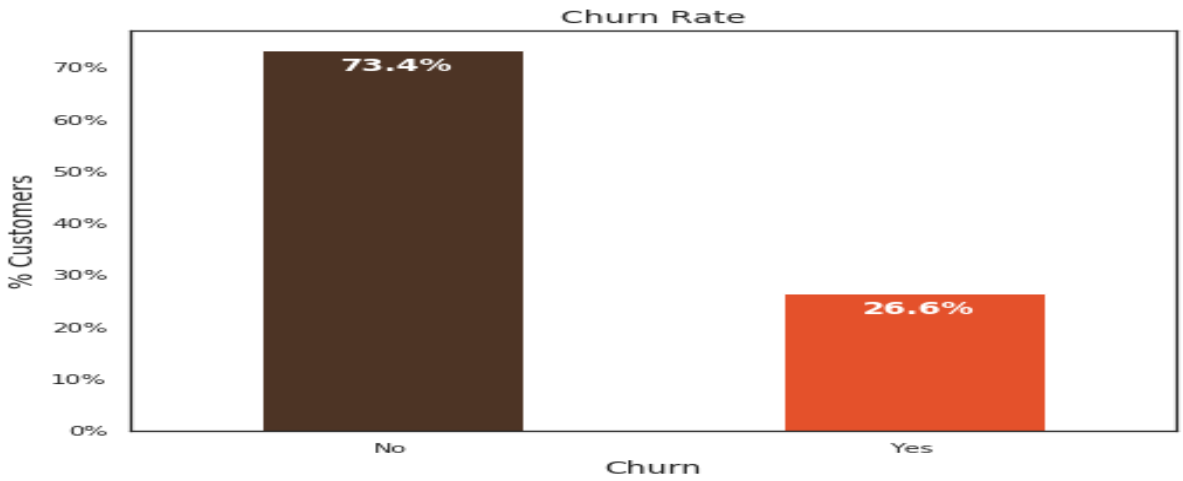
Interestingly, the majority of monthly contracts only last a few weeks to a few months, although two-year contracts often continue for over 70 months. This demonstrates that clients who sign longer contracts are more devoted to the business and have a propensity to stick with it. This is also what the preceding correlation with turnover rate chart showed.

distribution of various services used by customers

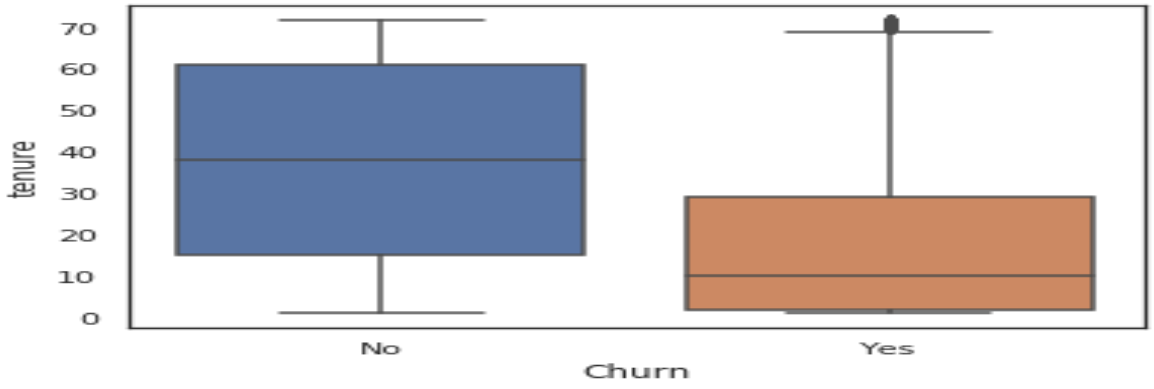


relation between monthly and total charges



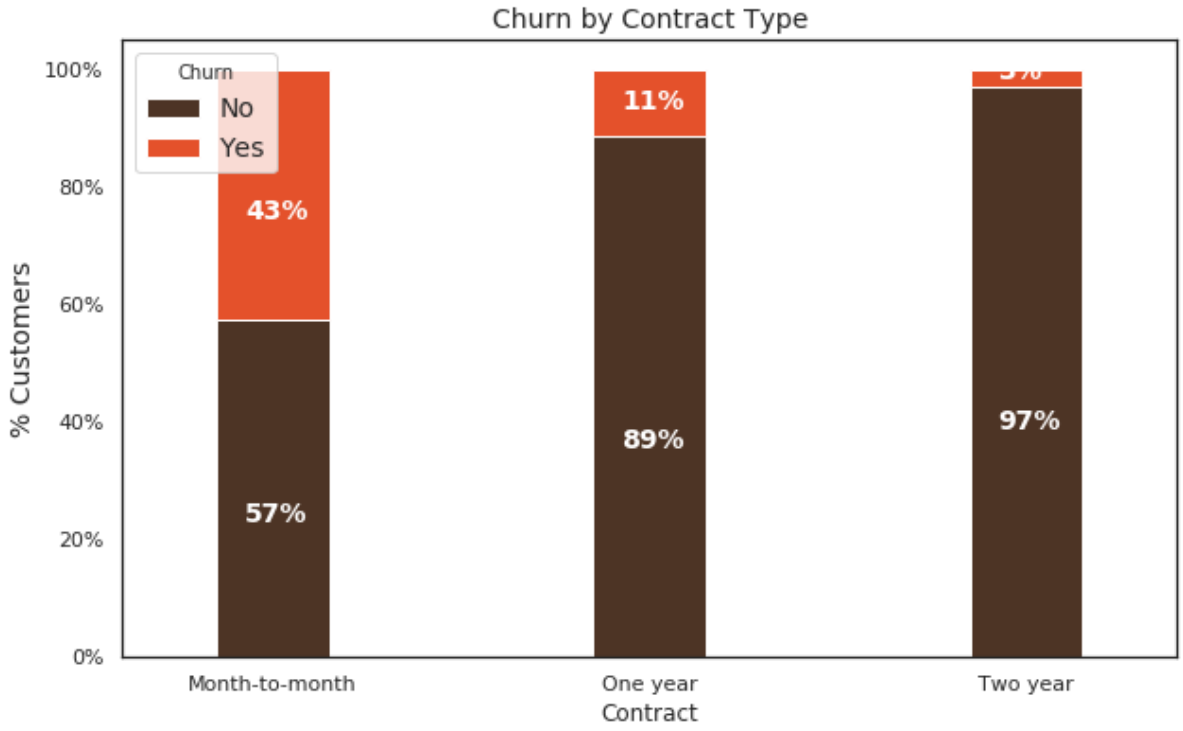


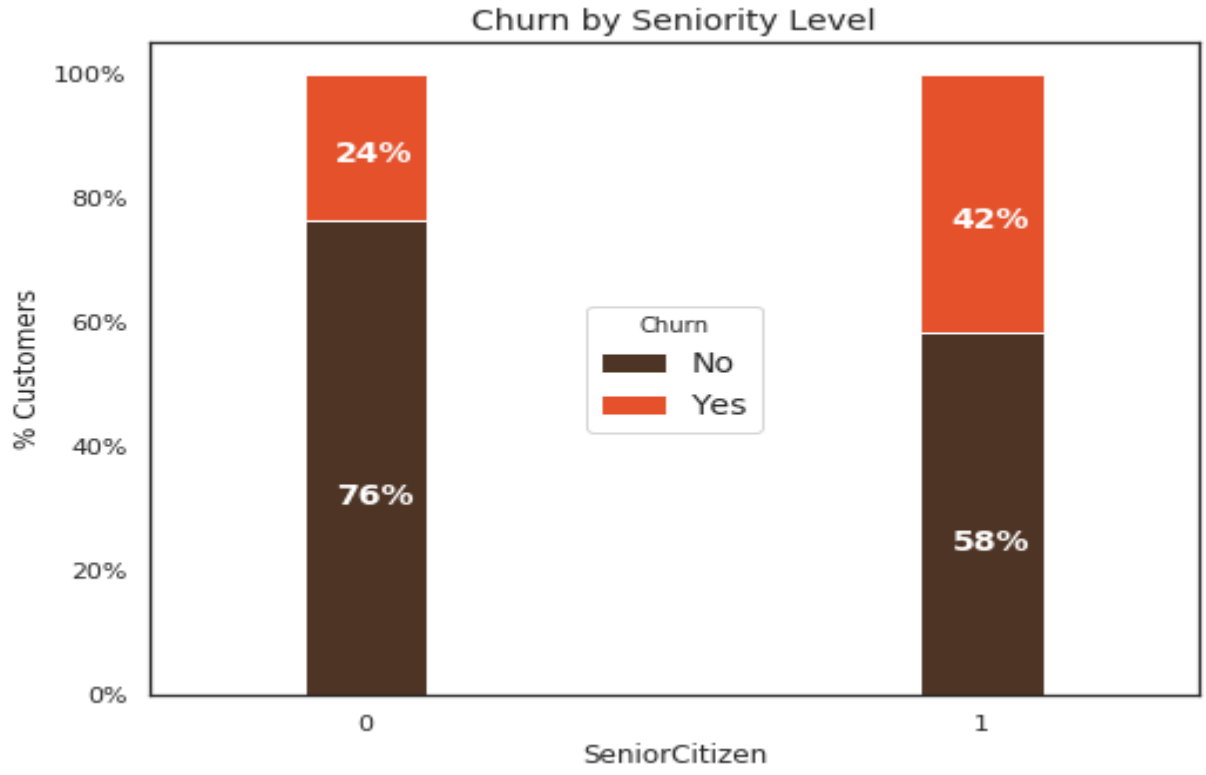
According to our analysis, 74% of clients remain loyal. We would anticipate that a sizable portion of the clients would remain loyal, therefore the data is obviously skewed. This is crucial to remember for our modelling since skewness may result in several false negatives. How to prevent skewness in the data will be covered in the section on modelling.



Churn vs. Tenure: As we can see from the below plot, consumers that do not churn tend to stay with the telecom firm for a longer period of time.

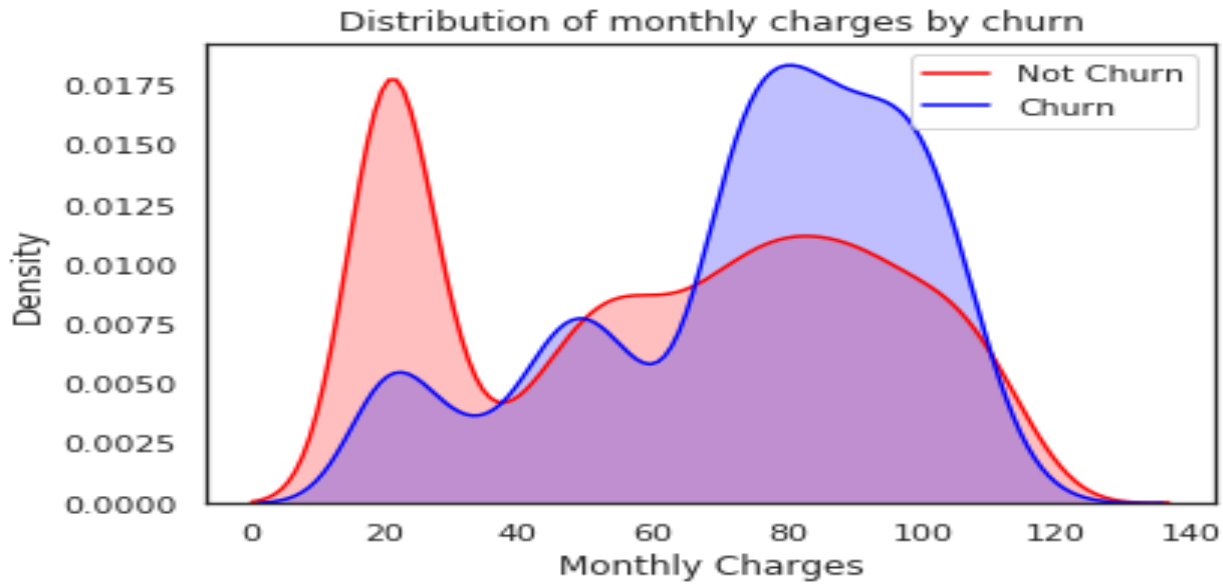
Churn by Contract Type: Similar to what we saw in the correlation plot, the customers who have a month to month contract have a very high churn rate.



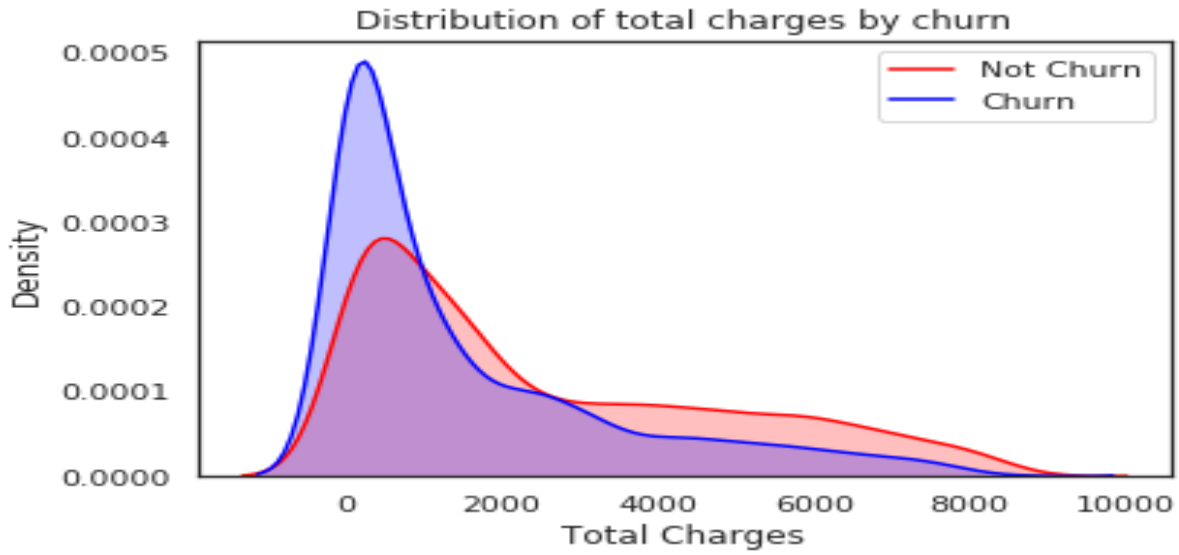


Churn by Seniority: The churn rate for seniors is nearly double that of the general population.

Churn by Monthly Charges: Higher % of customers churn when the monthly charges are high.



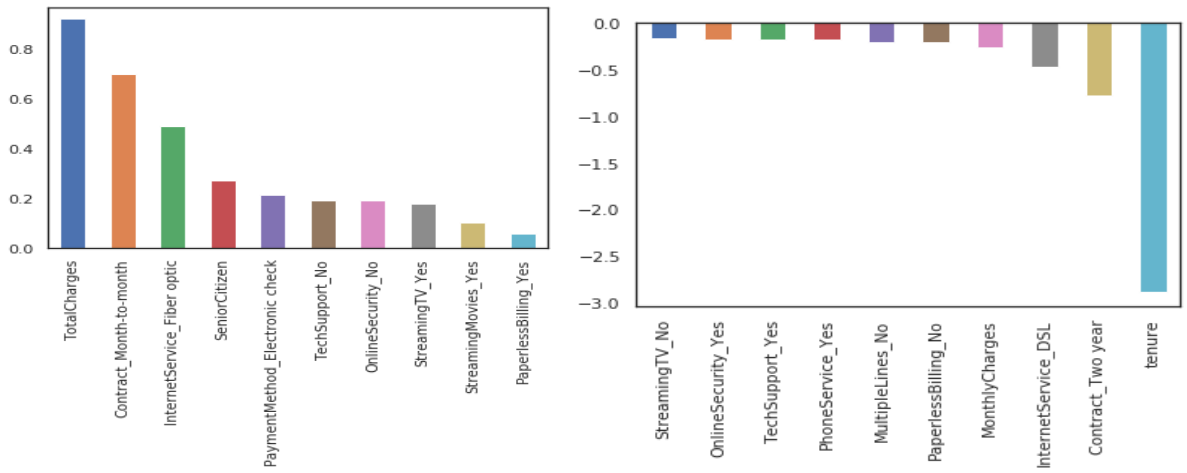
Churn by Total Charges: It seems that there is higer churn when the total charges are lower.



MODELS USED IN OUR PROJECT AND RESPECTIVE RESULTS:

1] LOGISTIC REGRESSION:

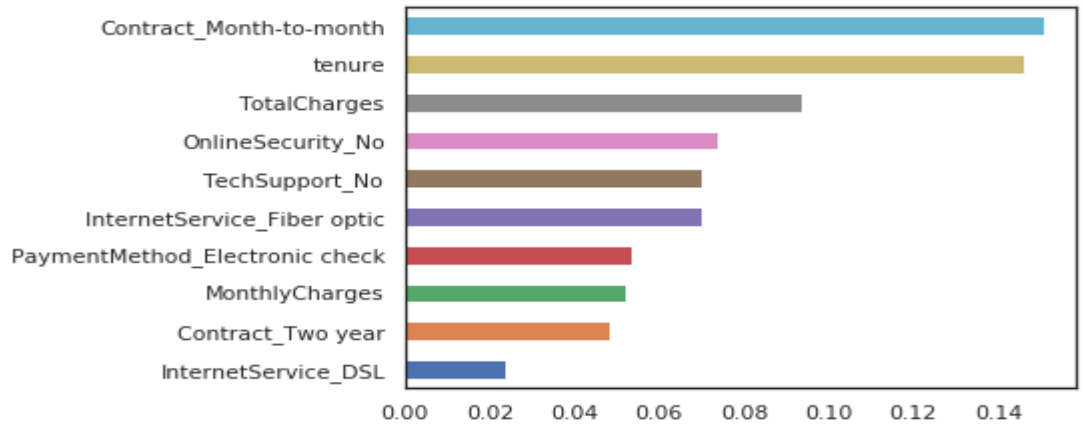
WE HAVEscale the variables in logistic regression so that all of them are within a range of 0 to 1. This helped me improve the accuracy from 79.7% to 80.7%. Further, you will notice below that the importance of variables is also aligned with what we are seeing in Random Forest algorithm and the EDA we conducted above.



As we can see, certain variables are positively correlated with our predicted variable (Churn), while others are negatively correlated. A negative relationship suggests that as that variable increases, the likelihood of churn reduces. Below, we'll list a few of the noteworthy characteristics:

The likelihood of churn is decreased by a two-month contract, as we observed in our EDA. A ccording to the results of logistic regressions, the 2 month contract and tenure had the most a dverse relationships with turnover. Additionally, having DSL internet service lessens the likel ihood of churn. Finally, increased churn rates may be caused by total costs, monthly commit ments, fibre optic internet services, and seniority. The accuracy of logistic is being estimated as **0.80758**

[2] RANDOM FOREST:

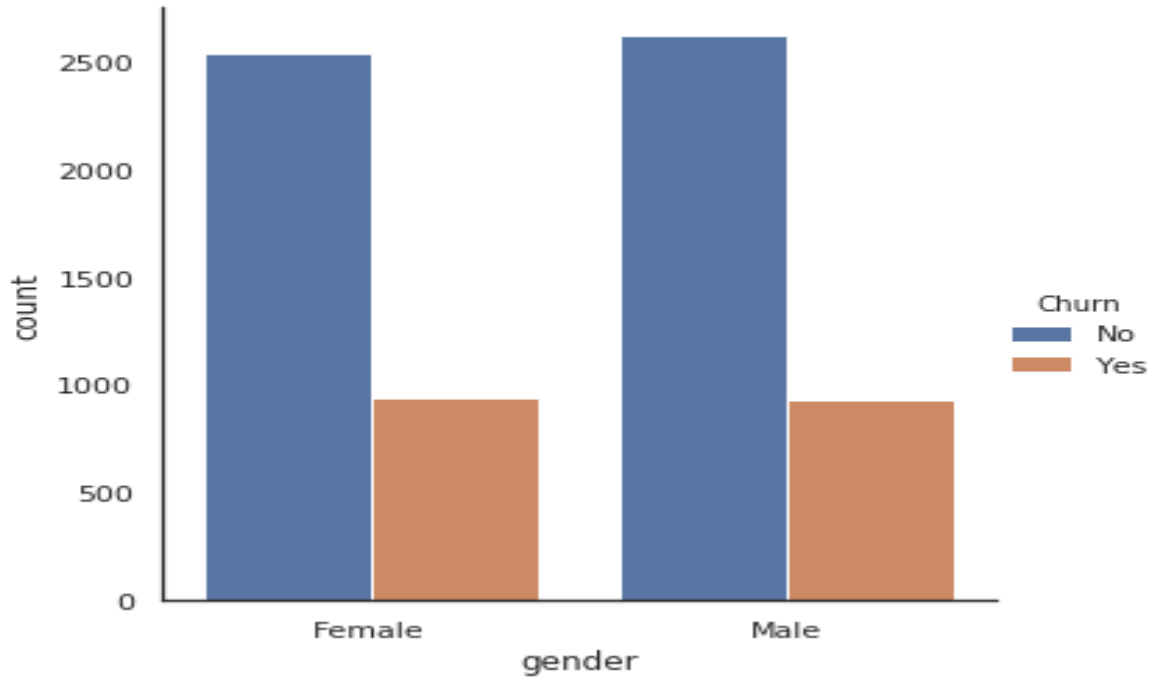


The most significant predictor variables for churn are monthly contract, tenure, and total charges according to the random forest algorithm, and for random forest we got accuracy as 0.8088

The results from random forest are consistent with our expectations from our EDA and fairly similar to those from logistic regression.

[3] **Support vector machine :**

In SVM we were successful in raising the accuracy to 82% by cross validating the accuracy which boost the result, using SVM we got 953 true positive records, 89 false positive records, 164 false negative records and 201 false positive records, which mean 953 records were found to be genuine in the case.



[4] **ADA BOOST:**

```
# AdaBoost Algorithm
from sklearn.ensemble import AdaBoostClassifier
model = AdaBoostClassifier()
# n_estimators = 50 (default value)
# base_estimator = DecisionTreeClassifier (default value)
model.fit(X_train,y_train)
preds = model.predict(X_test)
metrics.accuracy_score(y_test, preds)
```

```
0.8159203980099502
```

[5] **XG BOOST:**

```
from xgboost import XGBClassifier
model = XGBClassifier()
model.fit(X_train, y_train)
preds = model.predict(X_test)
metrics.accuracy_score(y_test, preds)
```

```
0.8294243070362474
```

RESULT:

It's interesting that using XG Boost, we were able to boost test data accuracy to approximately 83%. Without a doubt, XG Boost outperforms all other methods. The slow learning model XG Boost is based on the idea of boosting.

CONCLUSION:

The purpose of this kind of study in the telecom industry is to assist businesses in increasing their profits. It is well known that one of the most significant revenue streams for telecom firms is churn prediction. As a result, the goal of this research was to develop a system that could forecast customer turnover at the telecom business.

Our goal was to identify clients which are likely to churn, so we can do special purpose marketing strategies to avoid the churn event.

- For this we evaluated differently preprocessed datasets and different classifiers. The analysis has shown that the PCA transformation was not found to be useful. Instead, we suggest to use the whole dataset and apply a oversampling technique in order to deal with the unbalanced target variable.
- In the classification chapter we have trained several different classifiers, including a Logistic Regression, a Support-Vector Machine, a ADA BOOST, XG BOOST and a Random Forest.
- It was found that the best performance in accuracy, is achieved by XG boost.

- Concluding, we suggest the Telecom company to use the XG BOOST model to identify potential churn customers and according to the customers life-time value present them special offers and it mainly helps in reducing risk towards the company on behalf of loosing the present customers.

REFERENCE:

[1] AlOmari, D., Hassan, M.M. 2016.PredictingTelecommunication Customer Churn Using Data Mining Techniques. 9th International Conference on Internet and Distributed Computing Systems, 167-178.

[2] Amin, A., Khan, C., Ali, I., Anwar, A. 2014. Customer Churn Prediction in Telecommunication Industry: with and without counter-Example. European Network Intelligence Conference, 134-137

[3] Amin, A., Anwar, S., Adnan, A., Nawaz, M., Alawfi, K., Hussain, A., Huang, K. 2016. Customer Churn Prediction in Telecommunication Sector using Rough Set Approach. Neurocomputing, http://dx.doi.org/10.1016/j. neucom.2016.12.009, 2016:1-21

[4] Argüden Y., Erşahin B. 2008. Veri Madenciliği: Veriden Bilgiye, Masraftan Değere. ARGE Danışmanlık, ISBN: 978-975- 93641-9-9 1. Basım.

[5] Backiel, A., Verbinnen, Y., Baesens, B., Claeskens, G. 2015. Combining Local and Social Network Classifiers to Improve Churn Prediction. International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 651-658.

[6] Brandusoiu, I., Toderean, G. 2013. Churn Prediction In The Telecommunications Sector Using Support Vector Machines. Annals Of The Oradea Un., Fascicle Manag. and Tech. Eng., 1: 19-22

[7] Brandusoiu, I., Toderean, G., Beleiu, H. 2016. Methods for churn prediction in the pre-paid mobile telecommunications industry. International Conference on Communications (COMM), 97-100.

[8] Coussement, K., Lessmann, S., Verstraeten, G. 2016. A comparative analysis of data preparation algorithms for customer churn prediction: A case study in the telecommunication industry. Decis. Supp. Sys., 2016, http:// dx.doi.org/10.1016/j.dss.2016.11.007.

[9] Coussement, K., Lessmann, S., Verstraeten, G. 2016. A comparative analysis of data preparation algorithms for customer churn prediction: A case study in the telecommunication industry. Decis. Supp. Sys., 2016, http:// dx.doi.org/10.1016/j.dss.2016.11.007.

[10] Dalvi, PK., Khandge, SK., Deomore, A. 2016. Analysis of customer churn prediction in telecom industry using decision trees and logistic regression. Symposium on Colossal Data Analysis and Networking (CDAN), DOI: 10.1109/ CDAN.2016.7570883, 1-8

[11] Esteves, G., Mendes-Moreira, J. 2016. Churn Perdiction in the Telecom Business, The eleventh International Conference on Digital Information Management (ICDIM 2016), 254-259.

[12] Forhad, N., Hussain, S., Rahman, RM. 2014. Churn Analysis: Predicting Churners. Ninth International Conference on Digital Information Management (ICDIM), 237-241.

[13] Gok, M., Ozyer, T., Jida, J. 2015. A Case Study for the Churn Prediction in Turksat Internet Service Subscription. IEEE/ ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 1220-1224.

****

In [86]:
```
from sklearn import model_selection
from sklearn.model_selection import cross_validate
from sklearn import tree
from sklearn import svm
from sklearn import ensemble
from sklearn import neighbors
from sklearn import linear_model
from sklearn import metrics
from sklearn import preprocessing
from sklearn.model_selection import StratifiedKFold
```

In [41]:
```
%matplotlib inline

from IPython.display import Image
import matplotlib as mlp
import matplotlib.pyplot as plt
import numpy as np
import os
import pandas as pd
import sklearn
import seaborn as sns
```

In [43]:
```
#df = pd.read_csv('../input/mytest.csv')
df = pd.read_csv('/content/bigml_59c28831336c6604c800002a.csv')

print (df.shape)

#df.dtypes
```

```
(3333, 21)
```

In [44]:
```
# Load data
df.head(3)
```

Out[44]:

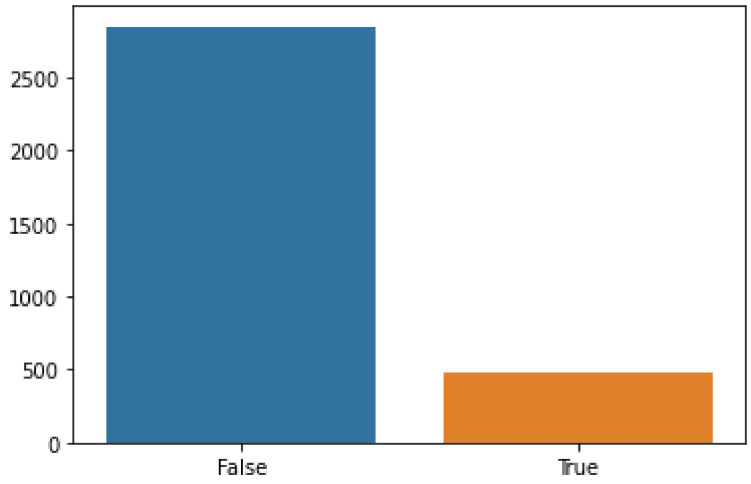
	state	account length	area code	phone number	international plan	voice mail plan	number vmail messages	total day minutes	total day calls	total day charge	...	tota eve calls
0	KS	128	415	382-4657	no	yes	25	265.1	110	45.07	...	99
1	OH	107	415	371-7191	no	yes	26	161.6	123	27.47	...	103
2	NJ	137	415	358-1921	no	no	0	243.4	114	41.38	...	110

3 rows × 21 columns

In [45]:
```
y = df["churn"].value_counts()
#print (y)
sns.barplot(y.index, y.values)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarnin
g: Pass the following variables as keyword args: x, y. From version 0.12, the o
nly valid positional argument will be `data`, and passing other arguments witho
ut an explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

Out[45]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7c000dcd10>
```



In [46]:
```
y_True = df["churn"][df["churn"] == True]
print ("Churn Percentage = "+str( (y_True.shape[0] / df["churn"].shape[0]) * 100
```

```
Churn Percentage = 14.491449144914492
```

Conclusion 1 = Imbalanced data - Lesser datapoints in True Churn category

Descriptive Analysis

In [47]:
```
df.describe()
```

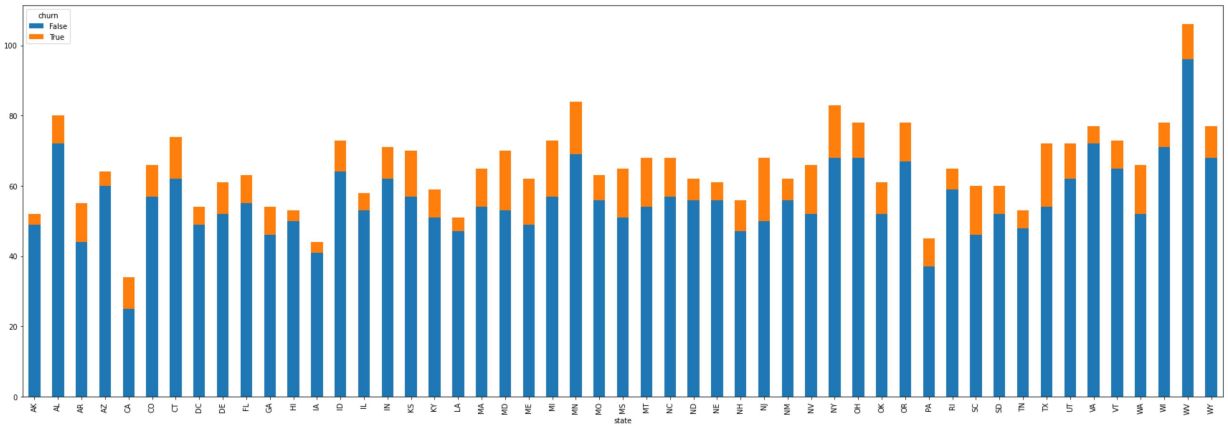
Out[47]:

	account length	area code	number vmail messages	total day minutes	total day calls	total day charge	total eve minutes
count	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000	3333.000000
mean	101.064806	437.182418	8.099010	179.775098	100.435644	30.562307	200.980348
std	39.822106	42.371290	13.688365	54.467389	20.069084	9.259435	50.713844
min	1.000000	408.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	74.000000	408.000000	0.000000	143.700000	87.000000	24.430000	166.600000
50%	101.000000	415.000000	0.000000	179.400000	101.000000	30.500000	201.400000
75%	127.000000	510.000000	20.000000	216.400000	114.000000	36.790000	235.300000
max	243.000000	510.000000	51.000000	350.800000	165.000000	59.640000	363.700000

Churn By State

In [48]:
```
df.groupby(["state", "churn"]).size().unstack().plot(kind='bar', stacked=True, fi
```

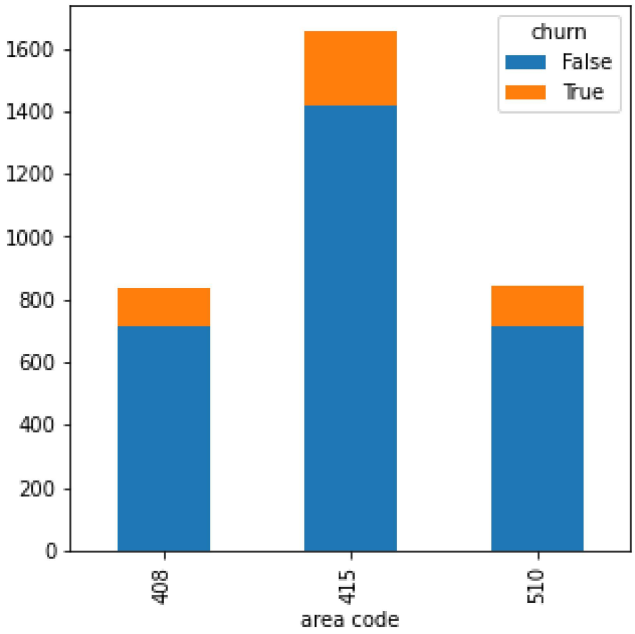
Out[48]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7c000d0450>
```



Churn By Area Code

In [49]:
```
df.groupby(["area code", "churn"]).size().unstack().plot(kind='bar', stacked=True
```

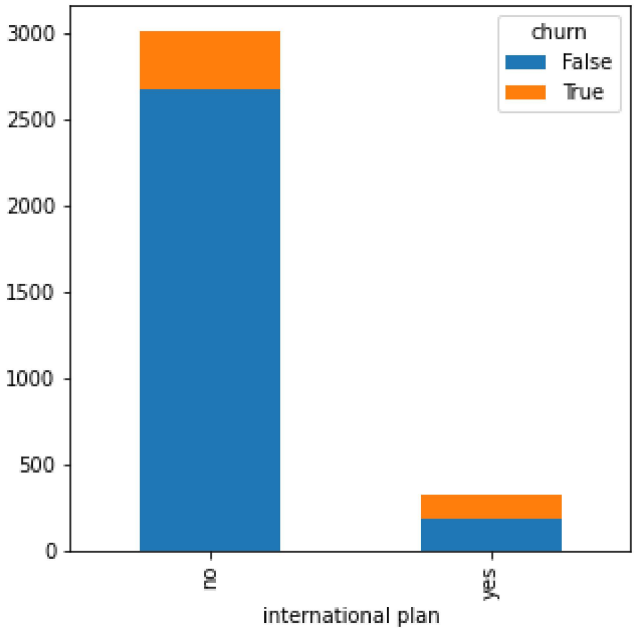
Out[49]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7bffe20850>
```



Churn By Customers with International plan

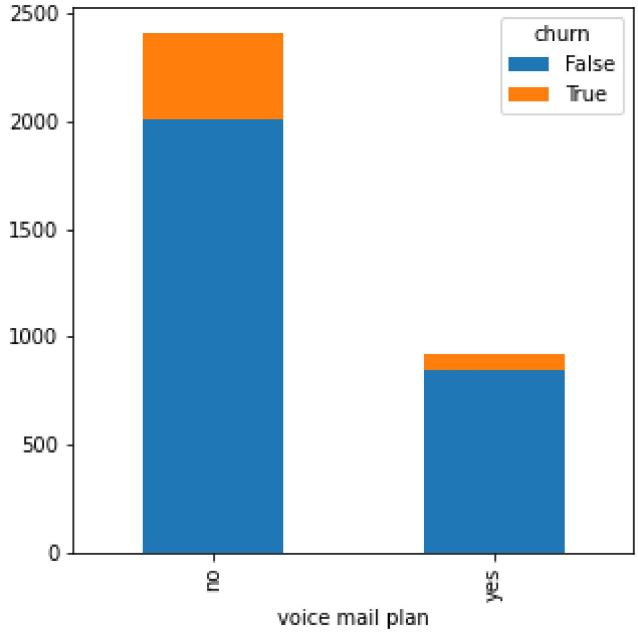
In [50]:
```
df.groupby(["international plan", "churn"]).size().unstack().plot(kind='bar', sta
```

Out[50]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7bffe17810>
```



In [51]:
```
#Churn By Customers with Voice mail plan
df.groupby(["voice mail plan", "churn"]).size().unstack().plot(kind='bar', stacke
```

Out[51]: <matplotlib.axes._subplots.AxesSubplot at 0x7f7bffd6e910>



Handle Categorical Cols - Label Encode

In [52]:
```
# Discreet value integer encoder
label_encoder = preprocessing.LabelEncoder()
```

In [53]:
```
# State is string and we want discreet integer values
df['state'] = label_encoder.fit_transform(df['state'])
df['international plan'] = label_encoder.fit_transform(df['international plan'])
df['voice mail plan'] = label_encoder.fit_transform(df['voice mail plan'])

#print (df['Voice mail plan'][:4])
print (df.dtypes)
```

```
state                     int64
account length            int64
area code                 int64
phone number             object
international plan        int64
voice mail plan           int64
number vmail messages     int64
total day minutes       float64
total day calls           int64
total day charge        float64
total eve minutes       float64
total eve calls           int64
total eve charge        float64
total night minutes     float64
total night calls         int64
total night charge      float64
total intl minutes      float64
total intl calls          int64
total intl charge       float64
customer service calls    int64
churn                      bool
dtype: object
```

In [54]:
```
df.shape
```

Out[54]:
```
(3333, 21)
```

In [55]:
```
df.head()
```

Out[55]:

	state	account length	area code	phone number	international plan	voice mail plan	number vmail messages	total day minutes	total day calls	total day charge	...	tota eve calls
0	16	128	415	382-4657	0	1	25	265.1	110	45.07	...	99
1	35	107	415	371-7191	0	1	26	161.6	123	27.47	...	103
2	31	137	415	358-1921	0	0	0	243.4	114	41.38	...	110
3	35	84	408	375-9999	1	0	0	299.4	71	50.90	...	88
4	36	75	415	330-6626	1	0	0	166.7	113	28.34	...	122

5 rows × 21 columns

Strip of Response value

In [57]:
```
y = df['churn'].to_numpy().astype(np.int)
y.size
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: DeprecationWarn
ing: `np.int` is a deprecated alias for the builtin `int`. To silence this warn
ing, use `int` by itself. Doing this will not modify any behavior and is safe.
When replacing `np.int`, you may wish to use e.g. `np.int64` or `np.int32` to s
pecify the precision. If you wish to review your current use, check the release
note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  """Entry point for launching an IPython kernel.
```

Out[57]:
```
3333
```

Strip off Redundant cols

In [58]:
```
# df = df.drop(["Id","Churn"], axis = 1, inplace=True)
df.drop(["phone number","churn"], axis = 1, inplace=True)
```

In [59]:
```
df.head(3)
```

Out[59]:

	state	account length	area code	international plan	voice mail plan	number vmail messages	total day minutes	total day calls	total day charge	total eve minutes	total eve calls	c
0	16	128	415	0	1	25	265.1	110	45.07	197.4	99	
1	35	107	415	0	1	26	161.6	123	27.47	195.5	103	
2	31	137	415	0	0	0	243.4	114	41.38	121.2	110	

Build Feature Matrix

In [61]:
```
X = df.to_numpy().astype(np.float)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: DeprecationWarn
ing: `np.float` is a deprecated alias for the builtin `float`. To silence this
warning, use `float` by itself. Doing this will not modify any behavior and is
safe. If you specifically wanted the numpy scalar type, use `np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devd
ocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/release/
1.20.0-notes.html#deprecations)
  """Entry point for launching an IPython kernel.
```

In [62]:
```
X
```

Out[62]:
```
array([[ 16.  , 128.  , 415.  , ...,   3.  ,   2.7 ,   1.  ],
       [ 35.  , 107.  , 415.  , ...,   3.  ,   3.7 ,   1.  ],
       [ 31.  , 137.  , 415.  , ...,   5.  ,   3.29,   0.  ],
       ...,
       [ 39.  ,  28.  , 510.  , ...,   6.  ,   3.81,   2.  ],
       [  6.  , 184.  , 510.  , ...,  10.  ,   1.35,   2.  ],
       [ 42.  ,  74.  , 415.  , ...,   4.  ,   3.7 ,   0.  ]])
```

In [63]:
```
X.shape
```

Out[63]:
```
(3333, 19)
```

Standardize Feature Matrix values

In [64]:
```
scaler = preprocessing.StandardScaler()
X = scaler.fit_transform(X)
```

In [65]:
```
X
```

Out[65]:
```
array([[-0.6786493 ,  0.67648946, -0.52360328, ..., -0.60119509,
        -0.0856905 , -0.42793202],
       [ 0.6031696 ,  0.14906505, -0.52360328, ..., -0.60119509,
         1.2411686 , -0.42793202],
       [ 0.33331299,  0.9025285 , -0.52360328, ...,  0.21153386,
         0.69715637, -1.1882185 ],
       ...,
       [ 0.87302621, -1.83505538,  1.71881732, ...,  0.61789834,
         1.3871231 ,  0.33235445],
       [-1.35329082,  2.08295458,  1.71881732, ...,  2.24335625,
        -1.87695028,  0.33235445],
       [ 1.07541867, -0.67974475, -0.52360328, ..., -0.19483061,
         1.2411686 , -1.1882185 ]])
```

Build Models and Train

In [89]:
```
# Create Train & Test Data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_s
```

In [90]:
```
# Running logistic regression model
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
result = model.fit(X_train, y_train)
```

In [91]:
```
from sklearn import metrics
prediction_test = model.predict(X_test)
# Print the prediction accuracy
print (metrics.accuracy_score(y_test, prediction_test))
```

```
0.859
```

In [92]:
```
#RANDOM FOREST
from sklearn.ensemble import RandomForestClassifier
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_s
model_rf = RandomForestClassifier(n_estimators=1000 , oob_score = True, n_jobs =
                                  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))
```

```
0.9250374812593704
```

In [93]:
```
#SUPPORT VECTOR MACHINE
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_s
from sklearn.svm import SVC
model.svm = SVC(kernel='linear')
model.svm.fit(X_train,y_train)
preds = model.svm.predict(X_test)
metrics.accuracy_score(y_test, preds)
```

Out[93]: 0.8860569715142429

In [94]:
```
# Create the Confusion matrix
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,preds))
```

```
[[591   0]
 [ 76   0]]
```

In [97]:
```
# AdaBoost Algorithm
from sklearn.ensemble import AdaBoostClassifier
model = AdaBoostClassifier()
# n_estimators = 50 (default value)
# base_estimator = DecisionTreeClassifier (default value)
model.fit(X_train,y_train)
preds = model.predict(X_test)
metrics.accuracy_score(y_test, preds)
```

Out[97]: 0.8995502248875562

In [98]:
```
#XGBOOST
from xgboost import XGBClassifier
model = XGBClassifier()
model.fit(X_train, y_train)
preds = model.predict(X_test)
metrics.accuracy_score(y_test, preds)
```

Out[98]: 0.967016491754123

with XG Boost I was able to increase the accuracy on test data to almost 96%

Clearly, XG Boost is a winner among all other techniques

XG Boost is a slow learning model and is based on the concept of Boosting

Type *Markdown* and LaTeX: α^2

In [14]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call d
rive.mount("/content/drive", force_remount=True).
```

In [1]:
```
print("ALAGARSAMY N | 19MIA1082")
print("NIRANJAN J | 19MIA1003")
print("VIGNESH N | 19MIA1093")
print("ROSHAN SRINIVAAS | 19MIA1001")
print("T.S.S. ABINANDHAN KUMAR | 19MIA1062")
```

```
ALAGARSAMY N | 19MIA1082
NIRANJAN J | 19MIA1003
VIGNESH N | 19MIA1093
ROSHAN SRINIVAAS | 19MIA1001
T.S.S. ABINANDHAN KUMAR | 19MIA1062
```

In [15]:
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns # For creating plots
import matplotlib.ticker as mtick # For specifying the axes tick format
import matplotlib.pyplot as plt
sns.set(style = 'white')
```

In [58]:
```
file_path = '/content/drive/MyDrive/bigml_59c28831336c6604c800002a.csv'
```

In [22]:
```
#Reading csv file of dataset
telecom_cust = pd.read_csv(file_path2)
telecom_cust.head()
```

Out[22]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	Ir
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	
4	9237-HQITU	Female	0	No	No	2	Yes	No	

5 rows × 21 columns

In [23]:
```
telecom_cust.columns.values
```

Out[23]:
```
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 [24]:
```
# Checking the data types of all the columns
telecom_cust.dtypes
```

Out[24]:
```
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
```

In [25]:
```
# Converting Total Charges to a numerical data type.
telecom_cust.TotalCharges = pd.to_numeric(telecom_cust.TotalCharges, errors='coer
telecom_cust.isnull().sum()
```

Out[25]:
```
customerID          0
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
```

After looking at the above output, we can say that there are 11 missing values for Total Charges. Let us replace remove these 11 rows from our data set

In [26]:
```
#Removing missing values
telecom_cust.dropna(inplace = True)
#Remove customer IDs from the data set
df2 = telecom_cust.iloc[:,1:]
#Convertin the predictor variable in a binary numeric variable
df2['Churn'].replace(to_replace='Yes', value=1, inplace=True)
df2['Churn'].replace(to_replace='No',  value=0, inplace=True)

#Let's convert all the categorical variables into dummy variables
df_dummies = pd.get_dummies(df2)
df_dummies.head()
```

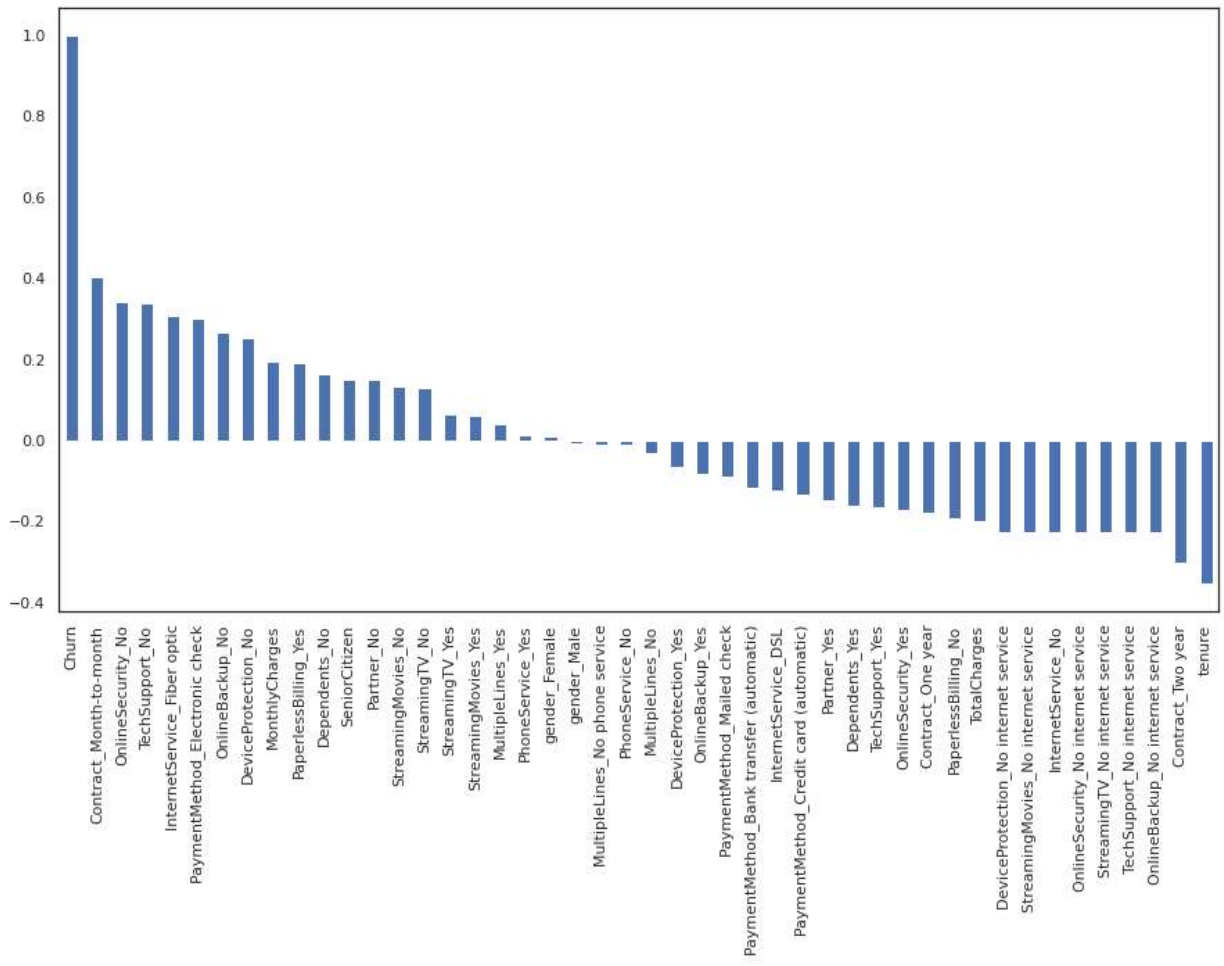
Out[26]:

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges	Churn	gender_Female	gender_Male	Partı
0	0	1	29.85	29.85	0	1	0	
1	0	34	56.95	1889.50	0	0	1	
2	0	2	53.85	108.15	1	0	1	
3	0	45	42.30	1840.75	0	0	1	
4	0	2	70.70	151.65	1	1	0	

5 rows × 46 columns

In [27]:
```
#Get Correlation of "Churn" with other variables:
plt.figure(figsize=(15,8))
df_dummies.corr()['Churn'].sort_values(ascending = False).plot(kind='bar')
```

Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x7f6a95fc4c10>



Month to month contracts, absence of online security and tech support seem to be positively correlated with churn. While, tenure, two year contracts seem to be negatively correlated with churn.

Interestingly, services such as Online security, streaming TV, online backup, tech support, etc. without internet connection seem to be negatively related to churn

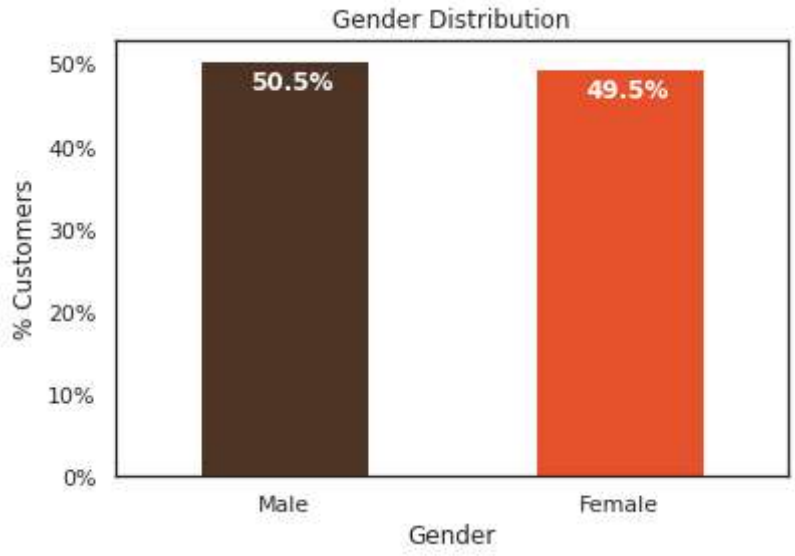
Data Exploration

Let us first start with exploring our data set, to better understand the patterns in the data and potentially form some hypothesis. First we will look at the distribution of individual variables and then slice and dice our data for any interesting trends.

A.) Demographics - Let us first understand the gender, age range, patner and dependent status of the customers

Gender Distribution - About half of the customers in our data set are male while the other half are female

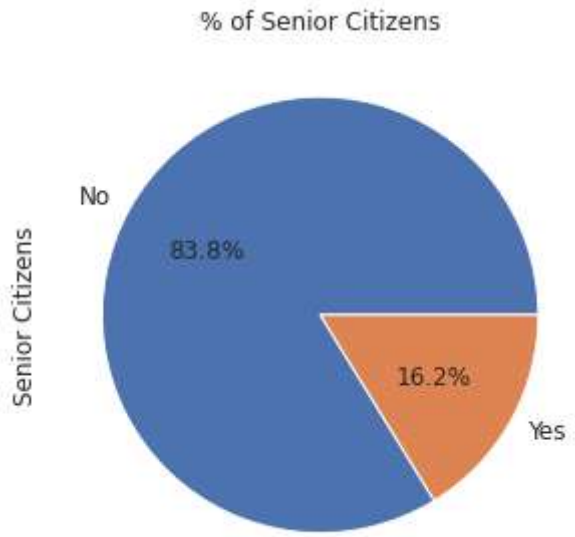
In [28]:
```
colors = ['#4D3425','#E4512B']
ax = (telecom_cust['gender'].value_counts()*100.0 /len(telecom_cust)).plot(kind='
                                                                            stacke
                                                                           rot = 0
                                                                           color =

ax.yaxis.set_major_formatter(mtick.PercentFormatter())
ax.set_ylabel('% Customers')
ax.set_xlabel('Gender')
ax.set_ylabel('% Customers')
ax.set_title('Gender Distribution')

# create a list to collect the plt.patches data
totals = []

# find the values and append to list
for i in ax.patches:
    totals.append(i.get_width())

# set individual bar lables using above list
total = sum(totals)

for i in ax.patches:
    # get_width pulls left or right; get_y pushes up or down
    ax.text(i.get_x()+.15, i.get_height()-3.5, \
            str(round((i.get_height()/total), 1))+'%',
            fontsize=12,
            color='white',
           weight = 'bold')
```



% Senior Citizens - There are only 16% of the customers who are senior citizens. Thus most of our customers in the data are younger people.

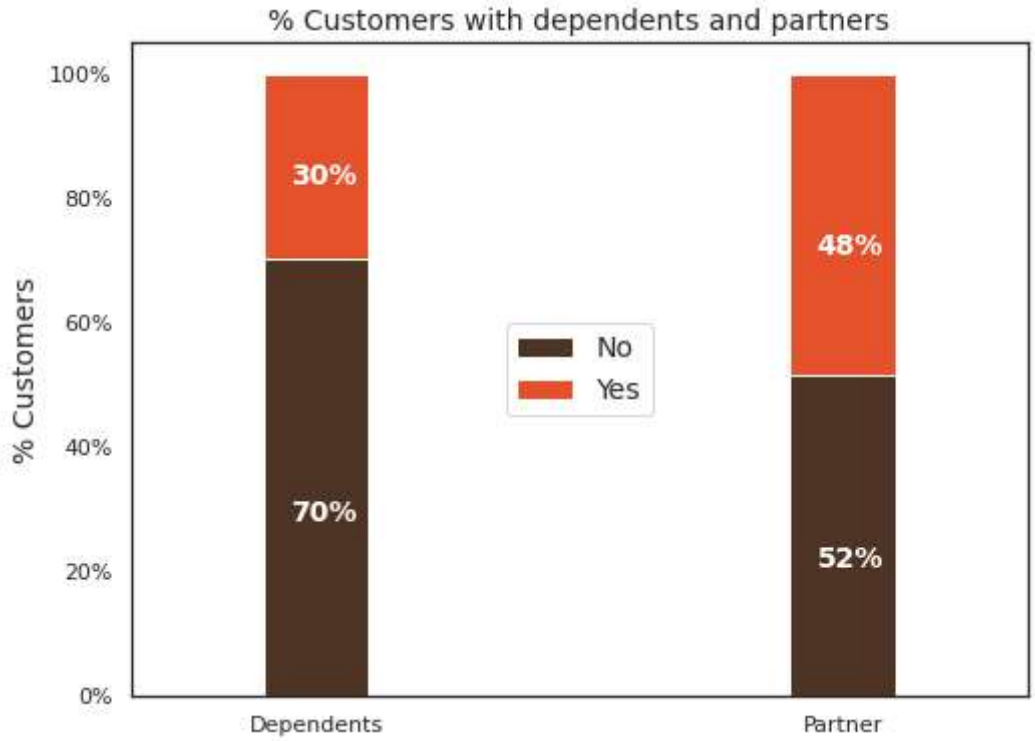
In [29]:
```
ax = (telecom_cust['SeniorCitizen'].value_counts()*100.0 /len(telecom_cust))\
.plot.pie(autopct='%.1f%%', labels = ['No', 'Yes'],figsize =(5,5), fontsize = 12
ax.yaxis.set_major_formatter(mtick.PercentFormatter())
ax.set_ylabel('Senior Citizens',fontsize = 12)
ax.set_title('% of Senior Citizens', fontsize = 12)
```

Out[29]: Text(0.5, 1.0, '% of Senior Citizens')



Partner and dependent status - About 50% of the customers have a partner, while only 30% of the total customers have dependents.

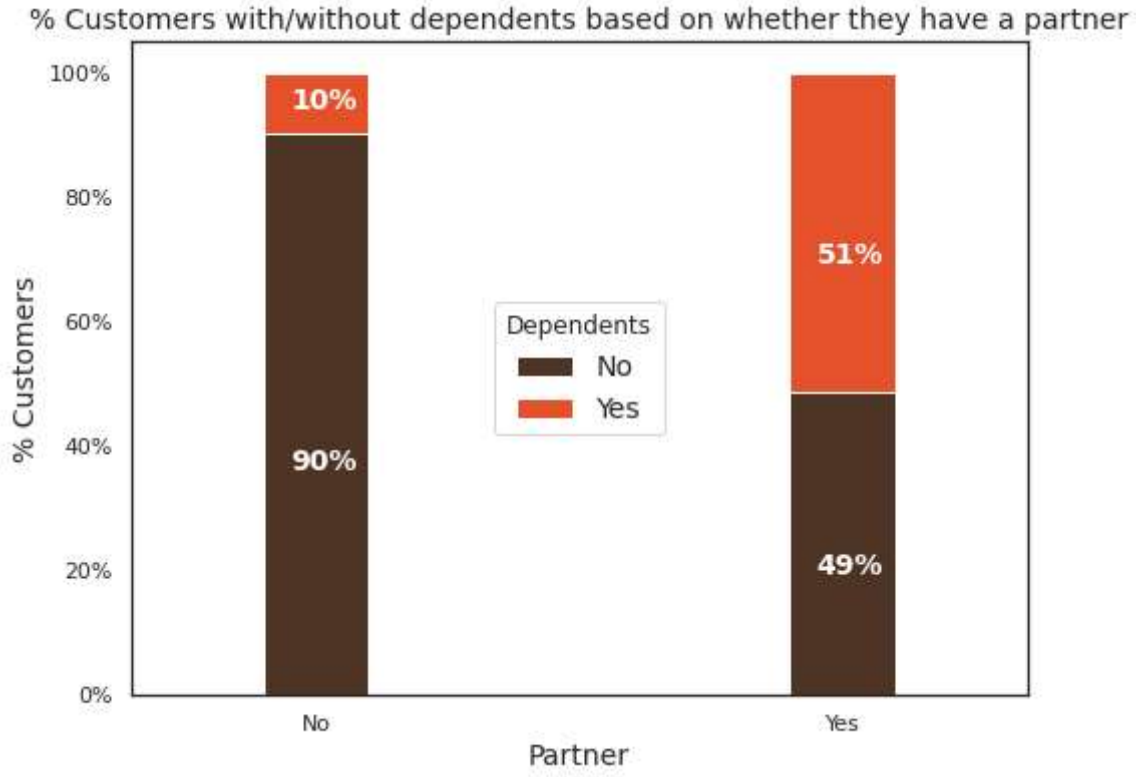
In [30]:
```
df2 = pd.melt(telecom_cust, id_vars=['customerID'], value_vars=['Dependents','Par
df3 = df2.groupby(['variable','value']).count().unstack()
df3 = df3*100/len(telecom_cust)
colors = ['#4D3425','#E4512B']
ax = df3.loc[:,'customerID'].plot.bar(stacked=True, color=colors,
                                      figsize=(8,6),rot = 0,
                                     width = 0.2)

ax.yaxis.set_major_formatter(mtick.PercentFormatter())
ax.set_ylabel('% Customers',size = 14)
ax.set_xlabel('')
ax.set_title('% Customers with dependents and partners',size = 14)
ax.legend(loc = 'center',prop={'size':14})

for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    x, y = p.get_xy()
    ax.annotate('{:.0f}%'.format(height), (p.get_x()+.25*width, p.get_y()+.4*heig
                color = 'white',
               weight = 'bold',
               size = 14)
```



What would be interesting is to look at the % of customers, who have partners, also have dependents. We will explore this next.

Interestingly, among the customers who have a partner, only about half of them also have a dependent, while other half do not have any independents. Additionally, as expected, among the customers who do not have any partner, a majority (80%) of them do not have any dependents.

In [31]:
```
colors = ['#4D3425','#E4512B']
partner_dependents = telecom_cust.groupby(['Partner','Dependents']).size().unstac

ax = (partner_dependents.T*100.0 / partner_dependents.T.sum()).T.plot(kind='bar',
                                                                      width = 0.2,
                                                                      stacked = True,
                                                                      rot = 0,
                                                                      figsize = (8,6),
                                                                      color = colors)
ax.yaxis.set_major_formatter(mtick.PercentFormatter())
ax.legend(loc='center',prop={'size':14},title = 'Dependents',fontsize =14)
ax.set_ylabel('% Customers',size = 14)
ax.set_title('% Customers with/without dependents based on whether they have a pa
ax.xaxis.label.set_size(14)

# Code to add the data labels on the stacked bar chart
for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    x, y = p.get_xy()
    ax.annotate('{:.0f}%'.format(height), (p.get_x()+.25*width, p.get_y()+.4*heig
                color = 'white',
               weight = 'bold',
               size = 14)
```



I also looked at any differences between the % of customers with/without dependents and partners by gender. There is no difference in their distribution by gender. Additionally, there is no difference in senior citizen status by gender.

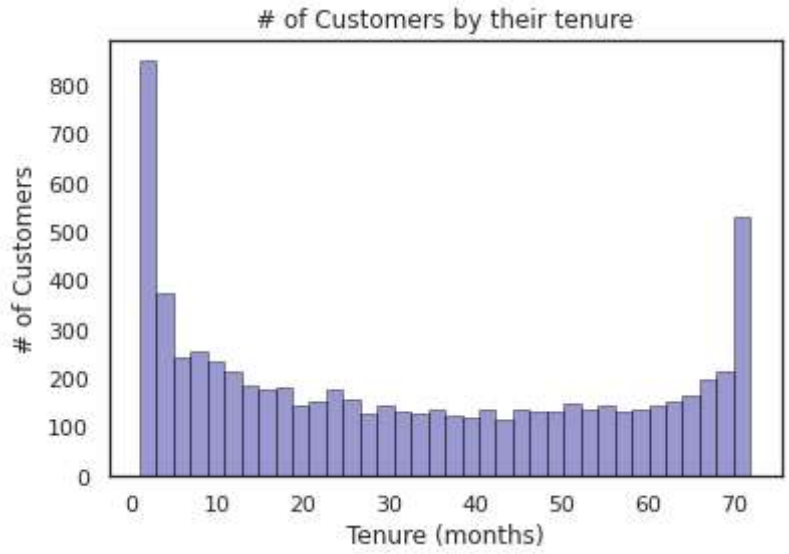
Customer Account Information: Let u now look at the tenure, contract

1. Tenure: After looking at the below histogram we can see that a lot of customers have been with the telecom company for just a month, while quite a many are there for about 72 months. This could be potentially because different customers have different contracts. Thus based on the contract they are into it could be more/less easier for the customers to stay/leave the telecom company.

In [32]:
```
ax = sns.distplot(telecom_cust['tenure'], hist=True, kde=False,
             bins=int(180/5), color = 'darkblue',
             hist_kws={'edgecolor':'black'},
             kde_kws={'linewidth': 4})
ax.set_ylabel('# of Customers')
ax.set_xlabel('Tenure (months)')
ax.set_title('# of Customers by their tenure')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWar
ning: `distplot` is a deprecated function and will be removed in a future versi
on. Please adapt your code to use either `displot` (a figure-level function wit
h similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

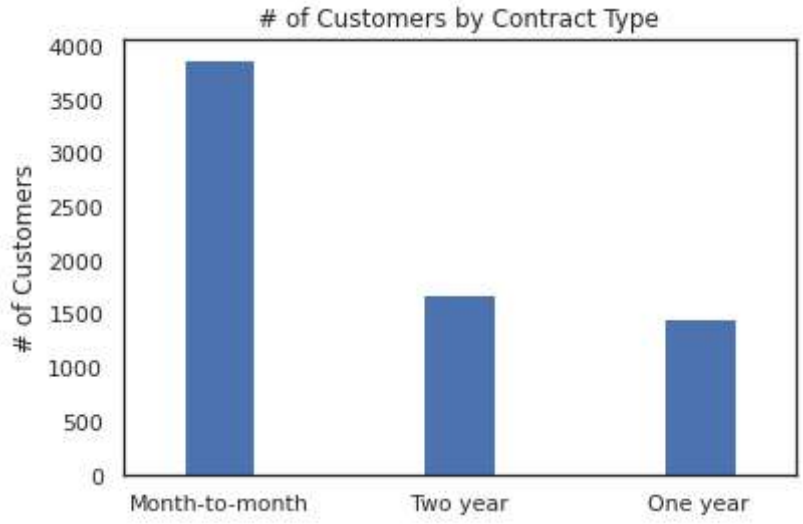
Out[32]:
```
Text(0.5, 1.0, '# of Customers by their tenure')
```



2. Contracts: To understand the above graph, lets first look at the # of customers by different contracts.

In [33]:
```
ax = telecom_cust['Contract'].value_counts().plot(kind = 'bar',rot = 0, width = 0
ax.set_ylabel('# of Customers')
ax.set_title('# of Customers by Contract Type')
```

Out[33]:
```
Text(0.5, 1.0, '# of Customers by Contract Type')
```

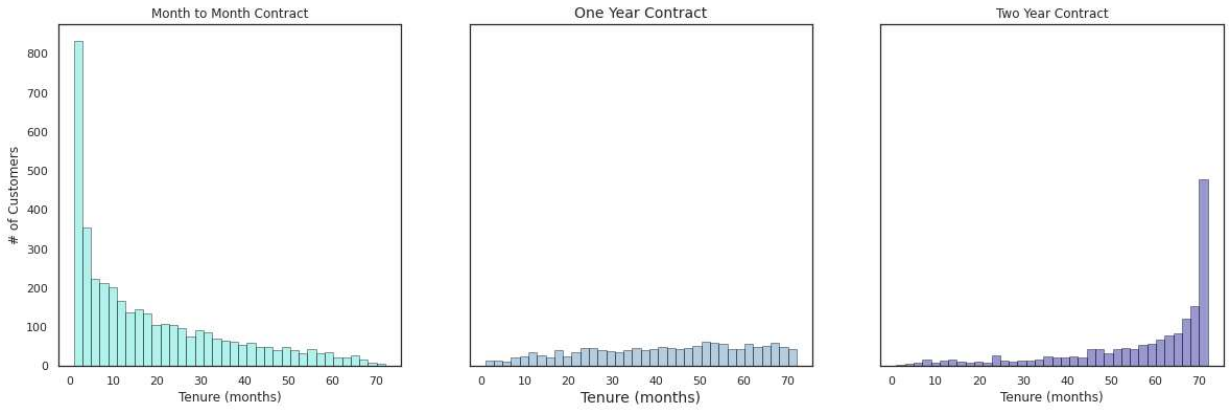


As we can see from this graph most of the customers are in the month to month contract. While there are equal number of customers in the 1 year and 2 year contracts.

Below we will understand the tenure of customers based on their contract type.

In [34]:
```
fig, (ax1,ax2,ax3) = plt.subplots(nrows=1, ncols=3, sharey = True, figsize = (20,

ax = sns.distplot(telecom_cust[telecom_cust['Contract']=='Month-to-month']['tenur
                  hist=True, kde=False,
                  bins=int(180/5), color = 'turquoise',
                  hist_kws={'edgecolor':'black'},
                  kde_kws={'linewidth': 4},
                 ax=ax1)
ax.set_ylabel('# of Customers')
ax.set_xlabel('Tenure (months)')
ax.set_title('Month to Month Contract')

ax = sns.distplot(telecom_cust[telecom_cust['Contract']=='One year']['tenure'],
                  hist=True, kde=False,
                  bins=int(180/5), color = 'steelblue',
                  hist_kws={'edgecolor':'black'},
                  kde_kws={'linewidth': 4},
                 ax=ax2)
ax.set_xlabel('Tenure (months)',size = 14)
ax.set_title('One Year Contract',size = 14)

ax = sns.distplot(telecom_cust[telecom_cust['Contract']=='Two year']['tenure'],
                  hist=True, kde=False,
                  bins=int(180/5), color = 'darkblue',
                  hist_kws={'edgecolor':'black'},
                  kde_kws={'linewidth': 4},
                 ax=ax3)

ax.set_xlabel('Tenure (months)')
ax.set_title('Two Year Contract')
```

Out[34]: Text(0.5, 1.0, 'Two Year Contract')



Interestingly most of the monthly contracts last for 1-2 months, while the 2 year contracts tend to last for about 70 months. This shows that the customers taking a longer contract are more loyal to the company and tend to stay with it for a longer period of time.

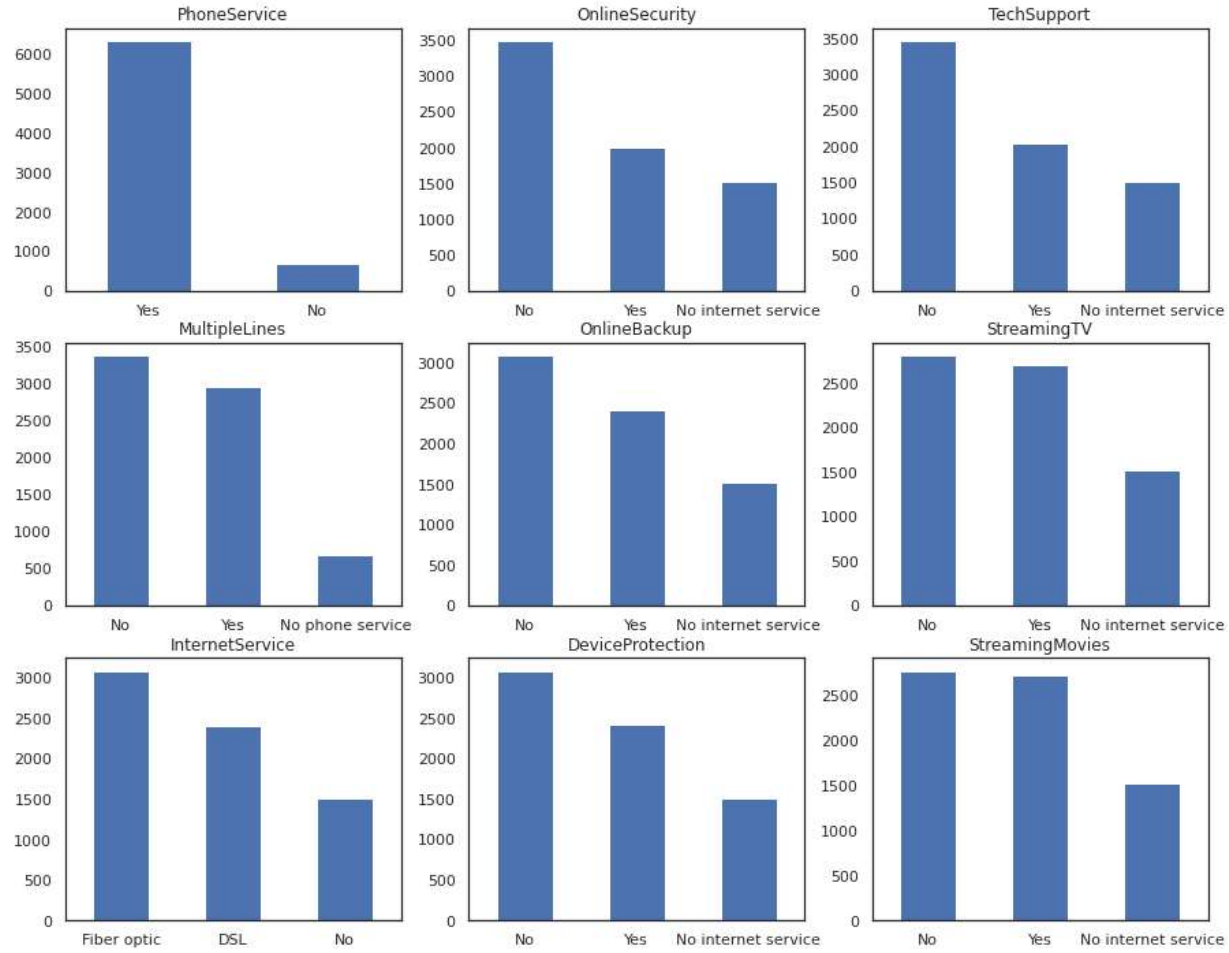
This is also what we saw in the earlier chart on correlation with the churn rate.

C. Let us now look at the distribution of various services used by customers

In [35]:
```
telecom_cust.columns.values
```

Out[35]:
```
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 [36]:
```
services = ['PhoneService','MultipleLines','InternetService','OnlineSecurity',
            'OnlineBackup','DeviceProtection','TechSupport','StreamingTV','Streami

fig, axes = plt.subplots(nrows = 3,ncols = 3,figsize = (15,12))
for i, item in enumerate(services):
    if i < 3:
        ax = telecom_cust[item].value_counts().plot(kind = 'bar',ax=axes[i,0],rot

    elif i >=3 and i < 6:
        ax = telecom_cust[item].value_counts().plot(kind = 'bar',ax=axes[i-3,1],r

    elif i < 9:
        ax = telecom_cust[item].value_counts().plot(kind = 'bar',ax=axes[i-6,2],r
    ax.set_title(item)
```

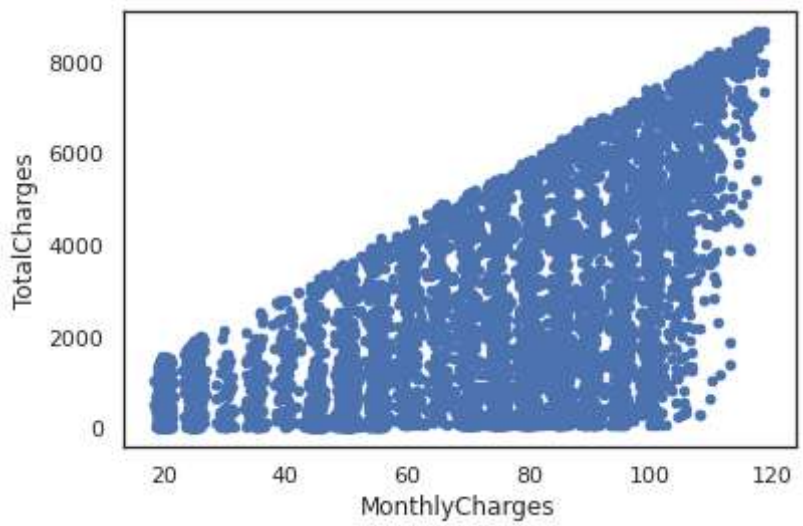


D.) Now let's take a quick look at the relation between monthly and total charges We will observe that the total charges increases as the monthly bill for a customer increases.

In [37]:
```
telecom_cust[['MonthlyCharges', 'TotalCharges']].plot.scatter(x = 'MonthlyCharges
                                                              y='TotalCharges')
```

```
WARNING:matplotlib.axes._axes:*c* argument looks like a single numeric RGB or R
GBA sequence, which should be avoided as value-mapping will have precedence in
case its length matches with *x* & *y*.  Please use the *color* keyword-argumen
t or provide a 2-D array with a single row if you intend to specify the same RG
B or RGBA value for all points.
```

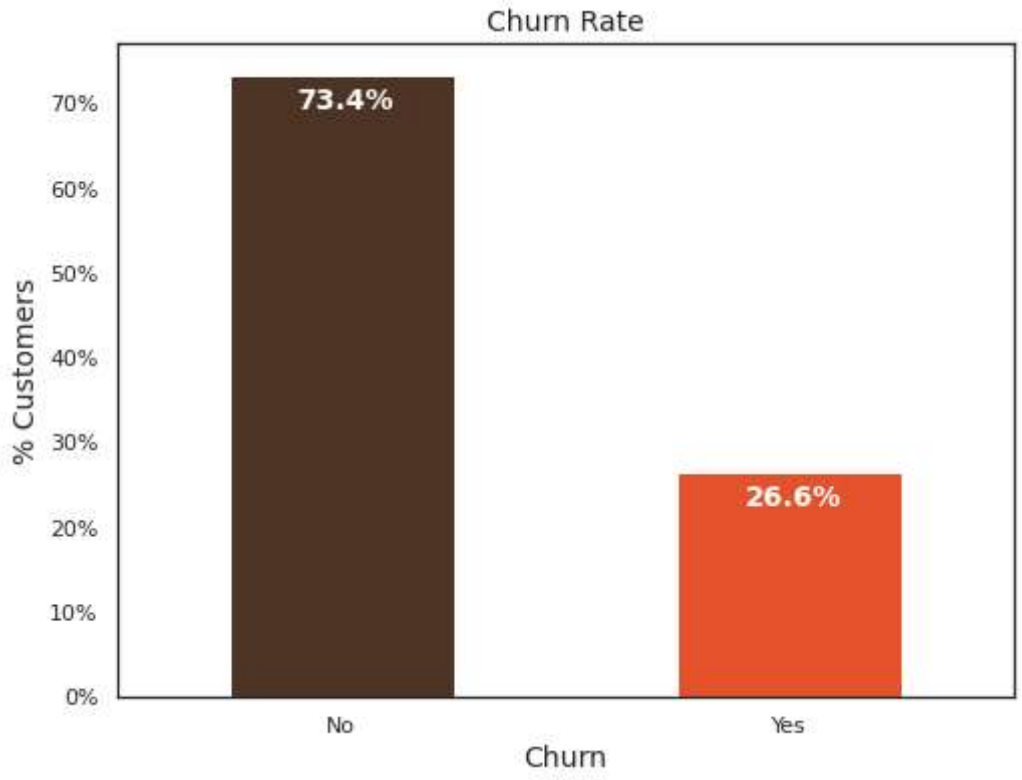
Out[37]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6a935dc910>
```



E.) Finally, let's take a look at out predictor variable (Churn) and understand its interaction with other important variables as was found out in the correlation plot

Lets first look at the churn rate in our data

In [38]:
```
colors = ['#4D3425','#E4512B']
ax = (telecom_cust['Churn'].value_counts()*100.0 /len(telecom_cust)).plot(kind='b
                                                                          stacke
                                                                         rot = 0
                                                                         color =
                                                                        figsize
ax.yaxis.set_major_formatter(mtick.PercentFormatter())
ax.set_ylabel('% Customers',size = 14)
ax.set_xlabel('Churn',size = 14)
ax.set_title('Churn Rate', size = 14)

# create a list to collect the plt.patches data
totals = []

# find the values and append to list
for i in ax.patches:
    totals.append(i.get_width())

# set individual bar lables using above list
total = sum(totals)

for i in ax.patches:
    # get_width pulls left or right; get_y pushes up or down
    ax.text(i.get_x()+.15, i.get_height()-4.0, \
            str(round((i.get_height()/total), 1))+'%',
            fontsize=12,
            color='white',
           weight = 'bold',
           size = 14)
```



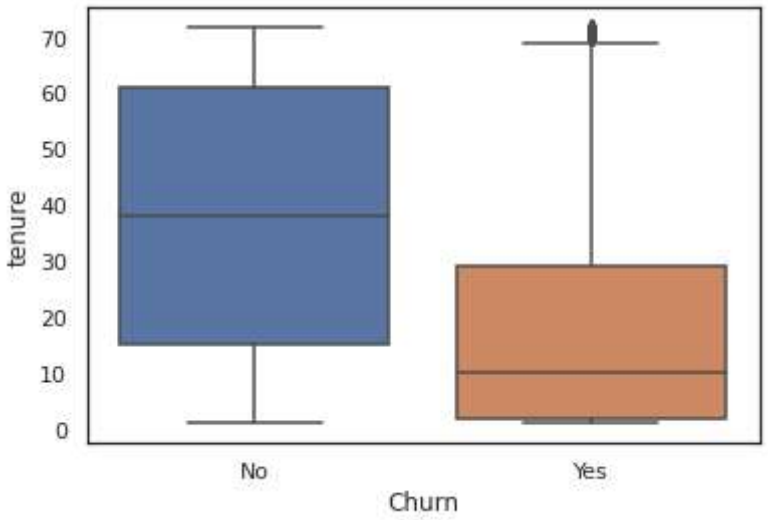
In our data, 74% of the customers do not churn. Clearly the data is skewed as we would expect a large majority of the customers to not churn. This is important to keep in mind for our modelling as skeweness could lead to a lot of false negatives. We will see in the modelling section on how to avoid skewness in the data.

Lets now explore the churn rate by tenure, seniority, contract type, monthly charges and total charges to see how it varies by these variables.

i.) Churn vs Tenure: As we can see form the below plot, the customers who do not churn, they tend to stay for a longer tenure with the telecom company

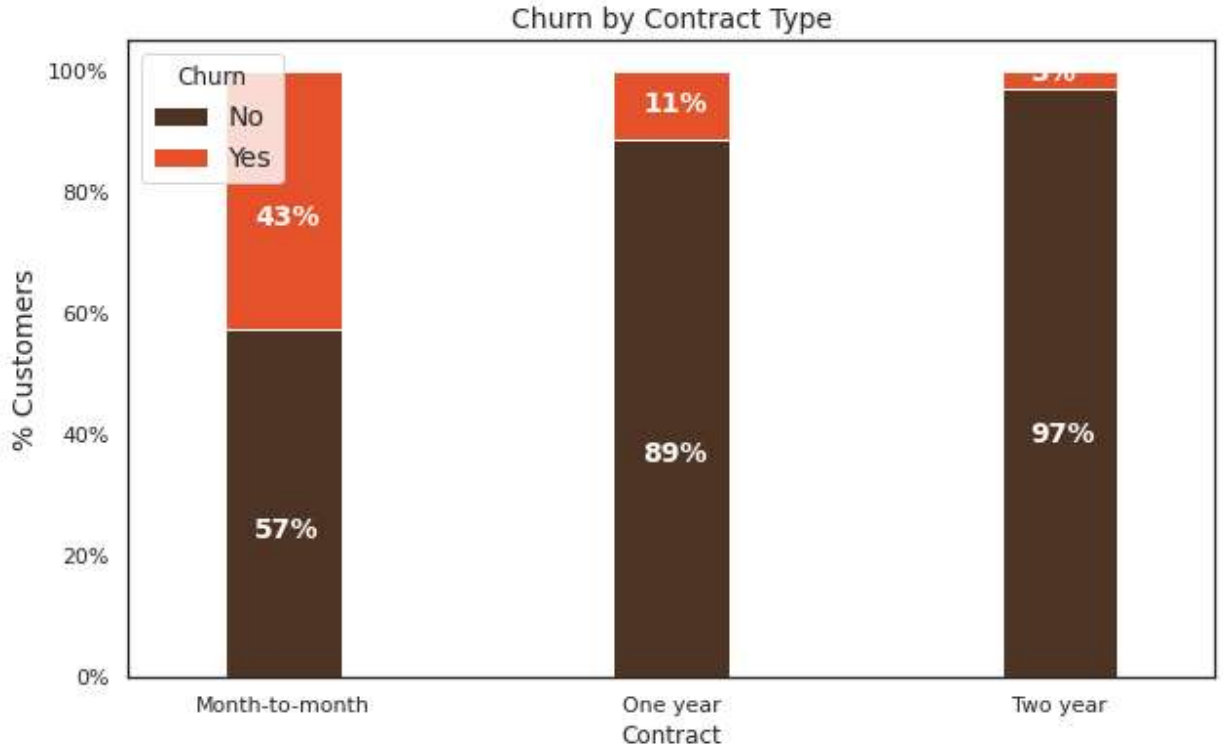
In [39]:
```
sns.boxplot(x = telecom_cust.Churn, y = telecom_cust.tenure)
```

Out[39]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6a935244d0>
```



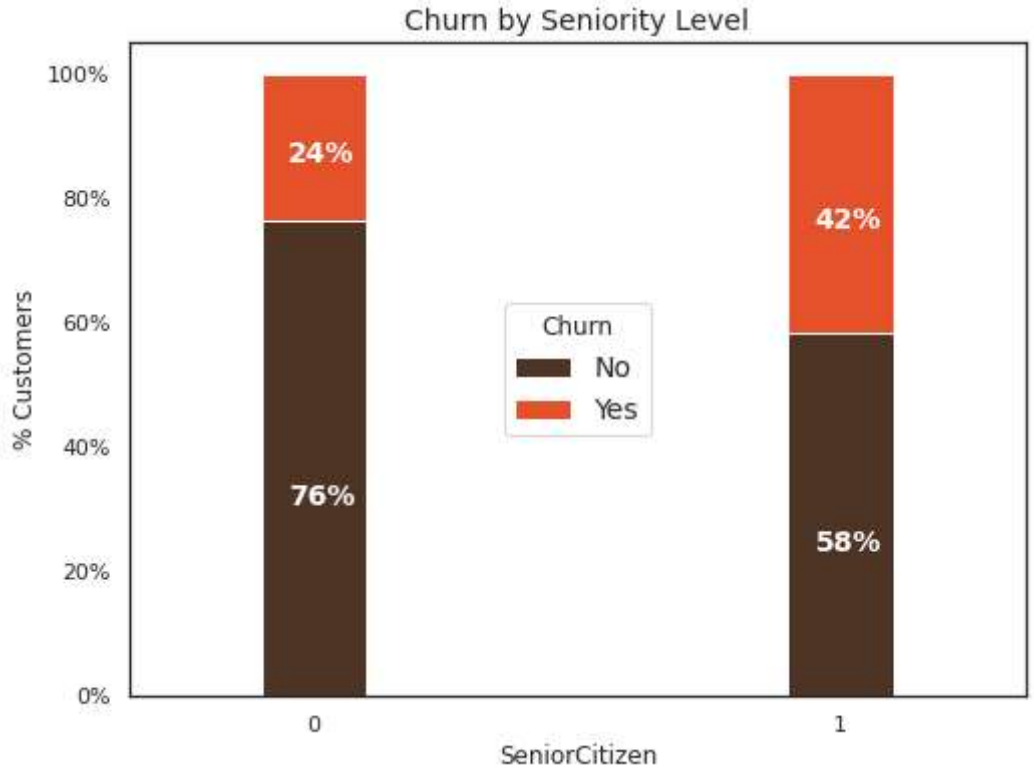
Churn by Contract Type: Similar to what we saw in the correlation plot, the customers who have a month to month contract have a very high churn rate.

In [40]:
```
colors = ['#4D3425','#E4512B']
contract_churn = telecom_cust.groupby(['Contract','Churn']).size().unstack()

ax = (contract_churn.T*100.0 / contract_churn.T.sum()).T.plot(kind='bar',
                                                              width = 0.3,
                                                              stacked = True,
                                                              rot = 0,
                                                              figsize = (10,6),
                                                              color = colors)
ax.yaxis.set_major_formatter(mtick.PercentFormatter())
ax.legend(loc='best',prop={'size':14},title = 'Churn')
ax.set_ylabel('% Customers',size = 14)
ax.set_title('Churn by Contract Type',size = 14)

# Code to add the data labels on the stacked bar chart
for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    x, y = p.get_xy()
    ax.annotate('{:.0f}%'.format(height), (p.get_x()+.25*width, p.get_y()+.4*heig
                color = 'white',
               weight = 'bold',
               size = 14)
```



Churn by Seniority: Senior Citizens have almost double the churn rate than younger population

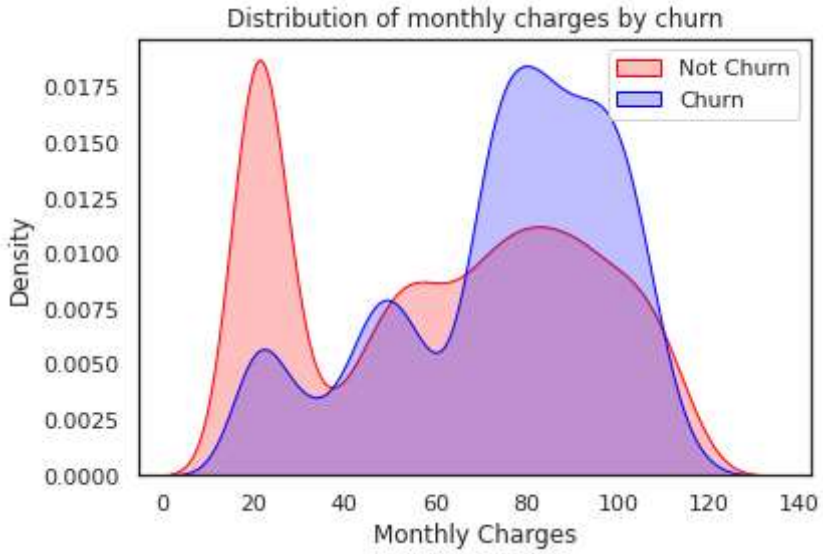
In [41]:
```
colors = ['#4D3425','#E4512B']
seniority_churn = telecom_cust.groupby(['SeniorCitizen','Churn']).size().unstack(

ax = (seniority_churn.T*100.0 / seniority_churn.T.sum()).T.plot(kind='bar',
                                                                width = 0.2,
                                                                stacked = True,
                                                                rot = 0,
                                                                figsize = (8,6),
                                                                color = colors)
ax.yaxis.set_major_formatter(mtick.PercentFormatter())
ax.legend(loc='center',prop={'size':14},title = 'Churn')
ax.set_ylabel('% Customers')
ax.set_title('Churn by Seniority Level',size = 14)

# Code to add the data labels on the stacked bar chart
for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    x, y = p.get_xy()
    ax.annotate('{:.0f}%'.format(height), (p.get_x()+.25*width, p.get_y()+.4*heig
                color = 'white',
               weight = 'bold',size =14)
```



Churn by Monthly Charges: Higher % of customers churn when the monthly charges are high.

In [42]:
```
ax = sns.kdeplot(telecom_cust.MonthlyCharges[(telecom_cust["Churn"] == 'No') ],
                 color="Red", shade = True)
ax = sns.kdeplot(telecom_cust.MonthlyCharges[(telecom_cust["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')
```

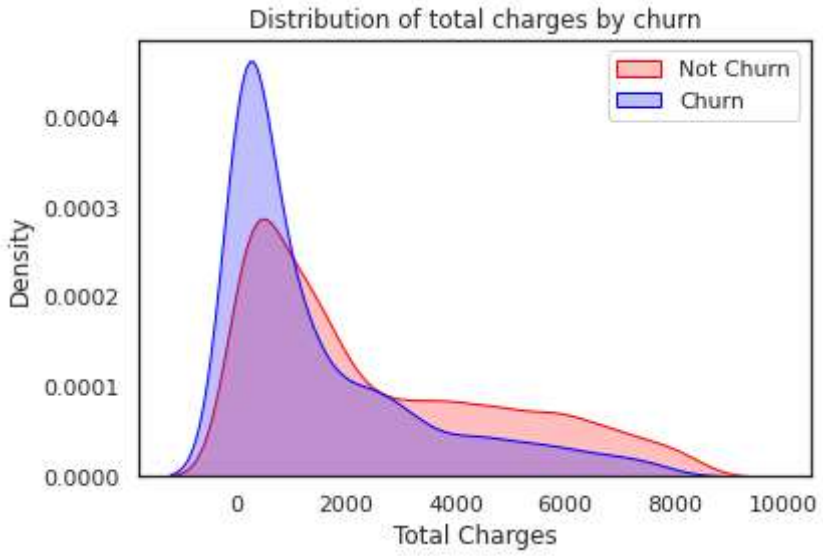
Out[42]:
```
Text(0.5, 1.0, 'Distribution of monthly charges by churn')
```



Churn by Total Charges: It seems that there is higer churn when the total charges are lower.

In [43]:
```
ax = sns.kdeplot(telecom_cust.TotalCharges[(telecom_cust["Churn"] == 'No') ],
                 color="Red", shade = True)
ax = sns.kdeplot(telecom_cust.TotalCharges[(telecom_cust["Churn"] == 'Yes') ],
                 ax =ax, color="Blue", shade= True)
ax.legend(["Not Churn","Churn"],loc='upper right')
ax.set_ylabel('Density')
ax.set_xlabel('Total Charges')
ax.set_title('Distribution of total charges by churn')
```

Out[43]:
```
Text(0.5, 1.0, 'Distribution of total charges by churn')
```



After going through the above EDA we will develop some predictive models and compare them.

1. Logistic Regression

In [44]:
```
# We will use the data frame where we had created dummy variables
y = df_dummies['Churn'].values
X = df_dummies.drop(columns = ['Churn'])

# Scaling all the variables to a range of 0 to 1
from sklearn.preprocessing import MinMaxScaler
features = X.columns.values
scaler = MinMaxScaler(feature_range = (0,1))
scaler.fit(X)
X = pd.DataFrame(scaler.transform(X))
X.columns = features
```

It is important to scale the variables in logistic regression so that all of them are within a range of 0 to 1. This helped me improve the accuracy from 79.7% to 80.7%. Further, you will notice below that the importance of variables is also aligned with what we are seeing in Random Forest algorithm and the EDA we conducted above.

In [45]:
```
# Create Train & Test Data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_s
```

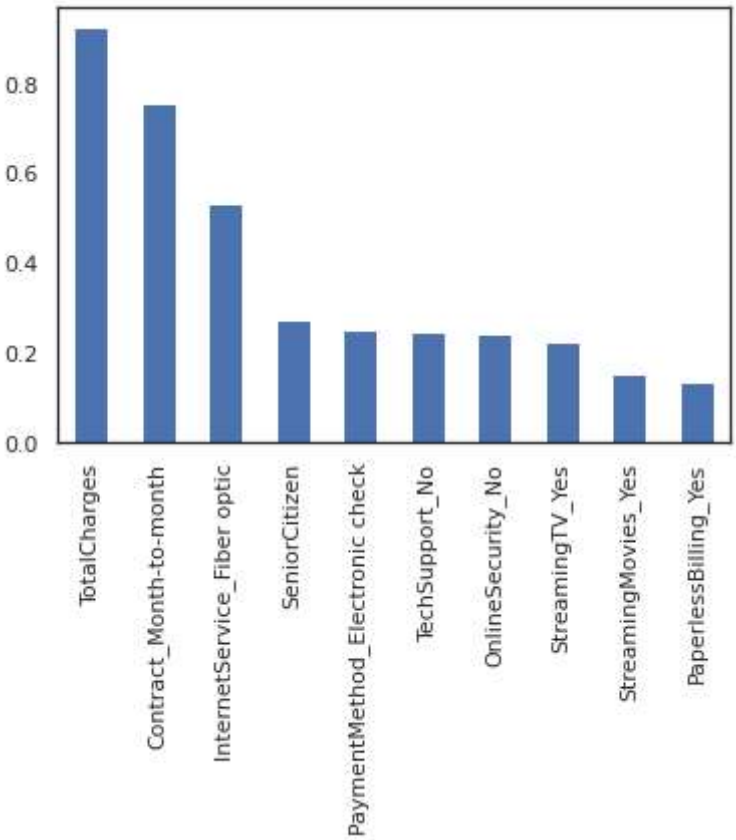
In [46]:
```
# Running logistic regression model
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
result = model.fit(X_train, y_train)
```

In [47]:
```
from sklearn import metrics
prediction_test = model.predict(X_test)
# Print the prediction accuracy
print (metrics.accuracy_score(y_test, prediction_test))
```

```
0.8075829383886256
```

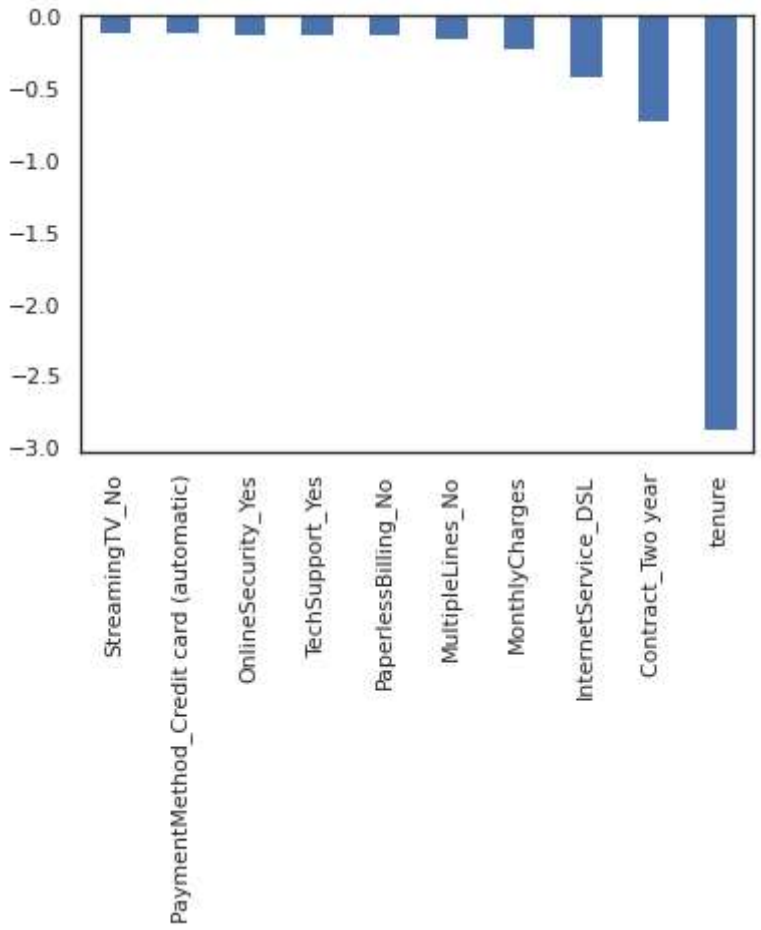
In [48]:
```
# To get the weights of all the variables
weights = pd.Series(model.coef_[0],
                    index=X.columns.values)
print (weights.sort_values(ascending = False)[:10].plot(kind='bar'))
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
```



In [49]:
```
print(weights.sort_values(ascending = False)[-10:].plot(kind='bar'))
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
```



Observations

We can see that some variables have a negative relation to our predicted variable (Churn), while some have positive relation. Negative relation means that likeliness of churn decreases with that variable. Let us summarize some of the interesting features below:

As we saw in our EDA, having a 2 month contract reduces chances of churn. 2 month contract along with tenure have the most negative relation with Churn as predicted by logistic regressions

Having DSL internet service also reduces the proability of Churn Lastly, total charges, monthly contracts, fibre optic internet services and seniority can lead to higher churn rates. This is interesting because although fibre optic services are faster, customers are likely to churn because of it. I think we need to explore more to better understad why this is happening. Any hypothesis on the above would be really helpful!

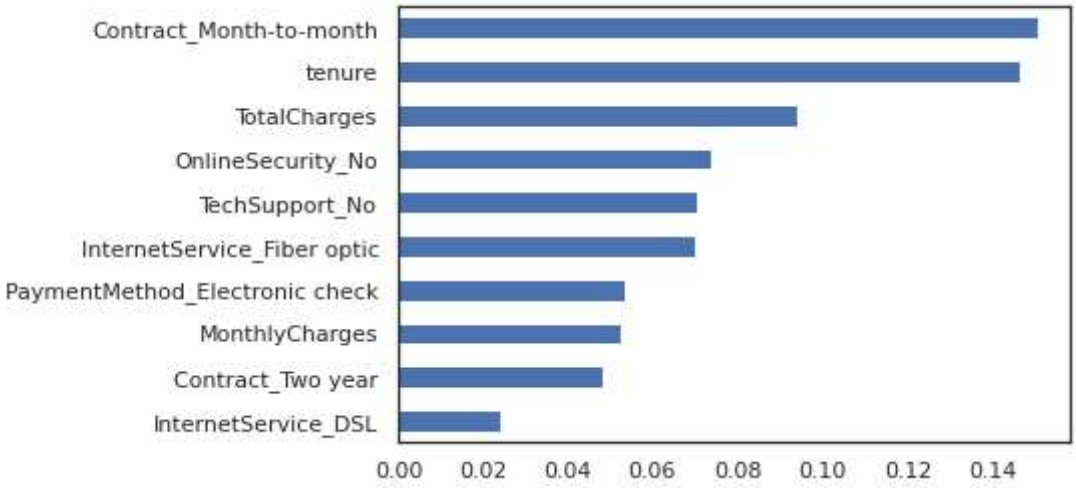
2. Random Forest

In [50]:
```
from sklearn.ensemble import RandomForestClassifier
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_s
model_rf = RandomForestClassifier(n_estimators=1000 , oob_score = True, n_jobs =
                                  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))
```

```
0.8088130774697939
```

In [51]:
```
importances = model_rf.feature_importances_
weights = pd.Series(importances,
                    index=X.columns.values)
weights.sort_values()[-10:].plot(kind = 'barh')
```

Out[51]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6a8f9c0b90>
```



Observations:

From random forest algorithm, monthly contract, tenure and total charges are the most important predictor variables to predict churn. The results from random forest are very similar to that of the logistic regression and in line to what we had expected from our EDA

3. Support Vecor Machine (SVM)

In [52]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_s
```

In [53]:
```
from sklearn.svm import SVC

model.svm = SVC(kernel='linear')
model.svm.fit(X_train,y_train)
preds = model.svm.predict(X_test)
metrics.accuracy_score(y_test, preds)
```

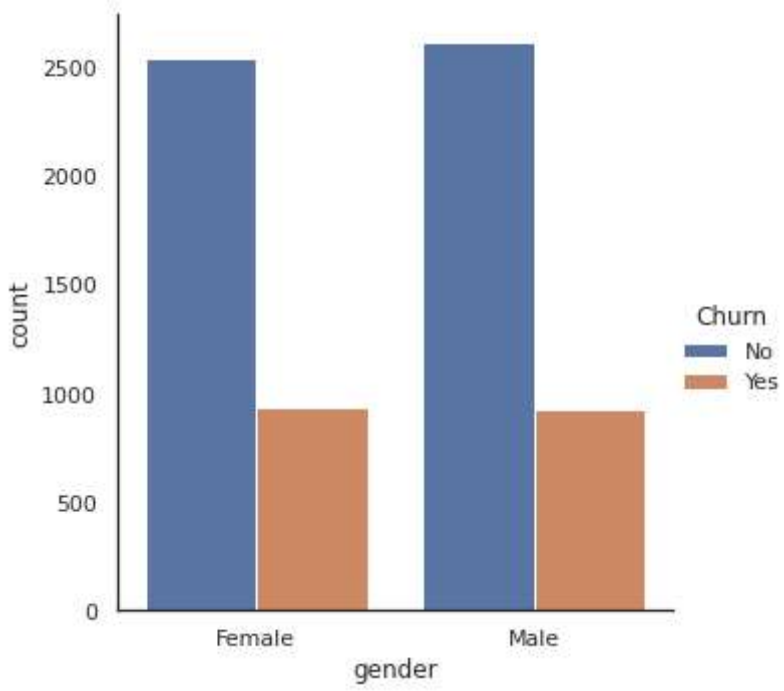
Out[53]:
```
0.820184790334044
```

In [54]:
```
# Create the Confusion matrix
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,preds))
```

```
[[953  89]
 [164 201]]
```

Wth SVM I was able to increase the accuracy to upto 82%. However, we need to take a deeper look at the true positive and true negative rates, including the Area Under the Curve (AUC) for a better prediction

In [55]:
```
ax1 = sns.catplot(x="gender", kind="count", hue="Churn", data=telecom_cust,
                  estimator=lambda x: sum(x==0)*100.0/len(x))
#ax1.yaxis.set_major_formatter(mtick.PercentFormatter())
```



4. ADA Boost

In [56]:
```
# AdaBoost Algorithm
from sklearn.ensemble import AdaBoostClassifier
model = AdaBoostClassifier()
# n_estimators = 50 (default value)
# base_estimator = DecisionTreeClassifier (default value)
model.fit(X_train,y_train)
preds = model.predict(X_test)
metrics.accuracy_score(y_test, preds)
```

Out[56]: 0.8159203980099502

5. XG Boost

In [57]:
```
from xgboost import XGBClassifier
model = XGBClassifier()
model.fit(X_train, y_train)
preds = model.predict(X_test)
metrics.accuracy_score(y_test, preds)
```

Out[57]: 0.8294243070362474

with XG Boost I was able to increase the accuracy on test data to almost 83%

Clearly, XG Boost is a winner among all other techniques

XG Boost is a slow learning model and is based on the concept of Boosting

RISK AND FRAUD ANALYTICS

REVIEW - 2

CUSTOMER CHURN ANALYSIS IN TELECOM INDUSTRY

TEAM MEMBERS:

ALAGARSAMY N | 19MIA1082

NIRANJAN J | 19MIA1003

VIGNESH N | 19MIA1093

ROSHAN SRINIVAAS S | 19MIA1001

ABINANDHAN KUMAR T S S | 19MIA1062

INTRODUCTION

Customer churn analysis and prediction in telecom sector is an issue now a days because it's very important for telecommunication industries to analyze behaviors of various customer to predict which customers are about to leave the from telecom company, So data mining techniques and algorithm plays an important role for companies in today's commercial conditions because gaining a new customer's cost is more than retaining the existing ones. We have planned to focus on various machine learning techniques for predicting customer churn through which we can build the classification models and also compare the performance of these models.

ABSTRACT

- Retaining customers is key to a company's success, especially in an industry as competitive as wireless services.
- Acquiring new customers is not only more difficult, but also much more costly to companies than maintaining existing customer relationships.
- In this project, we will predict behavior to retain customers at a home phone and internet service provider called Telco.
- We'll first use exploratory data analysis to understand the relationships between the features and the target variable and identify factors that are influential in predicting customer attrition.
- Using these features, We'll develop a predictive model to help the company proactively reduce their churn rate and use insights from the model to strengthen their customer retention strategies.

LITERATURE SURVEY

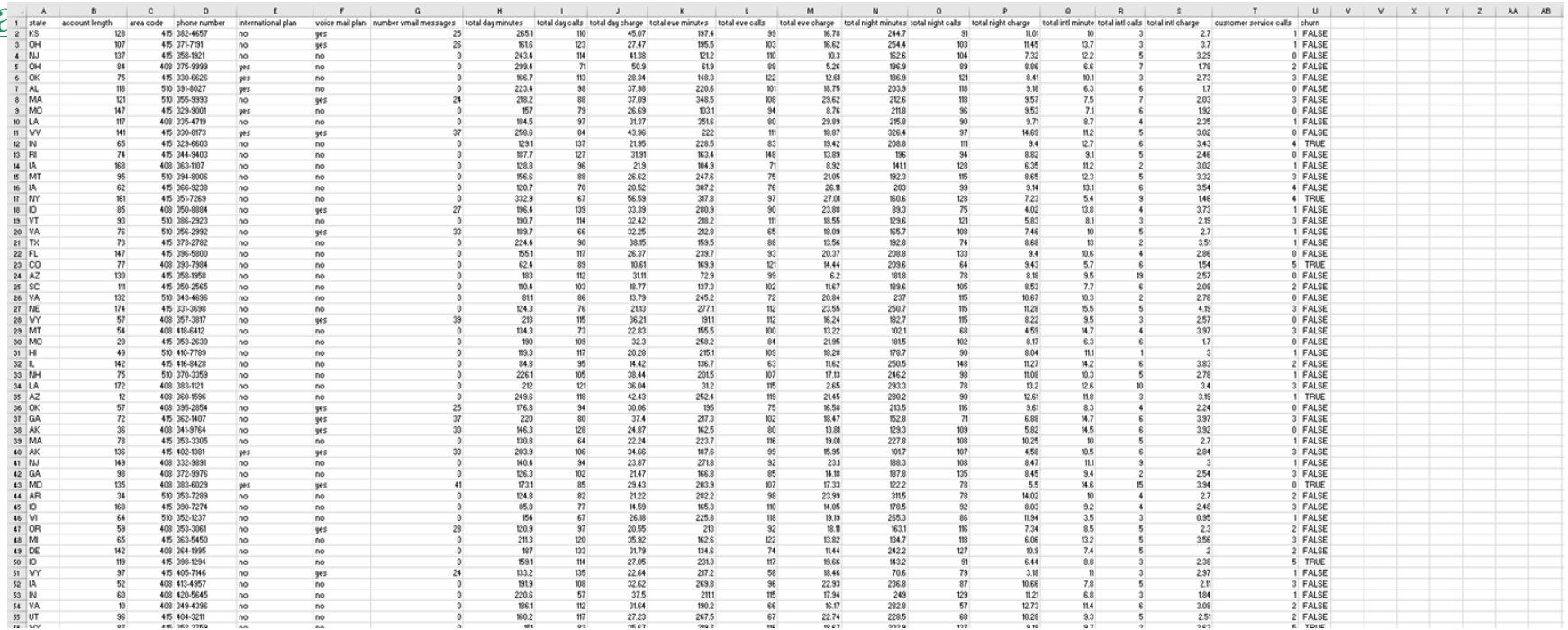
[1] M.A.H. Farquad [4] proposed a hybrid approach to overcome the drawbacks of general SVM model which generates a black box model (i.e., it does not reveal the knowledge gained during training in human understandable form).

[2] introduced the hybrid neural networks techniques to predict the customer churners in a CRM dataset provided by American telecom companies. Here, they built two hybrid models by combining two different neural network International Journal of Computer Applications (0975 – 8887).

[3] Wouter Verbeke [6] proposed the application of Ant-Miner+ and ALBA algorithms on a publicly available churn prediction dataset in order to build accurate as well as comprehensible classification rule-sets churn prediction models.

DATASET USED IN OUR PROJECT

SOURCE: https://www.kaggle.com/datasets/becksddf/churn-in-telecoms-da



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
1	state	account length	area code	phone number	international plan	voice mail plan	number vmail messages	total day minutes	total day calls	total day charge	total eve minutes	total eve calls	total eve charge	total night minutes	total night calls	total night charge	total intl minute	total intl calls	total intl charge	customer service calls	churn							
2	KS	128	415	382-4657	no	yes	25	265.1	110	45.07	197.4	99	16.78	244.7	91	11.01	10	3	2.7	1	FALSE							
3	OH	107	415	371-7191	no	yes	26	161.6	123	27.47	195.5	103	16.62	254.4	103	11.45	13.7	3	3.7	1	FALSE							
4	NJ	137	415	358-1921	no	no	0	243.4	114	41.38	121.2	110	10.3	162.6	104	7.32	12.2	5	3.29	0	FALSE							
5	OH	84	408	375-9999	yes	no	0	299.4	71	50.9	61.9	88	5.26	196.9	89	8.86	6.6	7	1.78	2	FALSE							
6	OK	75	415	330-6626	yes	no	0	166.7	113	28.34	148.3	122	12.61	186.9	121	8.41	10.1	3	2.73	3	FALSE							
7	AL	118	510	391-8027	yes	no	0	223.4	98	37.98	220.6	101	18.75	203.9	118	9.18	6.3	6	1.7	0	FALSE							
8	MA	121	510	355-9993	no	yes	24	218.2	88	37.09	348.5	108	29.62	212.6	118	9.57	7.5	7	2.03	3	FALSE							
9	MO	147	415	329-9001	yes	no	0	157	79	26.69	103.1	94	8.76	211.8	96	9.53	7.1	6	1.92	0	FALSE							
10	LA	117	408	335-4719	no	no	0	184.5	97	31.37	351.6	80	29.89	215.8	90	9.71	8.7	4	2.35	1	FALSE							
11	VY	141	415	330-8173	yes	yes	37	258.6	84	43.96	222	111	18.87	326.4	97	14.69	11.2	5	3.02	0	FALSE							
12	IN	65	415	329-6603	no	no	0	129.1	137	21.95	228.5	83	19.42	208.8	111	9.4	12.7	6	3.43	4	TRUE							
13	RI	74	415	344-9403	no	no	0	187.7	127	31.91	163.4	148	13.89	196	94	8.82	9.1	5	2.46	0	FALSE							
14	IA	168	408	363-1107	no	no	0	128.8	96	21.9	104.9	71	8.92	141.1	128	6.35	11.2	2	3.02	1	FALSE							
15	MT	95	510	394-8006	no	no	0	156.6	88	26.62	247.6	75	21.05	192.3	115	8.65	12.3	5	3.32	3	FALSE							
16	IA	62	415	366-9238	no	no	0	120.7	70	20.52	307.2	76	26.11	203	99	9.14	13.1	6	3.54	4	FALSE							
17	NY	161	415	351-7269	no	no	0	332.9	67	56.59	317.8	97	27.01	160.6	128	7.23	5.4	9	1.46	4	TRUE							
18	ID	85	408	350-8884	no	yes	27	196.4	139	33.39	280.9	90	23.88	89.3	75	4.02	13.8	4	3.73	1	FALSE							
19	VT	93	510	386-2923	no	no	0	190.7	114	32.42	218.2	111	18.55	129.6	121	5.83	8.1	3	2.19	3	FALSE							
20	VA	76	510	356-2992	no	yes	33	189.7	66	32.25	212.8	65	18.09	165.7	108	7.46	10	5	2.7	1	FALSE							
21	TX	73	415	373-2782	no	no	0	224.4	90	38.15	159.5	88	13.56	192.8	74	8.68	13	2	3.51	1	FALSE							
22	FL	147	415	396-5800	no	no	0	155.1	117	26.37	239.7	93	20.37	208.8	133	9.4	10.6	4	2.86	0	FALSE							
23	CO	77	408	393-7984	no	no	0	62.4	89	10.61	169.9	121	14.44	209.6	64	9.43	5.7	6	1.54	5	TRUE							
24	AZ	130	415	358-1958	no	no	0	183	112	31.11	72.9	99	6.2	181.8	78	8.18	9.5	19	2.57	0	FALSE							
25	SC	111	415	350-2565	no	no	0	110.4	103	18.77	137.3	102	11.67	189.6	105	8.53	7.7	6	2.08	2	FALSE							
26	VA	132	510	343-4696	no	no	0	81.1	86	13.79	245.2	72	20.84	237	115	10.67	10.3	2	2.78	0	FALSE							
27	NE	174	415	331-3698	no	no	0	124.3	76	21.13	277.1	112	23.55	250.7	115	11.28	15.5	5	4.19	3	FALSE							
28	VY	57	408	357-3817	no	yes	39	213	115	36.21	191.1	112	16.24	182.7	115	8.22	9.5	3	2.57	0	FALSE							
29	MT	54	408	418-6412	no	no	0	134.3	73	22.83	155.5	100	13.22	102.1	68	4.59	14.7	4	3.97	3	FALSE							
30	MO	20	415	353-2630	no	no	0	190	109	32.3	258.2	84	21.95	181.5	102	8.17	6.3	6	1.7	0	FALSE							
31	HI	49	510	410-7789	no	no	0	119.3	117	20.28	215.1	109	18.28	178.7	90	8.04	11.1	1	3	1	FALSE							
32	IL	142	415	416-8428	no	no	0	84.8	95	14.42	136.7	63	11.62	250.5	148	11.27	14.2	6	3.83	2	FALSE							
33	NH	75	510	370-3059	no	no	0	226.1	105	38.44	201.5	107	17.13	246.2	98	11.08	10.3	5	2.78	1	FALSE							
34	LA	172	408	383-1121	no	no	0	212	121	36.04	31.2	115	2.65	293.3	78	13.2	12.6	10	3.4	3	FALSE							
35	AZ	12	408	360-1596	no	no	0	249.6	118	42.43	252.4	119	21.45	280.2	90	12.61	11.8	3	3.19	1	TRUE							
36	OK	57	408	395-2954	no	yes	25	176.8	94	30.06	195	75	16.58	213.5	116	9.61	8.3	4	2.24	0	FALSE							
37	GA	72	415	362-3407	no	yes	37	220	80	37.4	217.3	102	18.47	152.8	71	6.88	14.7	6	3.97	3	FALSE							
38	AK	36	408	341-9764	no	yes	30	146.3	128	24.87	162.5	80	13.81	129.3	109	5.82	14.5	6	3.92	0	FALSE							
39	MA	78	415	353-3305	no	no	0	130.8	64	22.24	223.7	116	19.01	227.8	108	10.25	10	5	2.7	1	FALSE							
40	AK	136	415	402-1381	yes	yes	33	203.9	106	34.66	187.6	99	15.95	101.7	107	4.58	10.5	6	2.84	3	FALSE							
41	NJ	149	408	332-9891	no	no	0	140.4	94	23.87	271.8	92	23.1	188.3	108	8.47	11.1	9	3	1	FALSE							
42	GA	98	408	372-9976	no	no	0	126.3	102	21.47	168.8	85	14.18	197.8	135	8.45	9.4	2	2.54	3	FALSE							
43	MD	135	408	383-6029	yes	yes	41	173.1	85	29.43	203.9	107	17.33	122.2	78	5.5	14.6	15	3.94	0	TRUE							
44	AR	34	510	353-7289	no	no	0	124.8	82	21.22	282.2	98	23.99	311.5	79	14.02	10	4	2.7	2	FALSE							
45	ID	160	415	390-7274	no	no	0	85.8	77	14.59	165.3	110	14.05	178.5	92	8.03	9.2	4	2.48	3	FALSE							
46	VI	64	510	352-1237	no	no	0	154	67	26.18	225.8	118	19.19	265.3	86	11.94	3.5	3	0.95	1	FALSE							
47	OR	59	408	353-3061	no	yes	28	120.9	97	20.55	213	92	18.11	163.1	116	7.34	8.5	5	2.3	2	FALSE							
48	MI	65	415	363-5450	no	no	0	211.3	120	35.92	162.6	122	13.82	134.7	118	6.06	13.2	5	3.56	3	FALSE							
49	DE	142	408	364-1995	no	no	0	187	133	31.79	134.6	74	11.44	242.2	127	10.9	7.4	5	2	2	FALSE							
50	ID	119	415	398-1294	no	no	0	159.1	114	27.05	231.3	117	19.66	143.2	91	6.44	8.8	3	2.38	5	TRUE							
51	VY	97	415	405-7146	no	yes	24	133.2	135	22.64	217.2	58	18.46	70.6	79	3.18	11	3	2.97	1	FALSE							
52	IA	62	408	413-4957	no	no	0	191.9	108	32.62	269.8	96	22.93	236.8	87	10.66	7.8	5	2.11	0	FALSE							
53	IN	60	408	420-5645	no	no	0	220.6	57	37.5	211.1	115	17.94	249	129	11.21	6.8	3	1.84	1	FALSE							
54	VA	10	408	349-4396	no	no	0	186.1	112	31.64	190.2	66	16.17	282.8	57	12.73	11.4	6	3.08	2	FALSE							
55	UT	96	415	404-3211	no	no	0	160.2	117	27.23	267.5	67	22.74	228.5	68	10.28	9.3	5	2.51	2	FALSE							

METHODOLOGY

- **Initially we have explored and Visualized various plots on the data to understand the various relations between the predictor and features in the dataset.**
- **To forecast the churn factor, five machine learning techniques were used: logistic regression, XG Boost, Random Forest, SVM, and ADA Boost. From these three models, we discovered that XG Boost's performance is excellent in comparison to other models, increasing the accuracy of the test results.**

CONCLUSION

- The purpose of this kind of study in the telecom industry is to assist businesses in increasing their profits.
- It is well known that one of the most significant revenue streams for telecom firms is churn prediction.
- As a result, the goal of this research was to develop a system that could forecast customer turnover at the telecom business.

REFERRENCE

[1] T.Vafeiadis, K.I. Diamantaras, G.Sarigiannidis, K.Chatzisavvas “Customer churn prediction in telecommunications”, Simulation Modelling: Practice and Theory 55 (2015) 1-9.

[2] Burez J., & Van den Poel, D “Crm at a pay-TV company: Using analytical models to reduce customer attrition by targeted marketing for subscription services”, Expert Systems with Applications 32, 277–288.

[3] S.Parvathavardhini and Dr. S.Manju “Analysis on Machine Learning Techniques” International Journal of Computer Sciences and Engineering (IJCSE), Vol-4(8), pp 59-77 Aug 2016, E-ISSN: 2347-2693.

REFERENCE

[4] M.A.H. Farquad, Vadlamani Ravi, S. Bapi Raju “Churn prediction using comprehensible support vector machine: An analytical CRM application”, Applied Soft Computing 19 (2014) 31–40.

[5] Chih-Fong Tsai, Yu-Hsin Lu “Customer churn prediction by hybrid neural networks”, Expert Systems with Applications 36 (2009) 12547–12553.

[6] Wouter Verbeke, David Martens, Christophe Mues, Bart Baesens “Building comprehensible customer churn prediction models with advanced rule induction techniques”, Expert Systems with Applications 38 (2011) 2354–2364.

THANK YOU