PROJECT FINAL REPORT

TELECOM USERS

Elif Karataş

2016510040

22.06.2021

Contents

Description of the Dataset

This data set contains data of customers belonging to a telecommunications company.

This data is used by the telecommunications company to retain customers because it costs less to retain the existing customer than to attract and register a new customer.

According to this data, we can identify the customer who will leave on time and try to keep the customer who wants to leave the operator. Based on the data about the services used by the customer, we can try to change the decision to leave the operator, making a special offer for him/her. This will make the retention task easier to accomplish than the task of attracting new users we don't know anything about yet.

The data includes information about the demographic characteristics of 5985 users, the services they use, the duration of the operator's services, the payment method and the amount of payment.

The task is to analyze the data and predict the churn of users.

Tuples

- customerID - customer id
- gender - client gender (male / female)
- SeniorCitizen - is the client retired (1, 0)
- Partner - is the client married (Yes, No)
- tenure - how many months a person has been a client of the company
- PhoneService - is the telephone service connected (Yes, No)
- MultipleLines - are multiple phone lines connected (Yes, No, No phone service)
- InternetService - client's Internet service provider (DSL, Fiber optic, No)
- OnlineSecurity - is the online security service connected (Yes, No, No internet service)
- OnlineBackup - is the online backup service activated (Yes, No, No internet service)
- DeviceProtection - does the client have equipment insurance (Yes, No, No internet service)
- TechSupport - is the technical support service connected (Yes, No, No internet service)
- StreamingTV - is the streaming TV service connected (Yes, No, No internet service)
- StreamingMovies - is the streaming cinema service activated (Yes, No, No internet service)
- Contract - type of customer contract (Month-to-month, One year, Two year)

- PaperlessBilling - whether the client uses paperless billing (Yes, No)
- PaymentMethod - payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))
- MonthlyCharges - current monthly payment
- TotalCharges - the total amount that the client paid for the services for the entire time
- Churn - whether there was a churn (Yes or No)

Data Types of the Dataset

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

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5986 entries, 0 to 5985
Data columns (total 20 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   gender            5986 non-null   object
 1   SeniorCitizen     5986 non-null   int64
 2   Partner           5986 non-null   object
 3   Dependents        5986 non-null   object
 4   tenure            5986 non-null   int64
 5   PhoneService      5986 non-null   object
 6   MultipleLines     5986 non-null   object
 7   InternetService   5986 non-null   object
 8   OnlineSecurity    5986 non-null   object
 9   OnlineBackup      5986 non-null   object
 10  DeviceProtection  5986 non-null   object
 11  TechSupport       5986 non-null   object
 12  StreamingTV       5986 non-null   object
 13  StreamingMovies   5986 non-null   object
 14  Contract          5986 non-null   object
 15  PaperlessBilling  5986 non-null   object
 16  PaymentMethod     5986 non-null   object
 17  MonthlyCharges    5986 non-null   float64
 18  TotalCharges      5986 non-null   object
 19  Churn             5986 non-null   object
dtypes: float64(1), int64(2), object(17)
memory usage: 935.4+ KB
```

Some Statistical Information of the Dataset

Gender

Male 51% - Female 49%

Senior Citizen

Mean 0.16 - Std. Deviation 0.37

Partner

True 2904 49% - False 3082 51%

Dependents

True 1791 30% - False 4195 70%

Tenure

Mean 32.5 - Std. Deviation 24.5

Phone Service

True 5396 90% - False 590 10%

Multiple Lines

No 48% - Yes 43%

Internet Service

Fiber optic 44% - DSL 35% - Other 22%

Online Security

No 44% - Yes 35% - Other 22%

OnlineBackup

No 44% - Yes 35% - Other 22%

Device Protection

No 44% - Yes 34% - Other 22%

Tech Support

No 49% - Yes 29% - Other 22%

StreamingTV

No 40% - Yes 39% - Other 22%

StreamingMovies

No 39% - Yes3 9% - Other 22%

Contract

Month-to-month 55% - Two year 24% - Other 21%

Paperless Billing

True 3528 59% - False 2458 41%

Payment Method

Electronic check 34% - Mailed check 23% - Other 44%

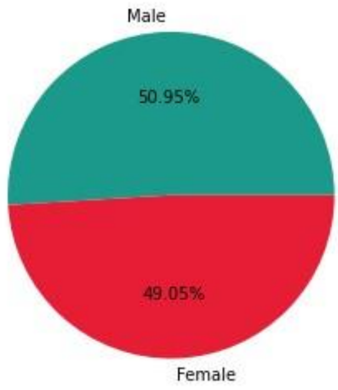
Churn

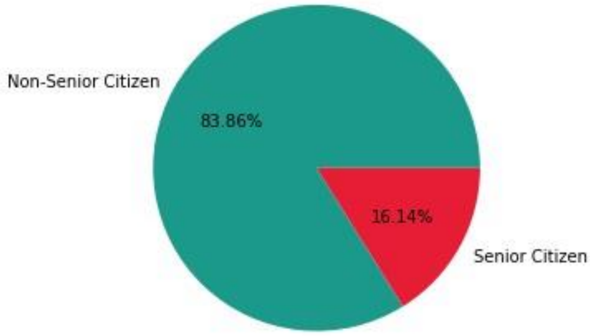
True 1587 27% - False 4399 73%

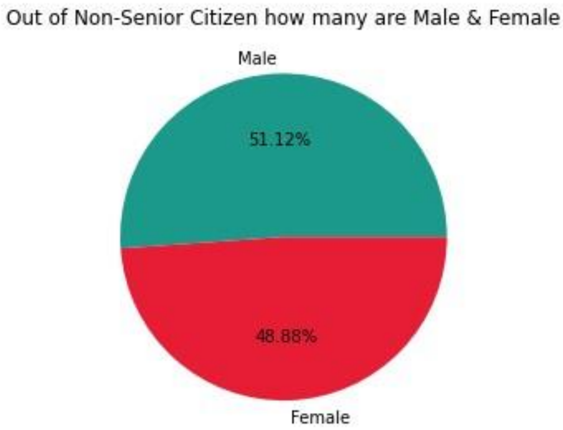
Also, there is no missing values.

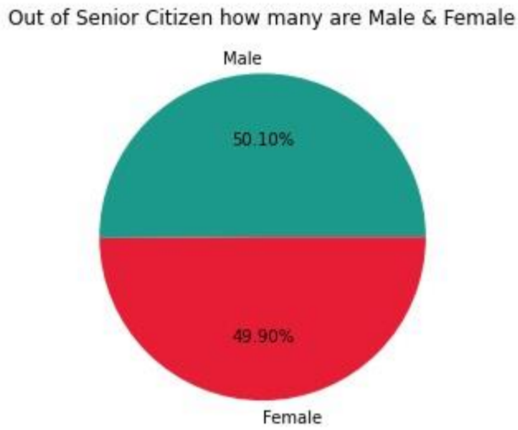
```
In [99]: df.isna().sum()

Out[99]: gender              0
         SeniorCitizen       0
         Partner             0
         Dependents          0
         tenure              0
         PhoneService        0
         MultipleLines       0
         InternetService     0
         OnlineSecurity      0
         OnlineBackup        0
         DeviceProtection    0
         TechSupport         0
         StreamingTV         0
         StreamingMovies     0
         Contract            0
         PaperlessBilling    0
         PaymentMethod       0
         MonthlyCharges      0
         TotalCharges        0
         Churn               0
         dtype: int64
```

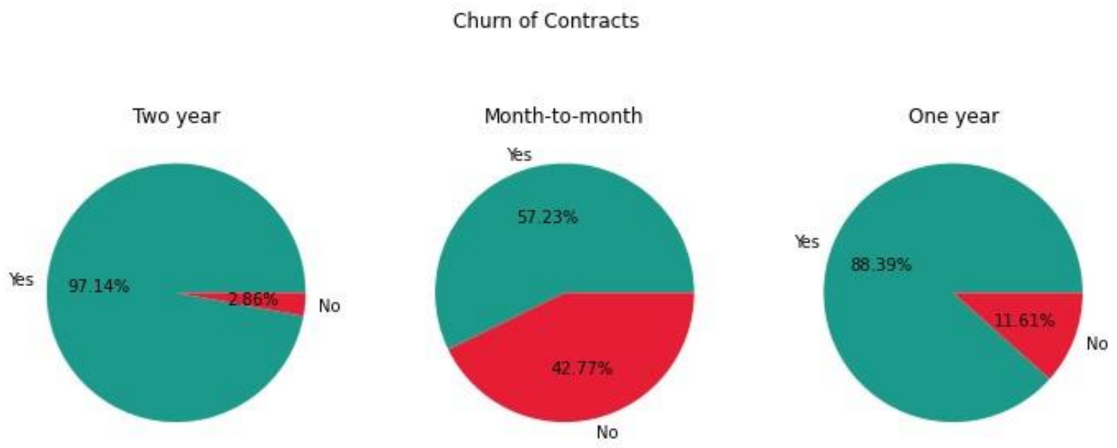
Some Pie Charts of the Dataset

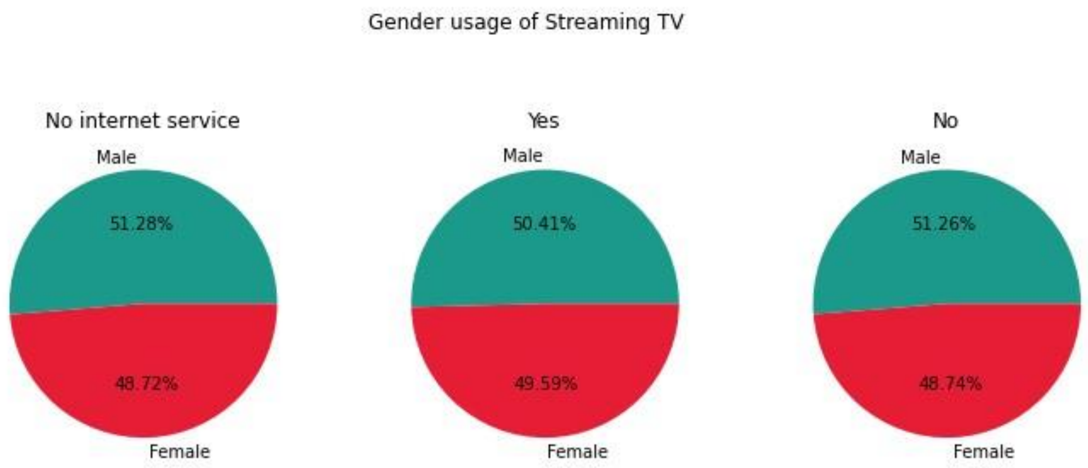


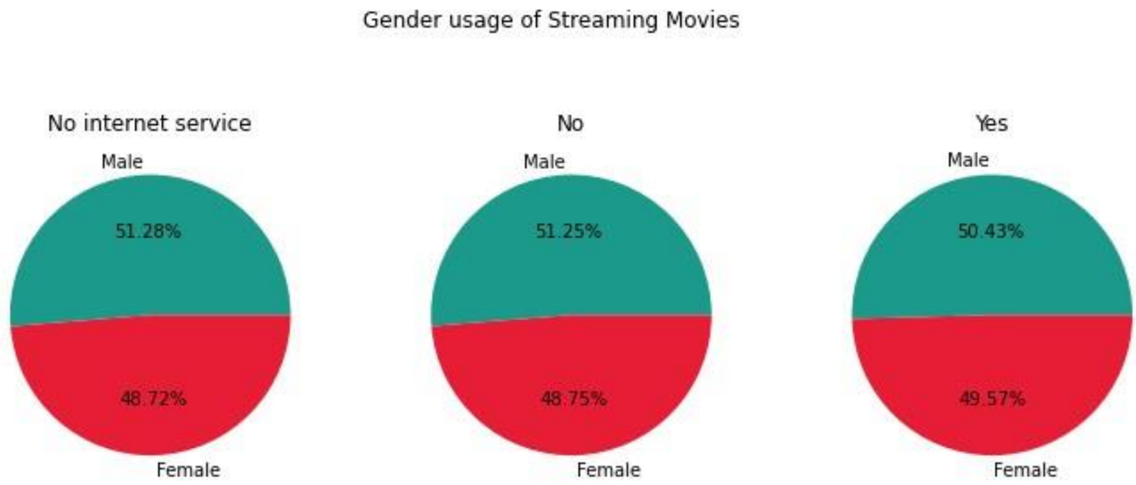






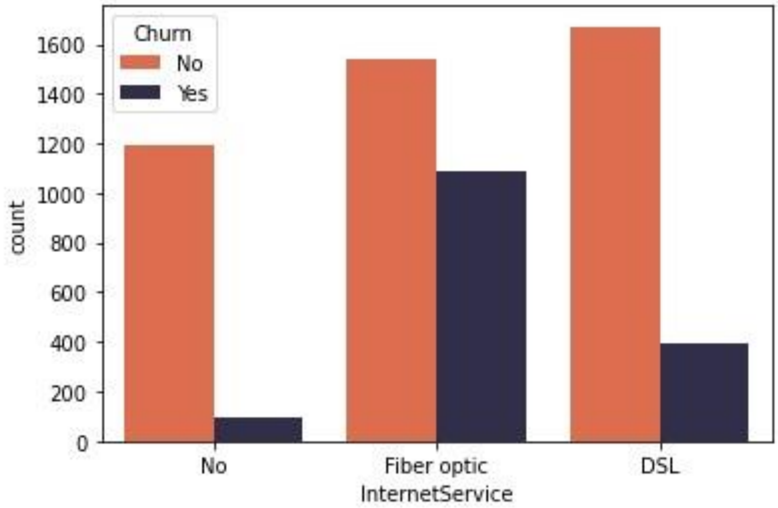



Gender usage of Streaming TV
No internet service
Male
51.28%
48.72%
Female
Yes
Male
50.41%
49.59%
Female
No
Male
51.26%
48.74%
Female

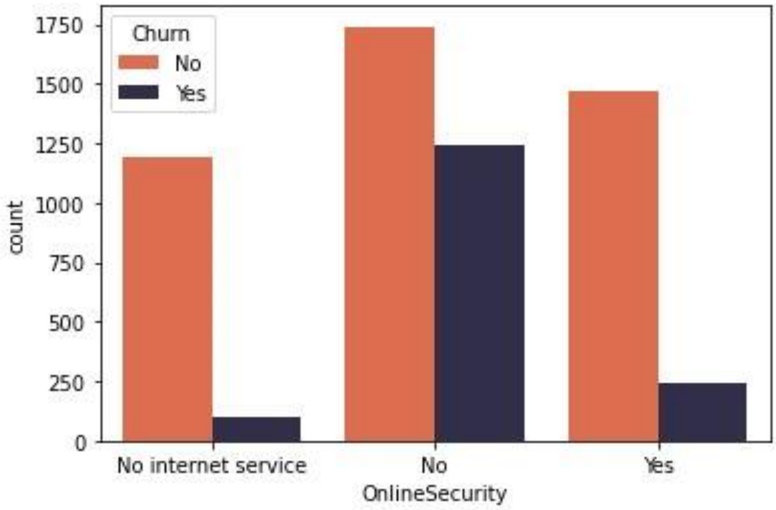

Gender usage of Streaming Movies
No internet service
Male
51.28%
48.72%
Female
No
Male
51.25%
48.75%
Female
Yes
Male
50.43%
49.57%
Female

Some Bar Charts of the Dataset

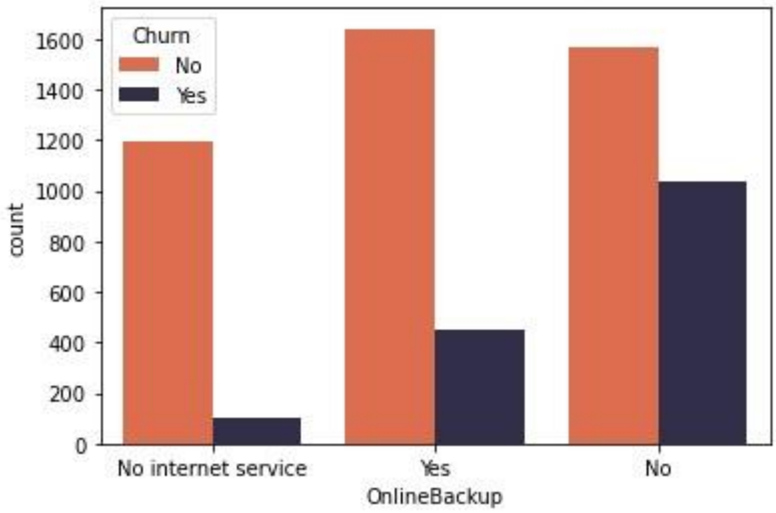
Out[64]: <AxesSubplot:xlabel='InternetService', ylabel='count'>



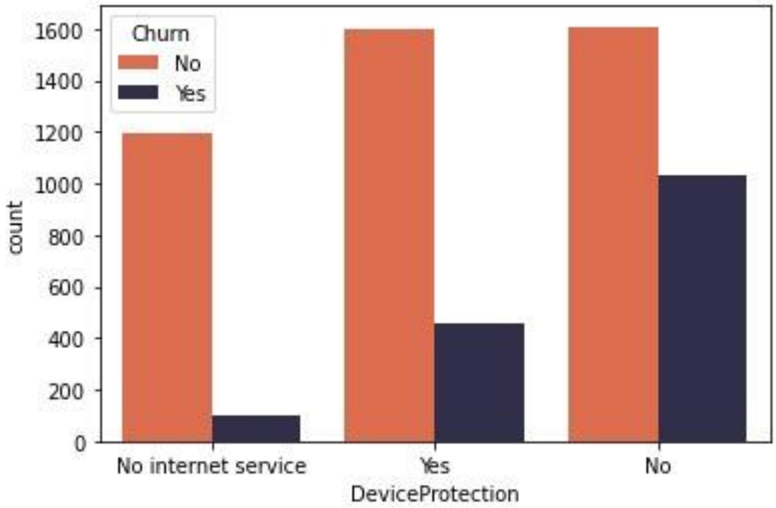
Out[65]: <AxesSubplot:xlabel='OnlineSecurity', ylabel='count'>



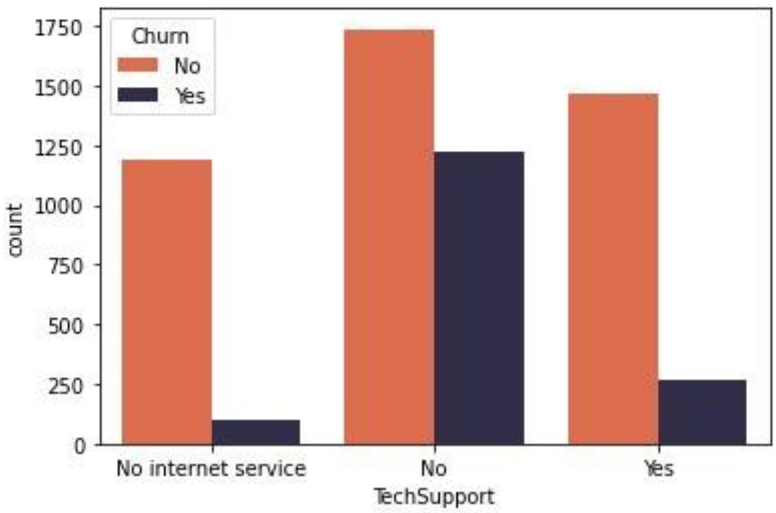
Out[66]: <AxesSubplot:xlabel='OnlineBackup', ylabel='count'>



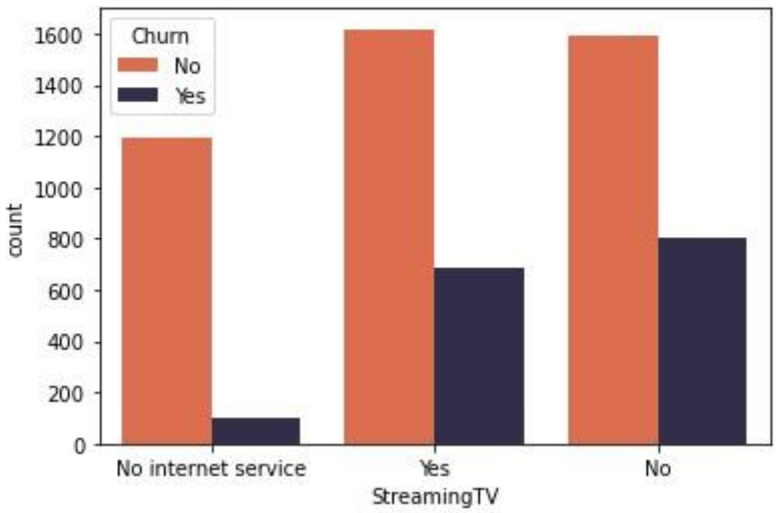
Out[67]: <AxesSubplot:xlabel='DeviceProtection', ylabel='count'>



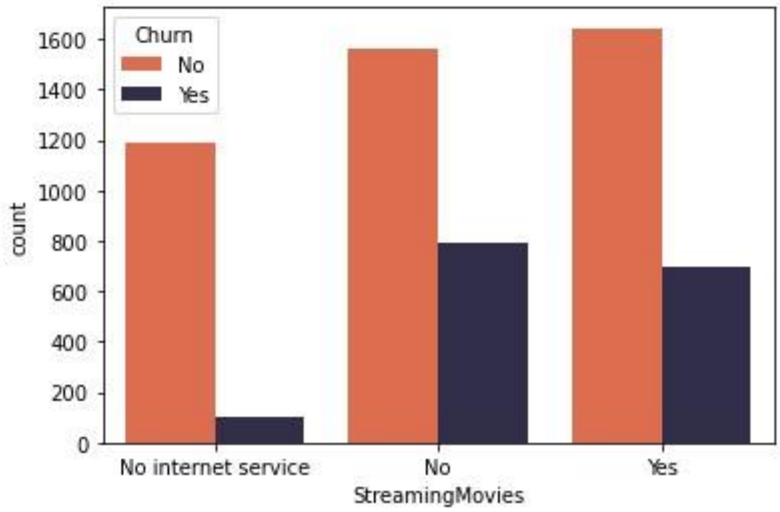
Out[68]: <AxesSubplot:xlabel='TechSupport', ylabel='count'>



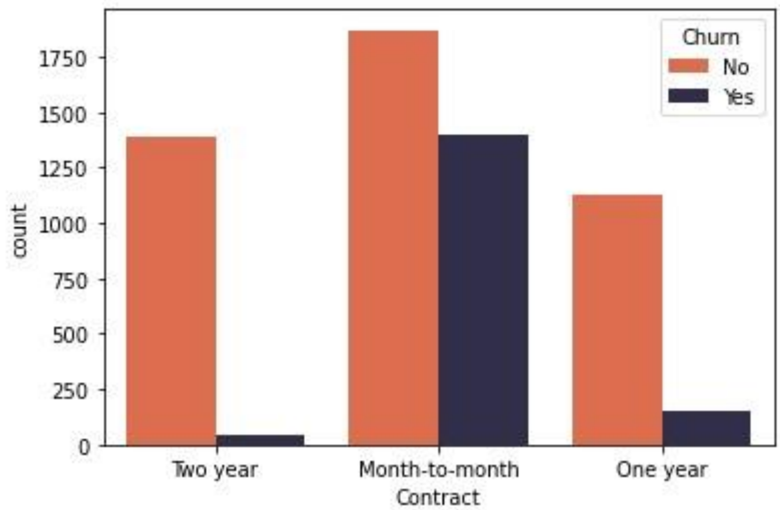
Out[69]: <AxesSubplot:xlabel='StreamingTV', ylabel='count'>



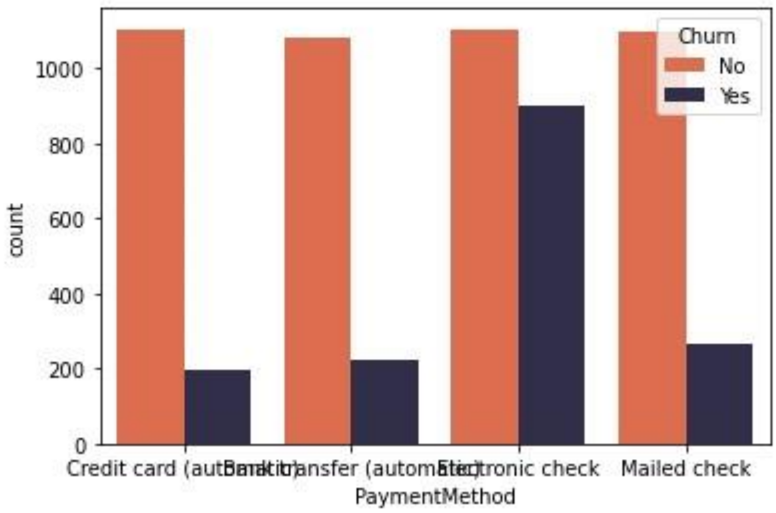
Out[70]: <AxesSubplot:xlabel='StreamingMovies', ylabel='count'>



Out[85]: <AxesSubplot:xlabel='Contract', ylabel='count'>

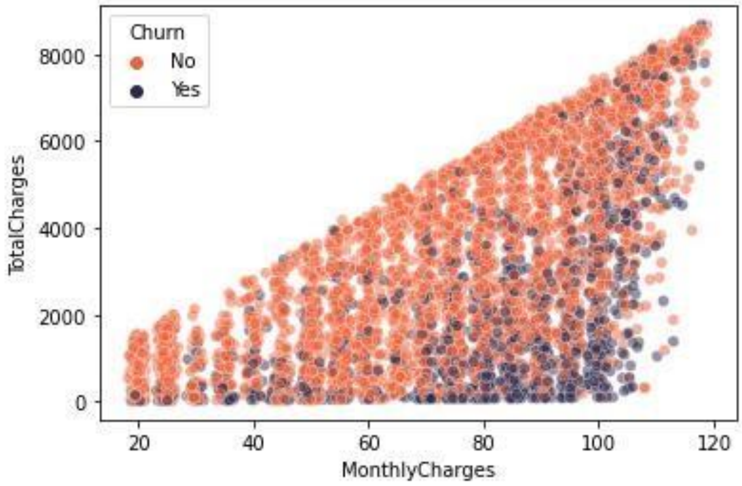


Out[86]: <AxesSubplot:xlabel='PaymentMethod', ylabel='count'>



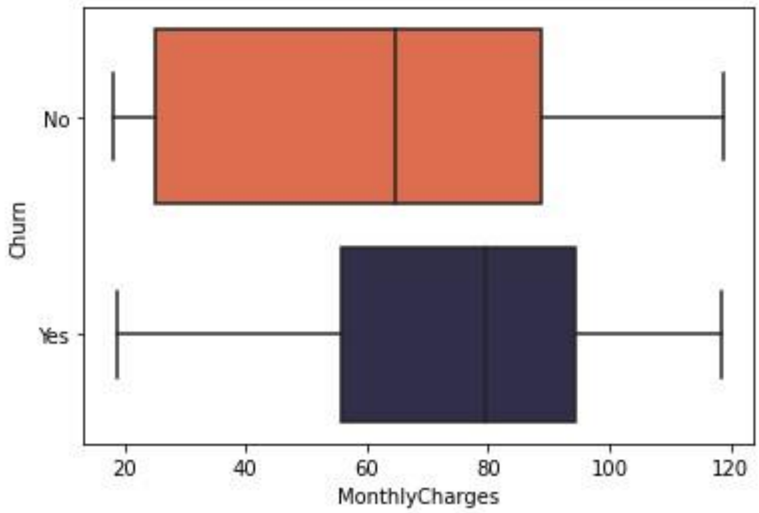
Scatter Plot for Charges

```
Out[88]: <AxesSubplot:xlabel='MonthlyCharges', ylabel='TotalCharges'>
```



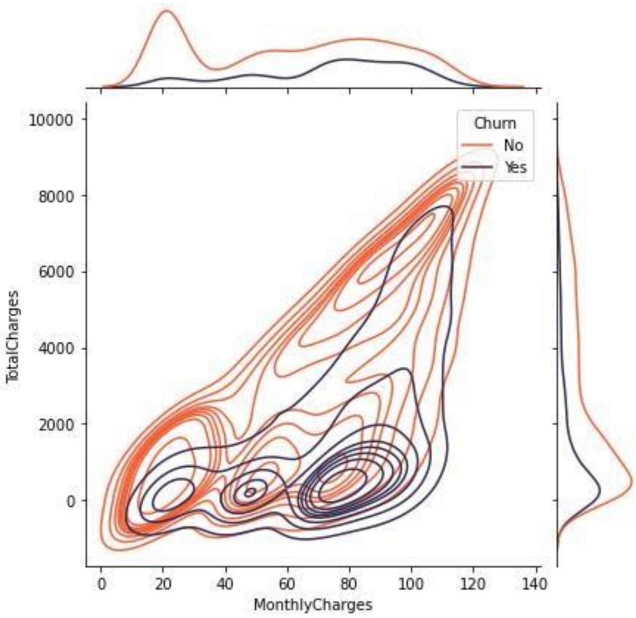
Box Plot for Monthly Charges

```
Out[89]: <AxesSubplot:xlabel='MonthlyCharges', ylabel='Churn'>
```

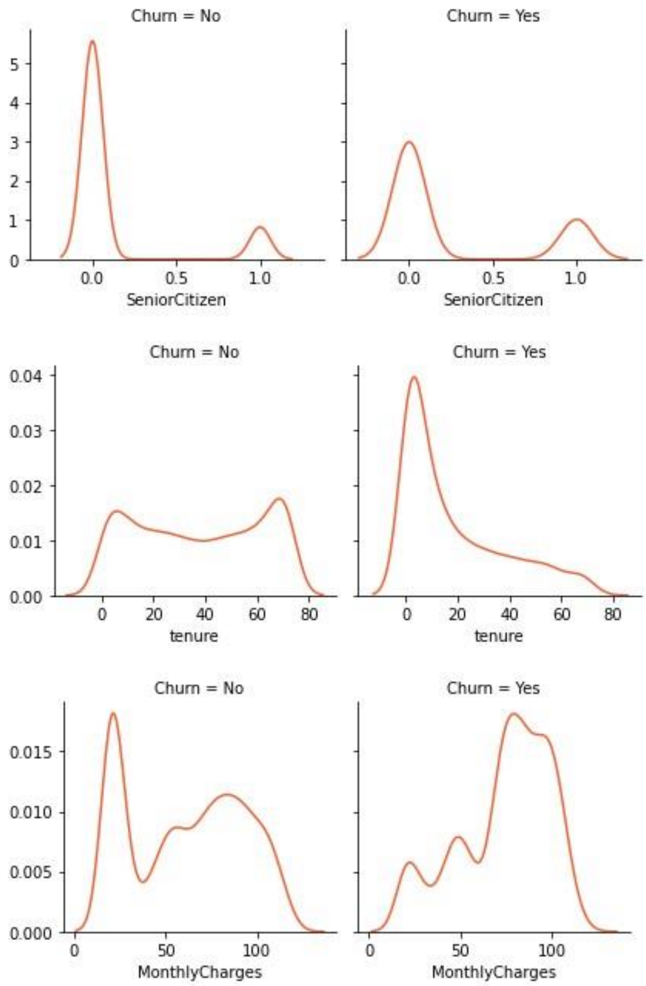


Joint Plot for Charges

Out[90]: <seaborn.axisgrid.JointGrid at 0x20b032324c0>

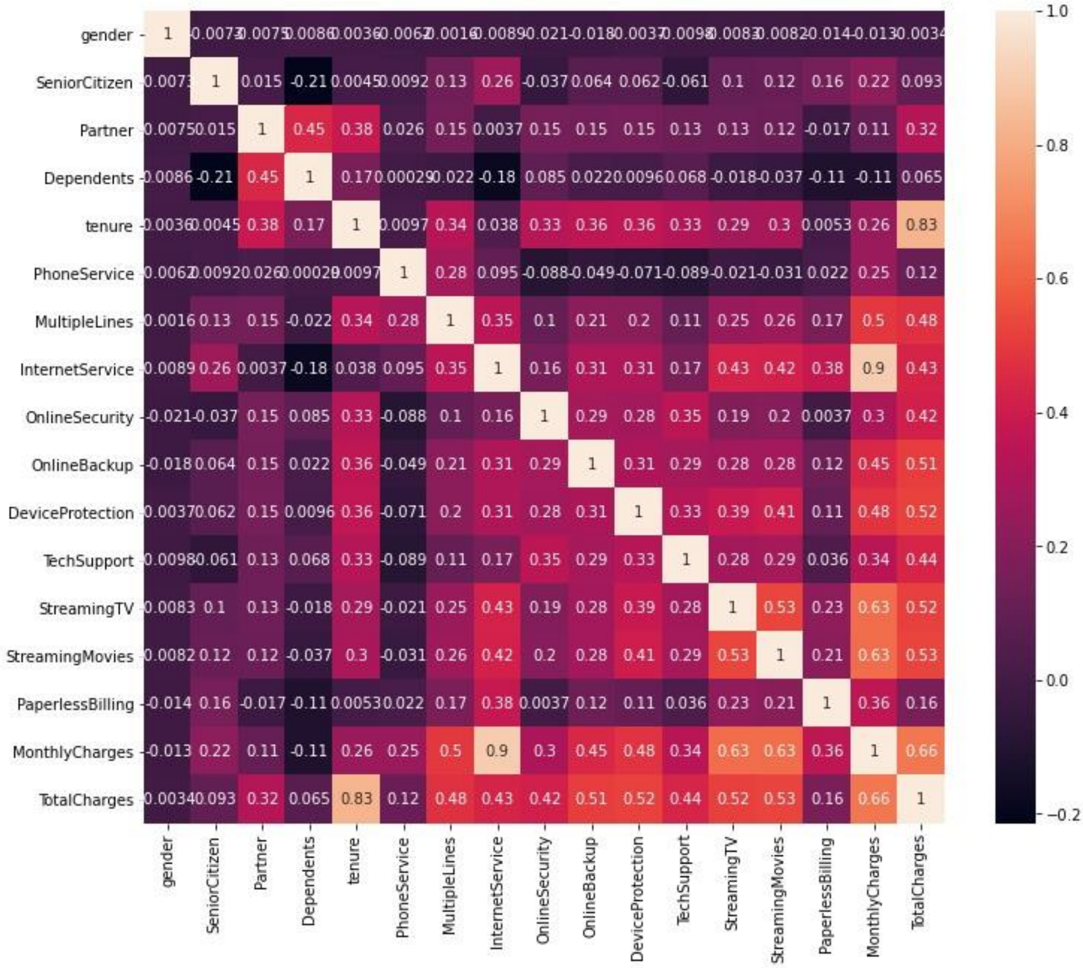


Facet Grid Plot



Heatmap

Out[138]: <AxesSubplot:>



Data Pre-processing Phases

There are lot of Yes/No values, let us replace it with 1 or 0.

- PhoneService - is the telephone service connected (Yes, No) - (1,0)
- MultipleLines - are multiple phone lines connected (Yes, No, No phone service) - (1,0,0)
- InternetService - client's Internet service provider (DSL, Fiber optic, No) -(2,1,0)
- OnlineSecurity - is the online security service connected (Yes, No, No internet service)-(1,0,0)
- OnlineBackup - is the online backup service activated (Yes, No, No internet service)(1,0,0)
- DeviceProtection - does the client have equipment insurance (Yes, No, No internet service)-(1,0,0)
- TechSupport - is the technical support service connected (Yes, No, No internet service)-(1,0,0)
- StreamingTV - is the streaming TV service connected (Yes, No, No internet service)(1,0,0)
- StreamingMovies - is the streaming cinema service activated (Yes, No, No internet service)-(1,0,0)
- Contract - type of customer contract (Month-to-month, One year, Two year) - (monthto-month - 1, One Year - 12, Two Year = 24)
- PaperlessBilling - whether the client uses paperless billing (Yes, No) - (1,0)
- PaymentMethod - payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))
- MonthlyCharges - current monthly payment
- TotalCharges - the total amount that the client paid for the services for the entire time Churn - whether there was a churn (Yes or No) - (0,1)

```
In [91]: df=df.replace('Yes',1)
         df=df.replace('No',0)
         df=df.replace('No internet service',0)
         df=df.replace('No phone service',0)
         df=df.replace('Fiber optic',2)
         df=df.replace('DSL',1)
         df=df.replace('Male',1)
         df=df.replace('Female',0)
```

Out[92]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSup
0	1	0	1	1	72	1	1	0	0	0	0	
1	0	0	0	0	44	1	0	2	0	1	1	
2	0	1	1	0	38	1	1	2	0	0	0	
3	1	0	0	0	4	1	0	1	0	0	0	
4	1	0	0	0	2	1	0	1	1	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	
5981	1	0	1	0	1	1	0	2	1	0	0	
5982	0	0	1	1	23	1	1	1	1	1	1	
5983	1	0	1	1	12	1	0	0	0	0	0	
5984	1	1	0	0	12	1	1	2	0	0	1	
5985	1	0	0	0	26	1	0	0	0	0	0	

5976 rows × 20 columns

Let see changed data types.

```
In [95]: df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 5976 entries, 0 to 5985
Data columns (total 20 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   gender            5976 non-null   int64
 1   SeniorCitizen     5976 non-null   int64
 2   Partner           5976 non-null   int64
 3   Dependents        5976 non-null   int64
 4   tenure            5976 non-null   int64
 5   PhoneService      5976 non-null   int64
 6   MultipleLines     5976 non-null   int64
 7   InternetService   5976 non-null   int64
 8   OnlineSecurity    5976 non-null   int64
 9   OnlineBackup      5976 non-null   int64
 10  DeviceProtection  5976 non-null   int64
 11  TechSupport       5976 non-null   int64
 12  StreamingTV       5976 non-null   int64
 13  StreamingMovies   5976 non-null   int64
 14  Contract          5976 non-null   object
 15  PaperlessBilling  5976 non-null   int64
 16  PaymentMethod     5976 non-null   object
 17  MonthlyCharges    5976 non-null   float64
 18  TotalCharges      5976 non-null   float64
 19  Churn             5976 non-null   int64
dtypes: float64(2), int64(16), object(2)
memory usage: 1.1+ MB
```

Contract and PaymentMethod are still object type.

Let it changes with get_dummies function in pandas. It converts categorical variable into dummy/indicator variables.

```
In [97]: df1=pd.get_dummies(data=df, columns=['Contract','PaymentMethod'], drop_first=True)
```

Data with object type has been converted to numerical data and data pre-processing phases are terminated.

Machine Learning Models

Logistic Regression

Logistic regression is a statistical method used to analyze a dataset with one or more independent variables that determine a result. The result is measured with a binary variable (there are only two possible outcomes).

In logistic regression, the dependent variable contains data encoded in binary or binary, i.e. only 1 (TRUE, success, churn, etc.) or 0 (FALSE, error, no-churn, etc.).

In this project, this variable is churn.

Logistic Regression with Split Method

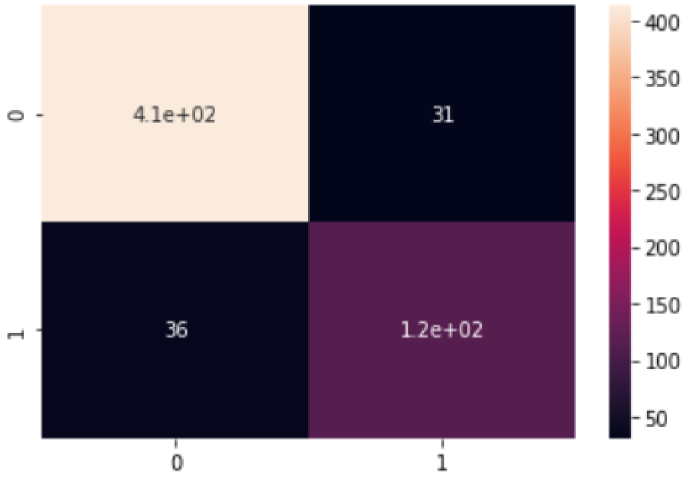
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

Metrics of Logistic Regression

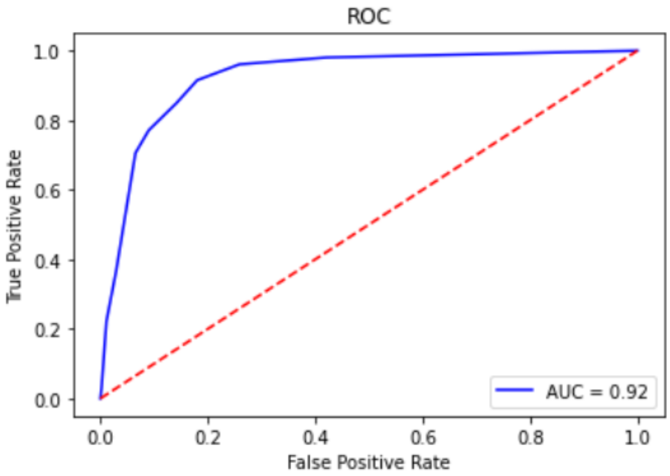
```
Metrics:
              precision    recall  f1-score   support

           0       0.92      0.93      0.92       444
           1       0.79      0.76      0.78       153

    accuracy                           0.89       597
   macro avg       0.86      0.85      0.85       597
weighted avg       0.89      0.89      0.89       597
```

Confusion Matrix of Logistic Regression

```
Confusion Matrix:
```



Plot TP and FP of Logistic Regression



Logistic Regression with Cross Validation Method

```
Confusion Matrix of Logistic Regression:
[[426  36]
 [ 33 103]]

Metrics:
              precision    recall  f1-score   support

           0       0.93      0.92      0.93       462
           1       0.74      0.76      0.75       136

    accuracy                           0.88       598
   macro avg       0.83      0.84      0.84       598
weighted avg       0.89      0.88      0.89       598
```

```
Confusion Matrix of Logistic Regression:
[[414  34]
 [ 44 106]]

Metrics:
              precision    recall  f1-score   support

           0       0.90      0.92      0.91       448
           1       0.76      0.71      0.73       150

    accuracy                           0.87       598
   macro avg       0.83      0.82      0.82       598
weighted avg       0.87      0.87      0.87       598
```

```
Confusion Matrix of Logistic Regression:
[[424  41]
 [ 34  99]]

Metrics:
              precision    recall  f1-score   support

           0       0.93      0.91      0.92       465
           1       0.71      0.74      0.73       133

    accuracy                           0.87       598
   macro avg       0.82      0.83      0.82       598
weighted avg       0.88      0.87      0.88       598
```

```
Confusion Matrix of Logistic Regression:
[[429  55]
 [ 28  86]]

Metrics:
              precision    recall  f1-score   support

           0       0.94      0.89      0.91       484
           1       0.61      0.75      0.67       114

    accuracy                           0.86       598
   macro avg       0.77      0.82      0.79       598
weighted avg       0.88      0.86      0.87       598
```

Parameter Tuning of Logistic Regression

```
  FitFailedWarning)
Best CV params {'C': 0.1, 'penalty': 'l2'}
Best CV accuracy 0.8804613123541107
Test accuracy of best hypers 0.8860971524288107
```

Decision Tree

The Decision Tree algorithm is used in the use and inference of multi-class data. It performs operations on numeric or non-numeric values.

It works by arranging the classes on the tree according to their entropy values and then giving the value to be controlled from the root node and progressing in the branches to reach the result. The structure of the tree changes and takes shape according to train dataset.

Decision Tree with Split Method

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=None, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
```

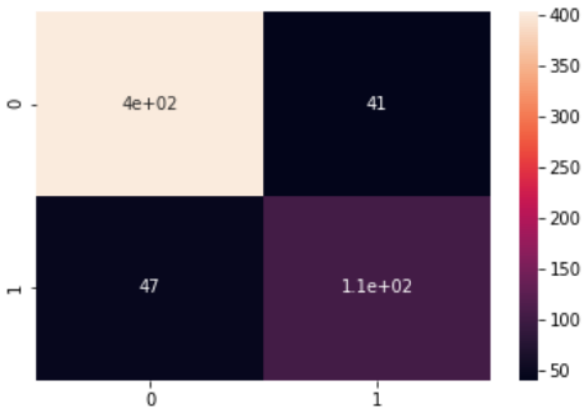
Metrics of Decision Tree

Metrics:

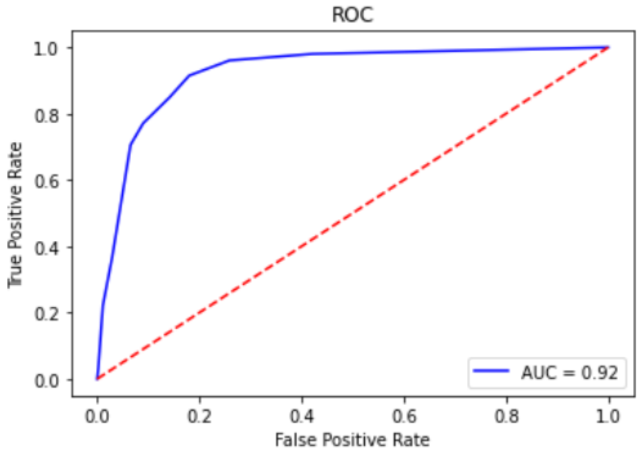
	precision	recall	f1-score	support
0	0.90	0.91	0.90	444
1	0.72	0.69	0.71	153
accuracy			0.85	597
macro avg	0.81	0.80	0.80	597
weighted avg	0.85	0.85	0.85	597

Confusion Matrix of Decision Tree

Confusion Matrix:



Plot TP and FP of Decision Tree



Decision Tree with Cross Validation Method

```
Confusion Matrix of Decision Tree Classifier:
[[412  50]
 [ 52  84]]

Metrics:
              precision    recall  f1-score   support

           0       0.89      0.89      0.89       462
           1       0.63      0.62      0.62       136

    accuracy                           0.83       598
   macro avg       0.76      0.75      0.76       598
weighted avg       0.83      0.83      0.83       598
```

```
Confusion Matrix of Decision Tree Classifier:
[[407  41]
 [ 44 106]]

Metrics:
              precision    recall  f1-score   support

           0       0.90      0.91      0.91       448
           1       0.72      0.71      0.71       150

    accuracy                           0.86       598
   macro avg       0.81      0.81      0.81       598
weighted avg       0.86      0.86      0.86       598
```

```
Confusion Matrix of Decision Tree Classifier:
[[415  50]
 [ 45  88]]

Metrics:
              precision    recall  f1-score   support

           0       0.90      0.89      0.90       465
           1       0.64      0.66      0.65       133

    accuracy                           0.84       598
   macro avg       0.77      0.78      0.77       598
weighted avg       0.84      0.84      0.84       598


Confusion Matrix of Decision Tree Classifier:
[[432  52]
 [ 46  68]]

Metrics:
              precision    recall  f1-score   support

           0       0.90      0.89      0.90       484
           1       0.57      0.60      0.58       114

    accuracy                           0.84       598
   macro avg       0.74      0.74      0.74       598
weighted avg       0.84      0.84      0.84       598
```

Parameter Tuning of Decision Tree

```
Best CV params {'criterion': 'gini', 'max_depth': 3, 'min_samples_leaf': 0.1}
Best CV accuracy 0.871725425780237
Test accuracy of best hypers 0.8793969849246231
```

K-Nearest Neighbor

In K-NN classification, the output is class membership. An object is classified by the majority of its neighbors; the object is given the class that is most common among its nearest neighbors (k is a small positive integer). If k = 1, the object is simply assigned to that nearest neighbor's class.

In K-NN regression, the output is the property value of the object. This value is the average of the values of its nearest neighbors.

K-NN is a type of pattern-based learning or lazy learning; where the function is only approximated locally and all computation is deferred until classification. The K-N algorithm is among the simplest of all machine learning algorithms.

K-Nearest Neighbor with Split Method

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=10, p=2,
                     weights='uniform')
```

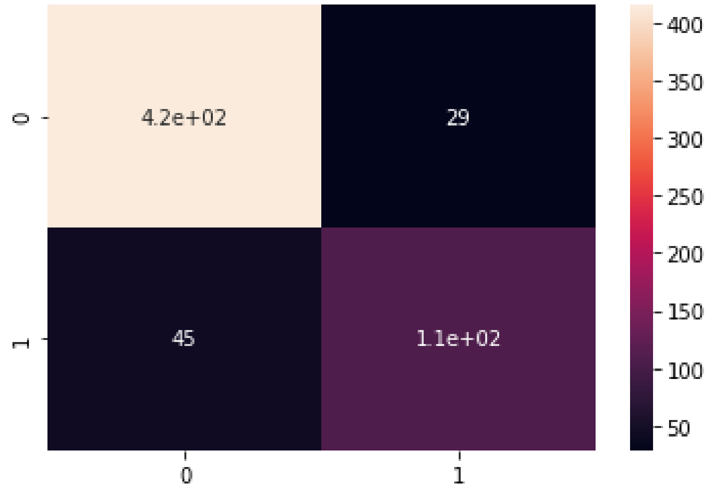
Metrics of K-Nearest Neighbor

Metrics:

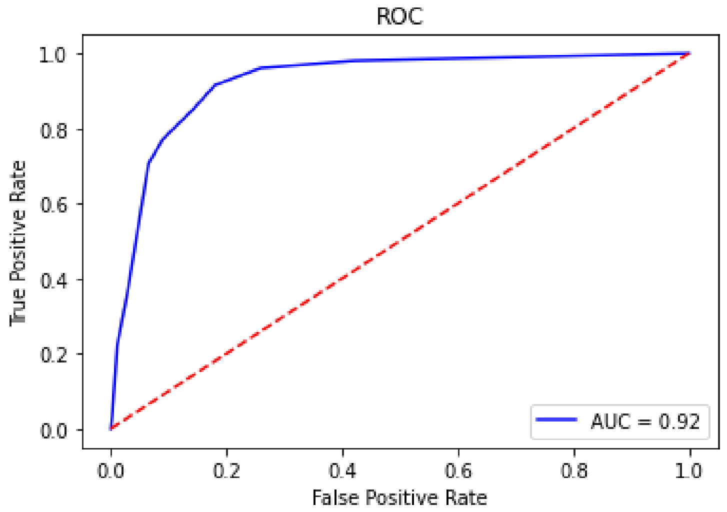
	precision	recall	f1-score	support
0	0.90	0.93	0.92	444
1	0.79	0.71	0.74	153
accuracy			0.88	597
macro avg	0.85	0.82	0.83	597
weighted avg	0.87	0.88	0.87	597

Confusion Matrix of K-Nearest Neighbor

Confusion Matrix:



Plot TP and FP of K-Nearest Neighbor



K-Nearest Neighbor with Cross Validation Model

```
Confusion Matrix of K-NN Classifier:
[[424  38]
 [ 50  86]]

Metrics:
              precision    recall  f1-score   support

           0       0.89      0.92      0.91       462
           1       0.69      0.63      0.66       136

    accuracy                           0.85       598
   macro avg       0.79      0.78      0.78       598
weighted avg       0.85      0.85      0.85       598
```

```
Confusion Matrix of K-NN Classifier:
[[410  38]
 [ 58  92]]

Metrics:
              precision    recall  f1-score   support

           0       0.88      0.92      0.90       448
           1       0.71      0.61      0.66       150

    accuracy                           0.84       598
   macro avg       0.79      0.76      0.78       598
weighted avg       0.83      0.84      0.84       598
```

```
Confusion Matrix of K-NN Classifier:
[[419  46]
 [ 45  88]]

Metrics:
              precision    recall  f1-score   support

           0       0.90      0.90      0.90       465
           1       0.66      0.66      0.66       133

    accuracy                           0.85       598
   macro avg       0.78      0.78      0.78       598
weighted avg       0.85      0.85      0.85       598
```

```
Confusion Matrix of K-NN Classifier:
[[426  58]
 [ 46  68]]

Metrics:
              precision    recall  f1-score   support

           0       0.90      0.88      0.89       484
           1       0.54      0.60      0.57       114

    accuracy                           0.83       598
   macro avg       0.72      0.74      0.73       598
weighted avg       0.83      0.83      0.83       598
```

Parameter Tuning of K-Nearest Neighbor

```
Best CV params {'n_neighbors': 5}
Best CV accuracy 0.8473683755511369
Test accuracy of best hypers 0.8743718592964824
```