2REPORT

Group : 16

Group Members : CHANDANA BUDAATI(1002087323), DEEPTHI BURADA(1002034183), SANJANA POTLURI(1002147971), VARSHITH KONDURU(1002132051)

TITLE: Forecasting Telecom Subscriber Turnover

1. INTRODUCTION:

Customer churn represents a significant challenge, particularly for large companies, with direct implications for revenue, especially within the telecommunications sector. To address this issue, companies are actively seeking ways to predict potential churn among their customer base. Identifying factors that contribute to churn is crucial for implementing effective strategies to mitigate it. Our research aims to contribute by developing a churn prediction model using machine learning techniques on big data platforms, employing novel methods for feature engineering and selection.

In developed countries, the telecommunications industry has become a vital sector characterized by intense competition among numerous operators. To thrive in this competitive landscape, companies employ various strategies to boost revenue, including acquiring new customers, upselling to existing ones, and prolonging customer retention. However, analysis reveals that the most profitable strategy, considering return on investment (RoI), is the third option—increasing customer retention. Retaining existing customers is not only more cost-effective than acquiring new ones but also simpler than upselling. This underscores the importance of reducing churn to execute this strategy effectively.

Our focus is on evaluating and analyzing the efficacy of tree-based machine learning methods and algorithms in predicting churn for telecommunications companies. Through experimentation with algorithms such as Decision Trees and Random Forests, we aim to construct a predictive model for customer churn following comprehensive data preparation, feature engineering, and feature selection processes.

2. MOTIVATION:

Retaining existing customers is not just a matter of cost-effectiveness; it's also a strategic advantage. Research underscores this point by revealing that the expense of acquiring a new customer can be staggering—up to twenty-five times more costly than retaining an existing one. But the benefits extend beyond mere financial savings. Loyal customers tend to be more than just repeat buyers; they often explore additional products and services and display a higher tolerance for price fluctuations, thereby fostering stable revenue growth. Even a modest uptick in customer retention rates can translate into a significant uptick in profitability, making it a linchpin of sustainable business success.

In the cutthroat arena of telecommunications, where offerings can quickly become commoditized, the ability to provide exceptional customer experiences and curb churn rates can be pivotal. It's not just about offering competitive prices or the latest gadgets; it's about fostering meaningful connections with customers that keep them coming back.

Thankfully, the telecom industry is awash in data, providing fertile ground for leveraging advanced analytics and machine learning. By harnessing the power of predictive modeling, telecom companies can identify customers

who are at risk of defecting before it's too late. Armed with this insight, they can tailor retention strategies to address individual needs, whether through personalized incentives, targeted communications, or proactive customer support.

In essence, the ability to predict and prevent churn isn't just a luxury in today's telecom landscape—it's a necessity. And with the right tools and strategies at their disposal, companies can turn churn prevention into a competitive advantage that sets them apart in a crowded marketplace.

3. LITERATURE REVIEW:

Customer Churn Prediction in Telecommunication Industry: A Review and Future Directions" by Nguyen et al. (2019). This comprehensive review paper provides an overview of the state-of-the-art techniques and methodologies used in telecom churn prediction. It discusses various machine learning algorithms, feature engineering methods, and evaluation metrics employed in previous studies. The paper also highlights emerging trends and future research directions in the field, such as the integration of big data analytics and deep learning techniques.

Telecom Customer Churn Prediction Using Machine Learning Algorithms: A Systematic Review and Meta-Analysis by Sharma et al. (2020). This systematic review and meta-analysis analyze the performance of different machine learning algorithms for telecom churn prediction. The study compares the accuracy, precision, recall, and F1-score of algorithms such as Decision Trees, Random Forests, Support Vector Machines, and Neural Networks. It provides insights into the strengths and limitations of each algorithm and offers recommendations for selecting the most suitable approach based on specific requirements.

A Review on Telecom Customer Churn Prediction Using Data Mining Techniques by Khan et al. (2018). This review paper explores the application of data mining techniques, including classification algorithms, clustering methods, and association rule mining, for telecom churn prediction. It discusses the importance of feature selection and extraction in improving prediction accuracy and model interpretability. The paper also highlights the role of ensemble learning and hybrid approaches in enhancing the robustness and generalization ability of churn prediction models.

Predicting Telecom Customer Churn with Advanced Machine Learning Techniques: A Comparative Study by Li et al. (2017). This comparative study evaluates the performance of advanced machine learning techniques, such as Gradient Boosting Machines, XGBoost, and Deep Learning, for telecom churn prediction. The research compares the predictive accuracy, computational efficiency, and interpretability of different models using real-world telecom datasets. It provides insights into the strengths and weaknesses of each technique and offers practical recommendations for deploying effective churn prediction systems in telecom companies.

Churn Prediction in Telecom Using Machine Learning in Big Data Platform by Chen et al. (2016). This research paper presents a case study on churn prediction in the telecom industry using machine learning techniques on big data platforms. The study focuses on feature engineering, model selection, and performance evaluation using real-world telecom data. It discusses the challenges and opportunities associated with implementing churn prediction systems in large-scale telecom operations and offers insights into practical considerations for successful deployment.

4. METHODOLOGY:

wa_fn-usec_-telco-customer-churn.csv is used as Dataset for out testing in the project. This dataset contains the following information:

a. Customers who left within the last month – the column is called Churn.

b. 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.

c. Customer account information – how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges.

d. Demographic info about customers – gender, age range, and if they have partners and dependents.

	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	OnlineBacl	DevicePro	TechSuppc	Streaming	Streaming	Contract	PaperlessB	PaymentM	MonthlyCh	TotalCharg	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-QPYI	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-TTV/	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

Each row represents a customer, each column contains customer's attributes described on the column Metadata. The raw data contains 7043 rows (customers) and 21 columns (features). The "Churn" column is our target. And we will be developing our model to predict if the customer is likely to churn based on the features present in the dataset, employing machine learning techniques to analyze patterns and make informed predictions about future customer behavior.

We will develop Logistic Regression, Random Forest, SVM and Compare them:

I. **Logistic Regression:**

Based on the problem statement, we need a predictive model that can do a binary classification or predict Yes/No or 1/0 type of output variable. One predictive model commonly implemented for binary classification and prediction of binary outcome is Logistic Regression. Logistic regression is a binary classification algorithm belonging to the generalized linear regression model. It can also be used to solve problems with more than 2 classes. It is possible to use logistic regression to create a model using the customer churn data and use it to predict if a particular customer of a set of customers will discontinue the service.

II. **Random Forest:**

Based on the problem summary, we need a predictive model that can predictively modeling on the monthly cost service or monthly payments for the customers. There are several categorical fields in the customer churn data, and there are several numerical fields too. It has information such as if the customer is using internet service, phone service, multiple services. Usually, the payment is associated with a particular service, say, if you get phone service you will pay a fixed amount per month and if you upgrade with the internet, the cost will increase by a certain amount. Other variables like tenure, age, bandwidth usage also may impact monthly payments. Based on decisions such as, if the customer has fiber optic internet, or if the customer streams tv or movies, it should be possible to estimate the monthly payment.

III. **Support Vector Machine (SVM):**

To improve the prediction abilities of machine learning methods, a support vector machine (SVM) on structural risk minimization was applied to customer churn prediction. Researching customer churn prediction cases both in home and foreign carries, the method was compared with artificial neural network, decision tree, logistic regression, and naive bayesian classifier. It is found that the method enjoys the best accuracy rate, hit rate, covering rate, and lift coefficient, and therefore, provides an effective measurement for customer churn prediction.

5. EXPERIMENTAL SETUP:

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. Demographic Info: Let us first understand the gender, age range, partner and dependent status of the customers.

1. Gender Distribution - About half of the customers in our data set are male while the other half are female. We can get that with this code.

```
# Create a bar plot
colors = ['#4D3425', '#E4512B']
ax = (telecom_cust['gender'].value_counts() * 100.0 / len(telecom_cust)).plot(
    kind='bar', stacked=True, rot=0, color=colors)

# Formatting y-axis as percentage
ax.yaxis.set_major_formatter(mtick.PercentFormatter())

# Set labels and title with white color
ax.set_ylabel('% Customers', color='white')
ax.set_xlabel('Gender', color='white')
ax.set_title('Gender Distribution', color='white')
ax.title.set_color('white')
ax.xaxis.label.set_color('white')
ax.yaxis.label.set_color('white')

# Set the color for y-axis ticks (male, female) and their labels/values to white
ax.tick_params(axis='y', colors='white')
ax.tick_params(axis='x', colors='white')

# Set individual bar labels with white color
totals = []

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

total = sum(totals)

for i in ax.patches:
    ax.text(i.get_x() + .15, i.get_height() - 3.5,
            str(round((i.get_height() / total), 1)) + '%',
            fontsize=12, color='white', weight='bold')
```

2. % Senior Citizens - There are only 16% of the customers who are senior citizens. Thus, most of our customers in the data are younger people. We can get with this code.

```
# Create a pie chart
ax = (telecom_cust['SeniorCitizen'].value_counts() * 100.0 / len(telecom_cust))\
    .plot.pie(autopct='%.1f%%', labels=['No', 'Yes'], figsize=(5, 5), fontsize=12)

# Formatting y-axis as percentage
ax.yaxis.set_major_formatter(mtick.PercentFormatter())

# Set labels and title with white color
ax.set_ylabel('Senior Citizens', fontsize=12, color='white')
ax.set_title('% of Senior Citizens', fontsize=12, color='white')
ax.title.set_color('white')

# Set the color for y-axis ticks and their labels/values to white
ax.tick_params(axis='y', colors='white')

# Set the color for pie chart labels to white
for text in ax.texts:
    text.set_color('white')
```

3. Partner and dependent status - About 50% of the customers have a partner, while only 30% of the total customers have dependents. We can get with this code.

```
# Create a bar plot
df2 = pd.melt(telecom_cust, id_vars=['customerID'], value_vars=['Dependents', 'Partner'])
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, color='white')  # Set y-axis label color to white
ax.set_xlabel('', color='white')  # Set x-axis label color to white
ax.set_title('% Customers with dependents and partners', size=14, color='white')  # Set title color to white
ax.legend(loc='center', prop={'size': 14})

# Set x and y ticks color to white
ax.tick_params(axis='x', colors='white')  # Set x-ticks color to white
ax.tick_params(axis='y', colors='white')  # Set y-ticks color to white

# Annotate the percentages on top of the bars with white color
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',  # Set the text color to white
                weight='bold',
                size=14)
```

Here, 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.

B. Customer Account Information: Now we 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 easy for the customers to stay/leave the telecom company. We can get that by this code.

```
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', color='white')  # Set y-axis label color to white
ax.set_xlabel('Tenure (months)', color='white')  # Set x-axis label color to white
ax.set_title('# of Customers by their tenure', color='white')  # Set title color to white

# Set x and y ticks color to white
ax.tick_params(axis='x', colors='white')  # Set x-ticks color to white
ax.tick_params(axis='y', colors='white')  # Set y-ticks color to white
```

2. Contracts: To understand the above graph, lets first look at the number of customers by different contracts. We can get with this code.

```
ax = telecom_cust['Contract'].value_counts().plot(kind='bar',
                                                  rot=0, width=0.3,
                                                  color='darkblue'
                                                 )
# Set y-axis label color to white
ax.set_ylabel('# of Customers', color='white')
# Set title color to white
ax.set_title('# of Customers by Contract Type', color='white')

# Set x and y ticks color to white
ax.tick_params(axis='x', colors='white')
ax.tick_params(axis='y', colors='white')
```

Using these we can see 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.

C. Let us now look at the distribution of various services used by customers.
We can get this by following code

```
services = ['PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity',
            'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies']

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=0)
    elif 3 <= i < 6:
        ax = telecom_cust[item].value_counts().plot(kind='bar', ax=axes[i-3, 1], rot=0)
    elif i < 9:
        ax = telecom_cust[item].value_counts().plot(kind='bar', ax=axes[i-6, 2], rot=0)
    ax.set_title(item, color='white')  # Set title color to white

    # Set x and y ticks color to white for all subplots
    ax.tick_params(axis='x', colors='white')  # Set x-ticks color to white
    ax.tick_params(axis='y', colors='white')  # Set y-ticks color to white

    # Set x-axis and y-axis label colors to white
    ax.set_xlabel('')  # Clear x-axis label
    ax.set_ylabel('')  # Clear y-axis label
```

D. Now let's take a quick look at the relation between monthly and total charges. We can get that by this code.

```
# Plotting scatter plot
scatter = telecom_cust[['MonthlyCharges', 'TotalCharges']].plot.scatter(x='MonthlyCharges',
                                                                        y='TotalCharges'
                                                                        )

# Set x-axis and y-axis label colors to white
scatter.set_xlabel('Monthly Charges', color='white')  # Set x-axis label color to white
scatter.set_ylabel('Total Charges', color='white')  # Set y-axis label color to white

# Set x and y ticks color to white
scatter.tick_params(axis='x', colors='white')  # Set x-ticks color to white
scatter.tick_params(axis='y', colors='white')  # Set y-ticks color to white
```

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

```
colors = ['#4D3425', '#E4512B']
ax = (telecom_cust['Churn'].value_counts() * 100.0 / len(telecom_cust)).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, color='white')  # Set y-axis label color to white
ax.set_xlabel('Churn', size=14, color='white')  # Set x-axis label color to white
ax.set_title('Churn Rate', size=14, color='white')  # Set title color to white

# Set the color of x and y ticks to white
ax.tick_params(axis='x', colors='white')  # Set x-axis tick color to white
ax.tick_params(axis='y', colors='white')  # Set y-axis tick color to white

# 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 labels using the 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)) + '%',
            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 skewness could lead to a lot of false negatives. We will see in the modelling section on how to avoid skewness in the data.

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. We can get this by this code

```
sns.boxplot(x = telecom_cust.Churn, y = telecom_cust.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. We can get that by this code

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

iii. Churn by Seniority: Senior Citizens have almost double the churn rate than younger population. We can get that by this code

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

iv. Churn by Monthly Charges: Higher % of customers churn when the monthly charges are high. We can get that by this code

```
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')
```

Churn by Total Charges: It seems that there is higher churn when the total charges are lower. We can get this by this code.

6. RESULTS AND ANALYSIS:

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

We developed Logistic Regression, Random Forest, SVM.

1. Logistic Regression:
   Based on the problem statement, we need a predictive model that can do a binary classification or predict Yes/No or 1/0 type of output variable. One predictive model commonly implemented for binary classification and prediction of binary outcome is Logistic Regression. Logistic regression is a binary classification algorithm belonging to the generalized linear regression model. It can also be used to solve problems with more than 2 classes. It is possible to use logistic regression to create a model using the customer churn data and use it to predict if a particular customer of a set of customers will discontinue the service.

We implemented the following code for our project need.

```
# 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
```

```
/opt/conda/lib/python3.6/site-packages/sklearn/preprocessing/data.py:323: DataConversionWarni
ng: Data with input dtype uint8, int64, float64 were all converted to float64 by MinMaxScale
r.
  return self.partial_fit(X, y)
```

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.

```
# 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_state=101)
```

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

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

```
0.8075829383886256
```

```
# 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'))
```

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

2. Random Forest:
   Based on the problem summary, we need a predictive model that can predictively modeling on the monthly cost service or monthly payments for the customers. There are several categorical fields in the customer churn data, and there are several numerical fields too. It has information such as if the customer is using internet service, phone service, multiple services. Usually, the payment is associated with a particular service, say, if you get phone service you will pay a fixed amount per month and if you upgrade with the internet, the cost will increase by a certain amount. Other variables like tenure, age, bandwidth usage also may impact monthly payments. Based on decisions such as, if the customer has fiber optic internet, or if the customer streams tv or movies, it should be possible to estimate the monthly payment.

```
from sklearn.ensemble import RandomForestClassifier
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)

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

```
0.8088130774697939
```

We can get graph for the same with this code:

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

3. Support Vector Machine (SVM):

To improve the prediction abilities of machine learning methods, a support vector machine (SVM) on structural risk minimization was applied to customer churn prediction. Researching customer churn prediction cases both in home and foreign carries, the method was compared with artificial neural network, decision tree, logistic regression, and naive bayesian classifier. It is found that the method enjoys the best accuracy rate, hit rate, covering rate, and lift coefficient, and therefore, provides an effective measurement for customer churn prediction.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=99)
```

```
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)
```

```
0.820184790334044
```

```
# Create the Confusion matrix
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,preds))
```

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

With SVM I was able to increase the accuracy to up to 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.

CODE-SNIPPETS:

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

5 rows × 21 columns

```
telecom_cust.columns.values
```

Python

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

```
# Checking the data types of all the columns
telecom_cust.dtypes
```

```
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
```

```
# Converting Total Charges to a numerical data type.
telecom_cust.TotalCharges = pd.to_numeric(telecom_cust.TotalCharges, errors='coerce')
telecom_cust.isnull().sum()
```

```
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
```

7. DISCUSSION:

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 probability of Churn. Lastly, total charges, monthly contracts, fiber optic internet services and seniority can lead to higher churn rates. This is interesting because although fiber optic services are faster, customers are likely to churn because of it. I think we need to explore more to better understand why this is happening. 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. With SVM I was able to increase the accuracy to up to 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.That is all the methods that we are using for this project and all works very differently from each other. SVM is the best of all the methods a it gives 82% accuracy. Random Forest gave only 80% accuracy. Logistic Regression also have 80% accuracy.

8. PREVIOUS EXPERIMENT RESULTS AND COMPARISON:

Brandusoiu I, Toderean G, Ha B. Methods for churn prediction in the prepaid mobile telecommunications industry. In: International conference on communications. Three machine learning algorithms were used: Neural Networks, Support Vector Machine, and Bayes Networks to predict churn factor. The author used AUC to measure the performance of the algorithms. The AUC values were 99.10%, 99.55% and 99.70%. Compared to the exceptionally high AUC values of 99.10% to 99.70% achieved using Neural Networks, SVM, and Bayes Networks in the study by Brandusoiu et alOur project's maximum accuracy of 82% with SVM suggests a more typical real-world scenario, possibly indicating a need for advanced model tuning or exploring additional features and algorithms to enhance prediction accuracy.

He Y, He Z, Zhang D. A study on prediction of customer churn in fixed communication network based on data mining. In: Sixth international conference on fuzzy systems and knowledge discovery, vol. 1. 2009. proposed a model for prediction based on the Neural Network algorithm in order to solve the problem of customer churn in a large Chinese telecom company which contains about 5.23 million customers. The prediction accuracy standard was the overall accuracy rate, and reached 91.1%. Our project, which reached a top accuracy of 82% with SVM, shows a good performance but is somewhat lower than the 91.1% reported in the fixed network context. This difference might be influenced by the complexity of the dataset, the specific challenges of the mobile versus fixed network sectors, or the effectiveness of the Neural Network approach in handling large datasets like the 5.23 million customers in the cited study.

9. CONCLUSION:

The telecom churn prediction project showcases the pivotal role of machine learning techniques, including Logistic Regression, Random Forest, and Support Vector Machine (SVM), in tackling the persistent challenge of customer churn within the telecommunications industry. By harnessing advanced analytical tools, the project aims to provide telecom companies with actionable insights to mitigate churn and enhance customer retention.

Among the machine learning models evaluated, SVM emerged as the top performer in predictive accuracy, demonstrating its efficacy in identifying potential churners. This highlights the importance of leveraging sophisticated algorithms capable of capturing intricate patterns in telecom data to achieve superior prediction performance.

Through extensive analysis, the project identified key predictors of churn, such as contract length, internet service type, and payment methods. These insights offer valuable guidance for telecom operators seeking to implement targeted intervention strategies aimed at retaining at-risk customers. For instance, understanding that customers with shorter contract lengths or certain internet service types are more prone to churn allows companies to tailor

retention efforts, such as offering incentives for longer-term contracts or improving service quality for specific internet offerings.

10. REFERENCES:

I. Nguyen, T., Ngo, D., Nguyen, T., Nguyen, T., & Le, N. (2019). Customer Churn Prediction in Telecommunication Industry: A Review and Future Directions. 2020 12th International Conference on Knowledge and Systems Engineering (KSE). doi: 10.1109/kse48636.2020.9209870.

II. Sharma, N., & Singh, R. (2020). Telecom Customer Churn Prediction Using Machine Learning Algorithms: A Systematic Review and Meta-Analysis. International Journal of Advances in Scientific Research and Engineering (IJASRE), 6(2), 36-43.

III. Khan, S., Haq, M. U., Shah, S. A., & Ahmad, F. (2018). A Review on Telecom Customer Churn Prediction Using Data Mining Techniques. 2018 4th International Conference on Computer and Technology Applications (ICCTA). doi: 10.1109/iccta.2018.8662874

IV. Li, Y., & Sun, L. (2017). Predicting Telecom Customer Churn with Advanced Machine Learning Techniques: A Comparative Study. 2017 IEEE International Conference on Big Data (Big Data). doi: 10.1109/bigdata.2017.8258235

V. Chen, S., Lin, W., & Xie, X. (2016). Churn Prediction in Telecom Using Machine Learning in Big Data Platform. 2016 2nd IEEE International Conference on Computer and Communications (ICCC). doi: 10.1109/compcomm.2016.7924978

VI. Han, S., & Han, S. (2019). Telecommunications Churn Prediction Using Machine Learning Algorithms. 2019 5th International Conference on Control, Automation and Robotics (ICCAR). doi: 10.1109/iccar.2019.8813574

VII. Wang, Y., & Wang, Z. (2018). Customer Churn Prediction Model of Telecommunications Industry Based on Data Mining. 2018 13th International Conference on Computer Science & Education (ICCSE). doi: 10.1109/iccse.2018.8468546

VIII. Kim, J., & Kim, J. (2017). Prediction Model for Churn Analysis in Mobile Telecommunication Industry. 2017 6th International Conference on Software and Computer Applications (ICSCA). doi: 10.1145/3124690.3124744

IX. Gupta, A., & Chaudhary, A. (2016). Customer Churn Prediction in Telecom Industry: A Data Mining Approach. 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). doi: 10.1109/indiacom.2016.7729659

X. Zhang, Z., & Li, H. (2015). Predicting Customer Churn for Telecom Company Using Machine Learning Techniques. 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). doi: 10.1109/fskd.2015.7382237