



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 Issue: XI Month of publication: November 2021

DOI: <https://doi.org/10.22214/ijraset.2021.38859>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Business Use of Data Science – Churn Prediction

Themaliparambil Naveed Abbas
Undergraduate Student at Delhi University

I. INTRODUCTION

Churn prediction is probably one of the most important applications of data science in the business sector. The thing that makes it popular is that its effect are more tangible to comprehend and it plays a major factor in the overall profits earned by the business.

A. What is Churn Prediction?

Churn quantifies the number of customers who have left your brand by cancelling their subscriptions or stopping paying for your services. This is bad news for any business as it costs five times as much to attract a new customer as it does to keep an existing one. A high customer churn rate will hit your company's finances hard. By leveraging advanced artificial intelligence techniques like machine learning (ML), you will be able to anticipate potential churners who are about to abandon your services.

B. Why is it important?

You probably already have more customer data than you know. By using this data, you are able to identify behaviour patterns of customers who are likely to churn. This knowledge will enable you to segment those customers and take the appropriate measures to win them back.

The simpler way to calculate churn rate is to divide the number of customers lost during a given time interval by the number of active customers at the beginning of the period. For example, if you got 1000 customers and lost 50 last month, then your monthly churn rate is 5 percent.

II. DATA EXPLORATION

I have used the Telecom Customer Churn dataset from kaggle.

A. Important Required Dependencies

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.metrics import accuracy_score, confusion_matrix

from sklearn.model_selection import RandomizedSearchCV

%matplotlib inline
```

B. Loading the Dataset

```
df=pd.read_csv('./input/telco-customer-churn/WA_Fn-UseC_-Telco-Customer-Churn.csv')
df.info()
```

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

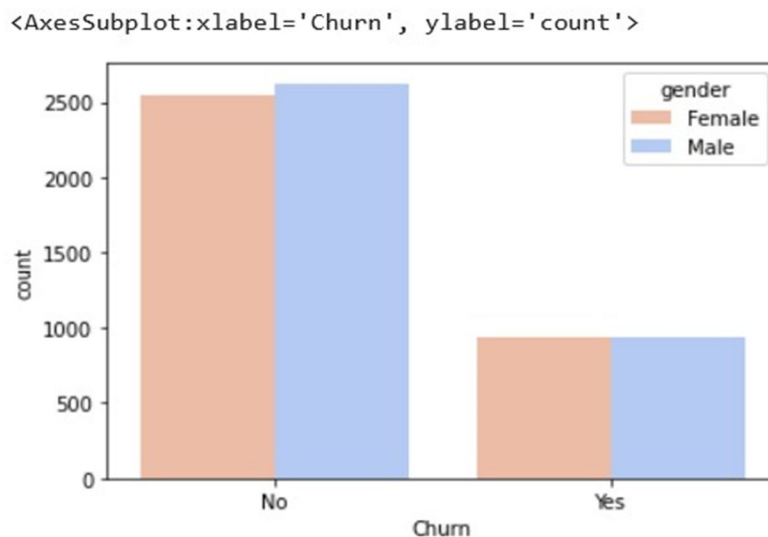
As we can see there are a total of 20 columns in our data set. Out of these,3 are of numeric data type.

C. Data Visualization

We need to explore the data to find some patterns.

For the columns in the dataset which are non-numerical, we can use a seaborn countplot to plot a graph against the Churn column.

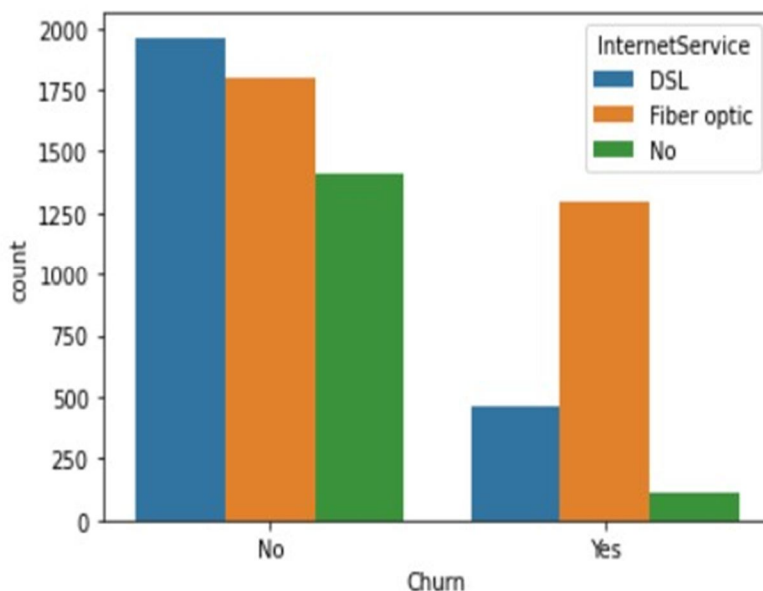
```
sns.countplot(x='Churn ',data=df,hue='gender',palette="coolwarm_r")
```



From the above graph we can see that gender is not a contributing factor for customer churn in this dataset as the number of both the genders, that have or haven't churned are almost the same.

```
sns.countplot(x='Churn',data=df,hue='InternetService')
```

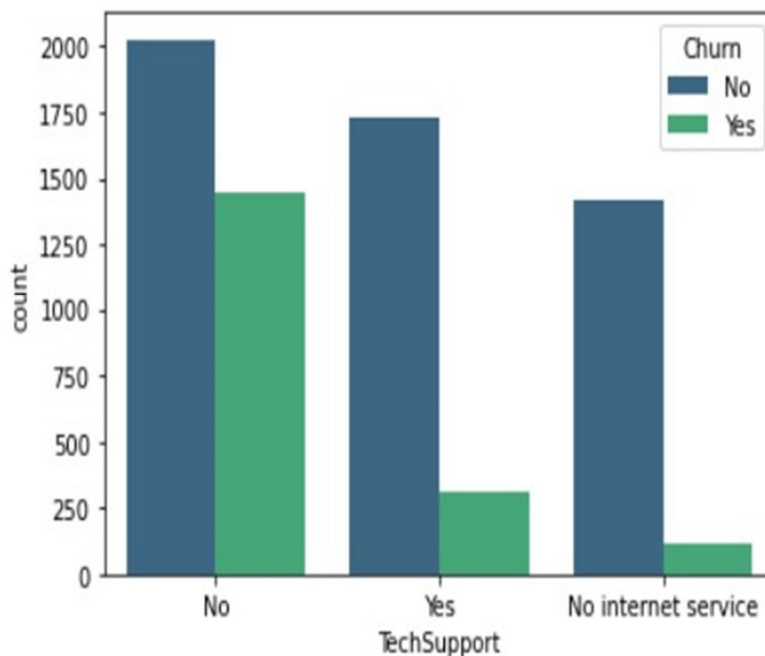
```
<AxesSubplot:xlabel='Churn', ylabel='count'>
```



We can see that people using Fiber-optic services have a higher churn percentage. This shows that the company needs to improve their Fiber-optic service.

```
sns.countplot(x='TechSupport',data=df,hue='Churn',palette='viridis')
```

```
<AxesSubplot:xlabel='TechSupport', ylabel='count'>
```



Those customers who don't have tech support have churned more, which is pretty self-explanatory. This also highlights the fact that the tech support provided by the company is up to the mark.

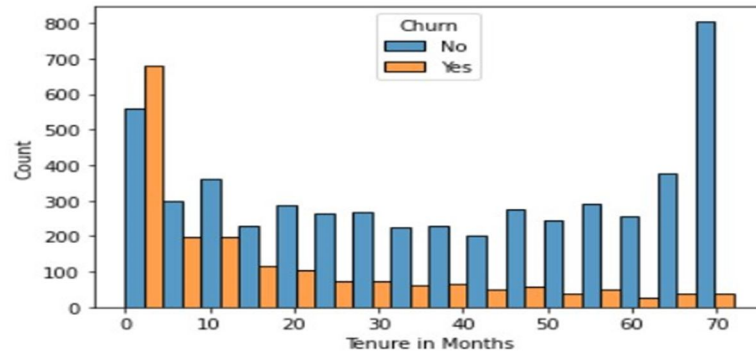
D. Tackling Numeric Data

Now let's look at some numerical value to see how to tackle them.

```
ax=sns.histplot(x='tenure',hue='Churn',data=df,multiple='dodge')
```

```
ax.set(xlabel="Tenure in months",ylabel="Count")
```

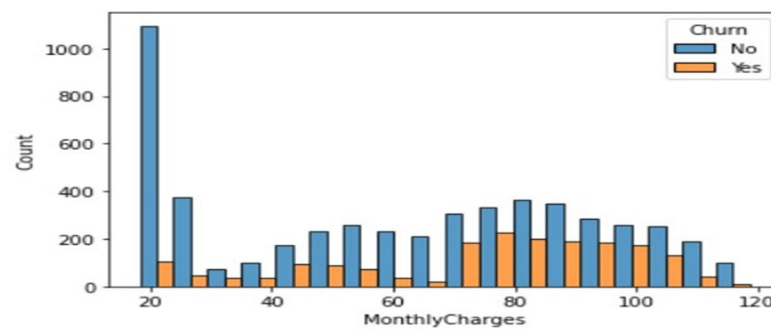
```
[Text(0.5, 0, 'Tenure in Months'), Text(0, 0.5, 'Count')]
```



The churn amount is higher in the initial 5 months, which is usually the time when the new customer try out the service and decide whether to continue or cancel. This pretty much can be attributed to the uncertainty in the customer's mind.

```
sns.histplot(x='MonthlyCharges',hue='Churn',data=df,multiple='dodge')
```

```
<AxesSubplot:xlabel='MonthlyCharges', ylabel='Count'>
```



We cannot see a definite pattern in this, but we can conclude that those who have monthly charges as high as 100 dollars have chosen not to churn. This indicates that the company has done well to retain high paying customers. Similarly, we can evaluate the other parameters as well and draw meaningful conclusions as to how the company should improve customer retention.

E. Data Preparation

We need to make sure that the data is in the right form to be used for prediction. Machine Learning models do not work well with categorical inputs. So, we convert the categorical variables in our data set to numerical values by using one-hot encoding.

```
df_copy=pd.get_dummies(df_copy,drop_first=True)
```

```
df_copy.head()
```

SeniorCitizen	Partner	Dependents	tenure	PhoneService	PaperlessBilling	MonthlyCharges	TotalCharges	Churn	gender_Male	...	TechSupport_Yes	StreamingTV_No internet service
0	1	0	1	0	1	29.85	29.85	0	0	...	0	0
0	0	0	34	1	0	56.95	1889.50	0	1	...	0	0
0	0	0	2	1	1	53.85	108.15	1	1	...	0	0
0	0	0	45	0	0	42.30	1840.75	0	1	...	1	0
0	0	0	2	1	1	70.70	151.65	1	0	...	0	0

The drop_first parameter helps in reducing the number of columns and hence prevents co-relation between the variables. Hence, it is set to True.

F. Scaling

Scaling data is important to increase prediction accuracy.

```
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
X.head()
```

SeniorCitizen	Partner	Dependents	tenure	PhoneService	PaperlessBilling	MonthlyCharges	TotalCharges	gender_Male	MultipleLines_No phone service	...	TechSupport_Yes
0.0	1.0	0.0	0.013889	0.0	1.0	0.115423	0.001275	0.0	1.0	...	0.0
0.0	0.0	0.0	0.472222	1.0	0.0	0.385075	0.215867	1.0	0.0	...	0.0
0.0	0.0	0.0	0.027778	1.0	1.0	0.354229	0.010310	1.0	0.0	...	0.0
0.0	0.0	0.0	0.625000	0.0	0.0	0.239303	0.210241	1.0	1.0	...	1.0
0.0	0.0	0.0	0.027778	1.0	1.0	0.521891	0.015330	0.0	0.0	...	0.0

III. PREDICTION

First of all, let's split the data into 2 datasets;

Training and testing.

```
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=41)
```

Now we can start with different algorithms for prediction.

A. Logistic Regression

```
from sklearn.linear_model import LogisticRegression
logreg=LogisticRegression()
logreg.fit(X_train,y_train)
prediction_logreg=logreg.predict(X_test)
print(accuracy_score(y_test,prediction_logreg))
Accuracy Score LogReg:0.7950780880265026
```

B. Random Forest using Random CV

From sklearn.ensemble import RandomForestClassifier

```
rf_c=RandomForestClassifier()
```

```
param_grid={'n_estimators':[int(x) for x in np.linspace(start=200,stop=1200,num=11)]
            'max_features':['auto','sqrt'],
            'max_depth':[int(x) for x in np.linspace(start=10,stop=100,num=11)],
            'min_samples_leaf':[1,2,3,5],
            'min_samples_split':[2,5,10,15]}
```

```
random_cv=RandomSearchCV(rf_c,param_grid,cv=3,verbose=2,random_state=42)
random_cv.fit(X_train,y_train)
best_random=random_cv.best_estimator_
prediction_cv=best_random.predict(X_test)
print(accuracy_score(y_test,prediction_cv))
```

Accuracy Score RF:0.8021769995267393



C. XGBoost

```
from xgboost import XGBClassifier
xgb_model=XGBClassifier()
xgb_model.fit(X_train,y_train)
prediction_xgb=xgb_model.predict(X_test)
print(accuracy_score(y_test,prediction_xgb))
Accuracy Score XGB:0.7875059157595835
```

From the above accuracy scores, we see that Random Forest clearly outperforms Logistic Regression and XGBoost. By using RandomCV, the accuracy is further improved.

Let's see the confusion matrix of Random Forest

```
print(confusion_matrix(y_test,prediction_cv))
```

Confusion Matrix:

```
[[1410  144]
 [ 274  285]]
```

It shows that our model needs to improve the False Negative classifications.

IV. CONCLUSION

We went through the various tasks involved in churn prediction in this article. It is important to note that finding patterns in Exploratory Data Analysis (EDA) is as important as the final prediction itself.

A Churn prediction task remains unfinished if the data patterns are not found in EDA. Most people can do the prediction part but struggle with data visualization and conveying the findings in an interesting way.

This skill is not only limited to Churn prediction but will also help you in the solving of the usual data science problems.

REFERENCES

- [1] Soham Naik (August,2021) Example of Churn Prediction from the given dataset.
- [2] Telecom Customer Dataset from kaggle.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)