**Mini Project on Business Intelligence**

1. **Problem definition**

Predicting behavior to retain customers for Telecom Services; analyzing all relevant customer data and developing focused customer retention. Churn is a one of the biggest problem in the telecom industry. Research has shown that the average monthly churn rate among the top 4 wireless carriers in the US is 1.9% - 2%. The churn rate is the percentage of subscribers to a service who discontinue their subscriptions to the service within a given time period. Hence, with this project, using the dataset provided and the various attribute related to Telecom Customers, we predict the attributes that strongly affect churn rate which will eventually help retain customers for Telecom Services.

1. Link of the selected data set

<https://www.kaggle.com/blastchar/telco-customer-churn>

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

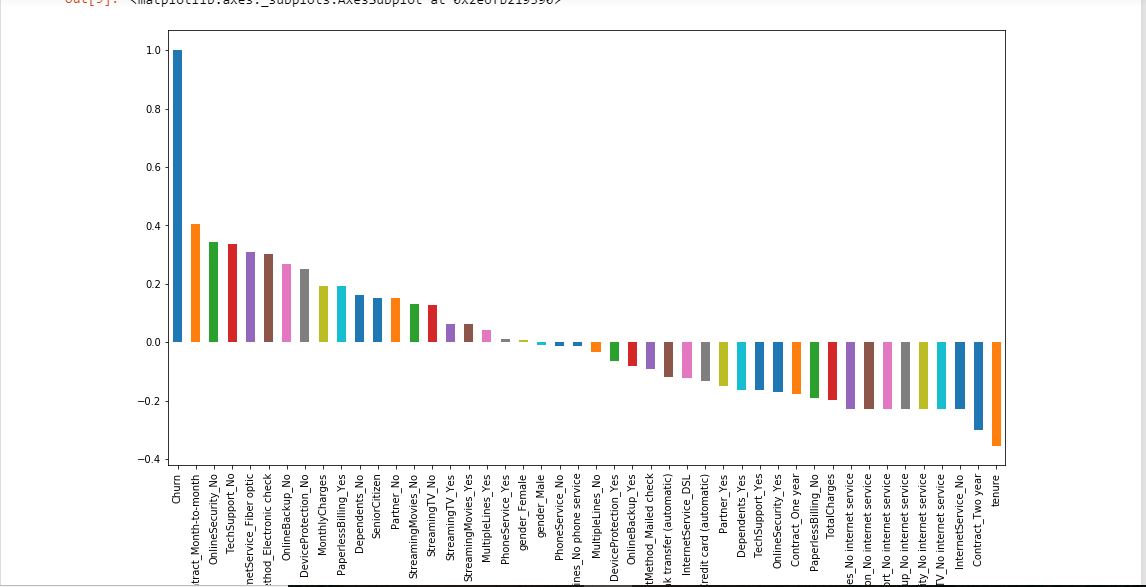
1. Code for steps performed in data exploration and final output of the dataset (screenshots 5-10 tuples)

**Correlation of "Churn" with other variables:**

**Code:**

plt.figure(figsize=(15,8))

df\_dummies.corr()['Churn'].sort\_values(ascending = False).plot(kind='bar')



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

**Churn Rate :**

**Code:**

colors = ['#4D3425','#E4512B']

ax = (data['Churn'].value\_counts()\*100.0 /len(data)).plot(kind='bar', stacked = True, rot = 0, color = colors, figsize = (8,6))

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, \

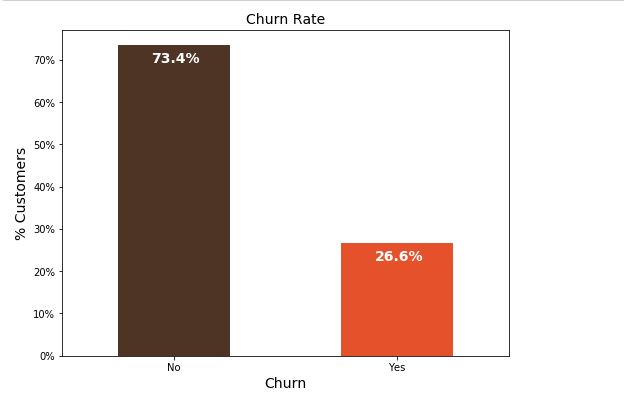
str(round((i.get\_height()/total), 1))+'%',

fontsize=12,

color='white',

weight = 'bold',

size = 14)



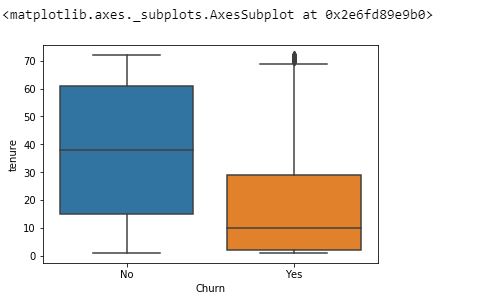
74% of the customers do not churn. Clearly the data is skewed as we would expect a large majority of the customers to not churn

**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.

**Code**

sns.boxplot(x = data.Churn, y = data.tenure)



**ii.) 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.

**Code**

colors = ['#4D3425','#E4512B']

contract\_churn = data.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)

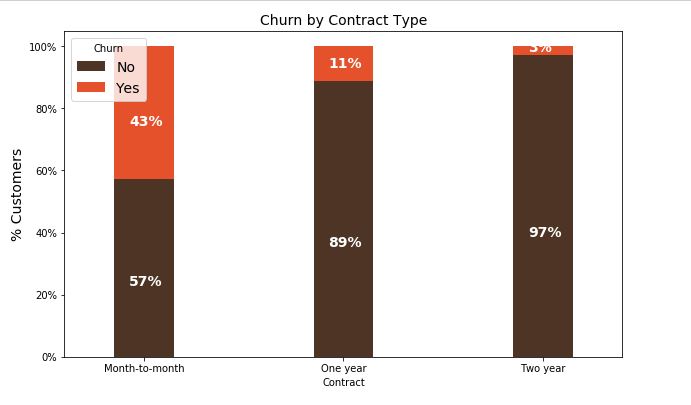
# Code to add the data labels on the stacked bar chart

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\*height),color = 'white', weight = 'bold', size = 14)



**iii.) Churn by Seniority:**

Senior Citizens have almost double the churn rate than younger population.

**Code**

colors = ['#4D3425','#E4512B']

seniority\_churn = data.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)

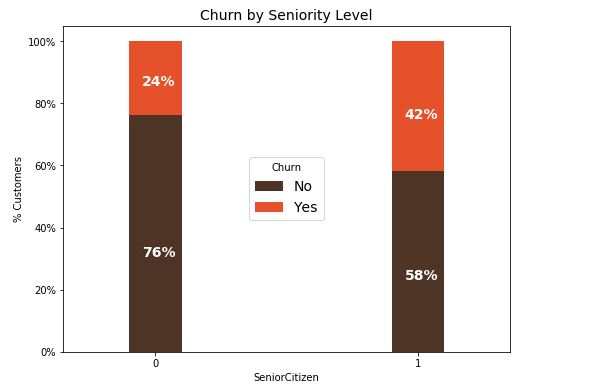
# Code to add the data labels on the stacked bar chart

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\*height), color = 'white', weight = 'bold',size =14)



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

**Code**

ax = sns.kdeplot(data.MonthlyCharges[(data["Churn"] == 'No') ],

color="Red", shade = True)

ax = sns.kdeplot(data.MonthlyCharges[(data["Churn"] == 'Yes') ],

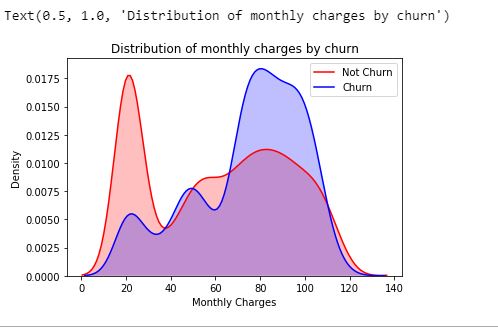
ax =ax, color="Blue", shade= True)

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')



**v.) Churn by Total Charges:** It seems that there is higher churn when the total charges are lower.

**Code:**

ax = sns.kdeplot(data.TotalCharges[(data["Churn"] == 'No') ],

color="Red", shade = True)

ax = sns.kdeplot(data.TotalCharges[(data["Churn"] == 'Yes') ],

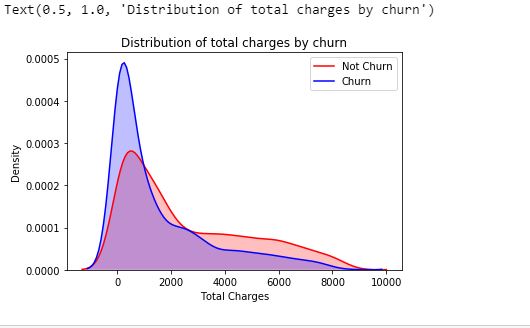
ax =ax, color="Blue", shade= True)

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')



1. Code for steps performed in data pre-processing and final output of the dataset (screenshots 5-10 tuples)

# 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

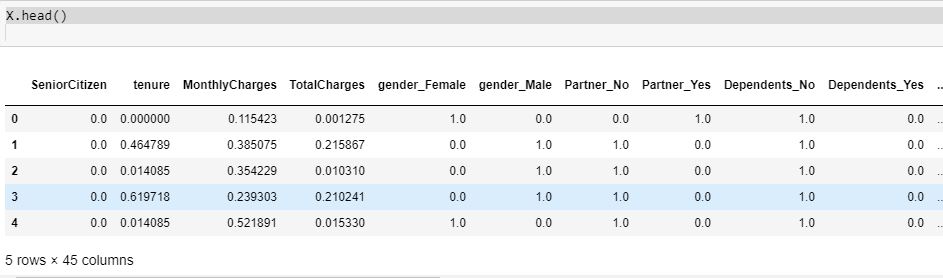
scaler = MinMaxScaler(feature\_range = (0,1))

scaler.fit(X)

X = pd.DataFrame(scaler.transform(X))

X.columns = features

Output after preprocessing:



1. Justification on why a particular data mining task is chosen. Also justify the algorithm selected.

Given various attributes related to churn rate like contracts, customer’s gender, his partners and other factors we need to predict which attributes strongly affect the churn rate. This is the case of supervised learning and since we need to predict the attributes that strongly affect the churn rate, random forest classification algorithm will be used.

We have used random forest classification algorithm for the following reasons:

* Reduction in overfitting: by averaging several trees, there is a significantly lower risk of overfitting.
* Less variance: By using multiple trees, you reduce the chance of stumbling across a classifier that doesn’t perform well because of the relationship between the train and test data.
* Also, random forest gave a better accuracy than logistic regression and naïve bayes classification algorithm.
* One of the biggest advantages of random forest over decision tree is the algorithm on which the former one works I.e. Bagging algorithm.

1. Code of the selected algorithm – Random Forest Classification

from sklearn.ensemble import RandomForestClassifier

from sklearn import metrics

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=101)

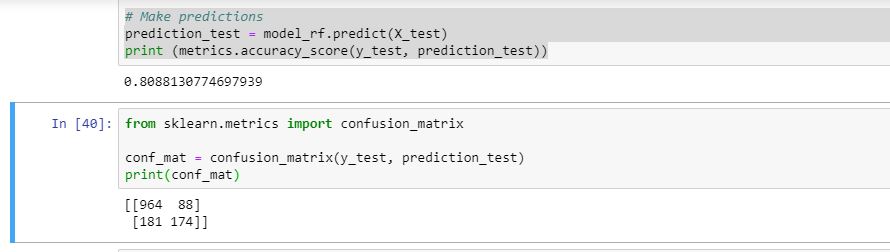
model\_rf = RandomForestClassifier(n\_estimators=1000 , oob\_score = True, n\_jobs = -1,

random\_state =50, max\_features = "auto",

max\_leaf\_nodes = 30)

model\_rf.fit(X\_train, y\_train)

1. Detailed output of the selected algorithm - screenshots (training and testing phase results if any)



8. Visualization of the result using any tool – screenshot

We used matplotlib to visualize the attributes strongly affecting the churn rate.

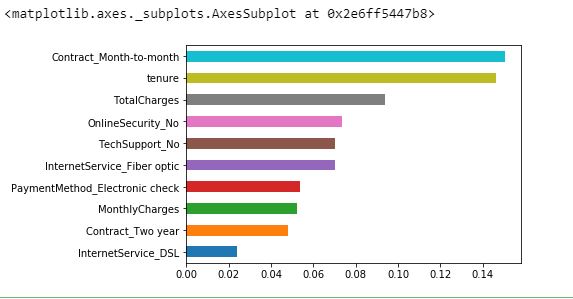
**Code :**

importances = model\_rf.feature\_importances\_

weights = pd.Series(importances,

index=X.columns.values)

weights.sort\_values()[-10:].plot(kind = 'barh')



9. Interpret the results obtained

Observations:

From random forest algorithm, monthly contract, tenure and total charges are the most important predictor variables to predict churn rate. The results from random forest are very similar to what we had expected from our exploratory data analysis.

10. Provide BI decision that can be taken based on the result obtained

From above observations we get an insight that monthly contract, tenure and total charges greatly affect churn rate. So, by implementing changes in the above attributes, we can considerably reduce the churn rate for Telecom Customers and retain customers for the same.

We inferred the following from the exploratory analysis and our data mining algorithm:

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

ii.) 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.

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

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

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