Customer Churn Analysis

Customer Churn Analysis loss of customers to competition) is a problem for telecom companies because it is expensive to acquire a new customer and companies want to retain their existing customers. Most telecom companies suffer from voluntary churn.

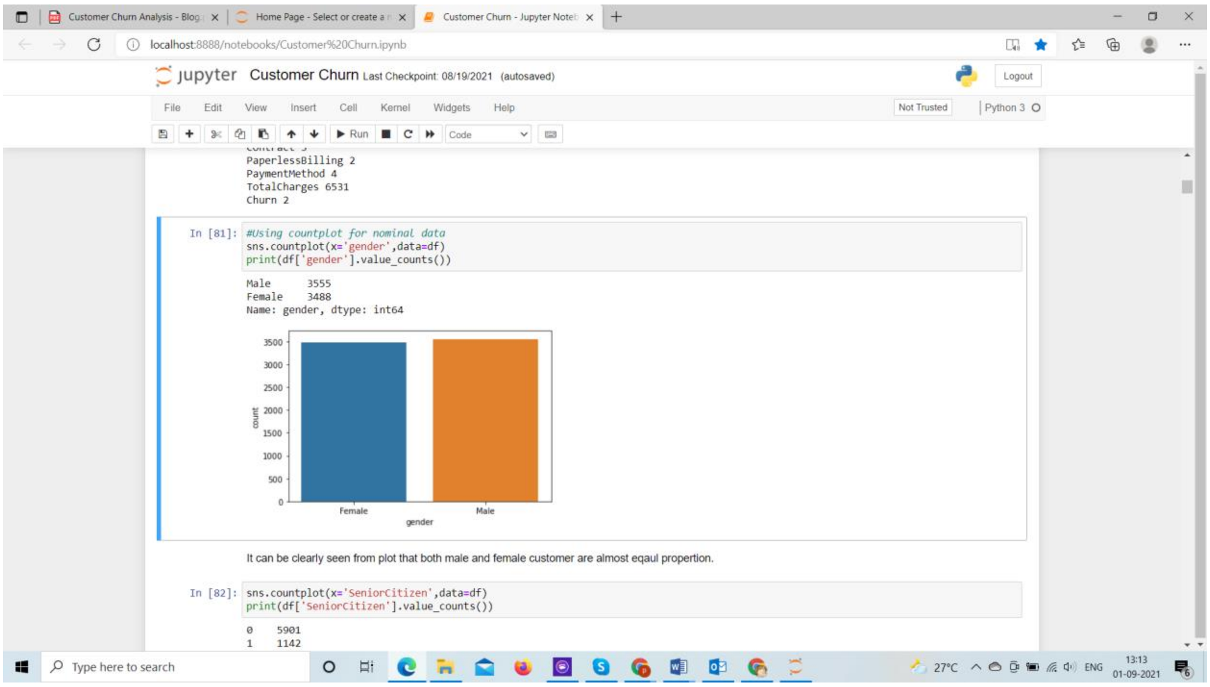
Customer churn is basically when telecom company start losing customers and goes in loss. Companies are usually need data for measuring their customer churn data of existing customers. Existing customers will often have a higher volume of service consumption and can generate additional customer referrals

Companies basically depends on the profits that they usually get from customer and this can be achieved by various means like by providing good customer support, online support and good services.

Preventing customer churn is critically important to the telecommunications sector, as the barriers to entry for switching services are so low. I will examine customer data from IBM Sample Data Sets with the aim of building and comparing several customer churn prediction models.

EDA

Gender Plot


Customer Churn Analysis - Blog
Home Page - Select or create a
Customer Churn - Jupyter Noteb
localhost:8888/notebooks/Customer%20Churn.ipynb
jupyter Customer Churn Last Checkpoint: 08/19/2021 (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3
Run
Code
3500
3000
2500
2000
1500
1000
500
0
count
Female
Male
gender
Type here to search
27°C
ENG
13:13
01-09-2021

```
PaperlessBilling 2
PaymentMethod 4
TotalCharges 6531
Churn 2
```

```
In [81]: #Using countplot for nominal data
         sns.countplot(x='gender',data=df)
         print(df['gender'].value_counts())
```

```
Male      3555
Female    3488
Name: gender, dtype: int64
```

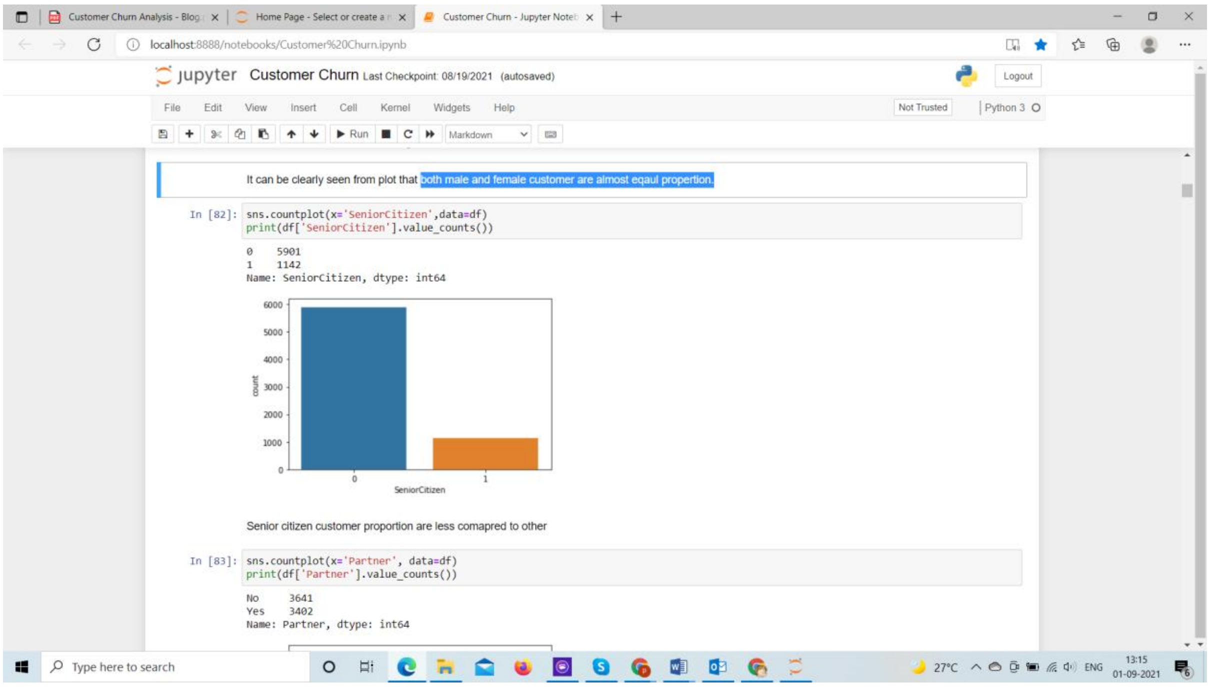
It can be clearly seen from plot that both male and female customer are almost eqaul propertion.

```
In [82]: sns.countplot(x='SeniorCitizen',data=df)
         print(df['SeniorCitizen'].value_counts())
```

```
0    5901
1    1142
```

As Gender Plot describes that both male and female customer are almost equal proportion.

Senior Citizen


Customer Churn Analysis - Blog
Home Page - Select or create a
Customer Churn - Jupyter Noteb
localhost:8888/notebooks/Customer%20Churn.ipynb
jupyter Customer Churn Last Checkpoint: 08/19/2021 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
6000
5000
4000
3000
2000
1000
0
count
0
1
SeniorCitizen
Type here to search
27°C
ENG
13:15
01-09-2021

It can be clearly seen from plot that both male and female customer are almost eqaul propertion.

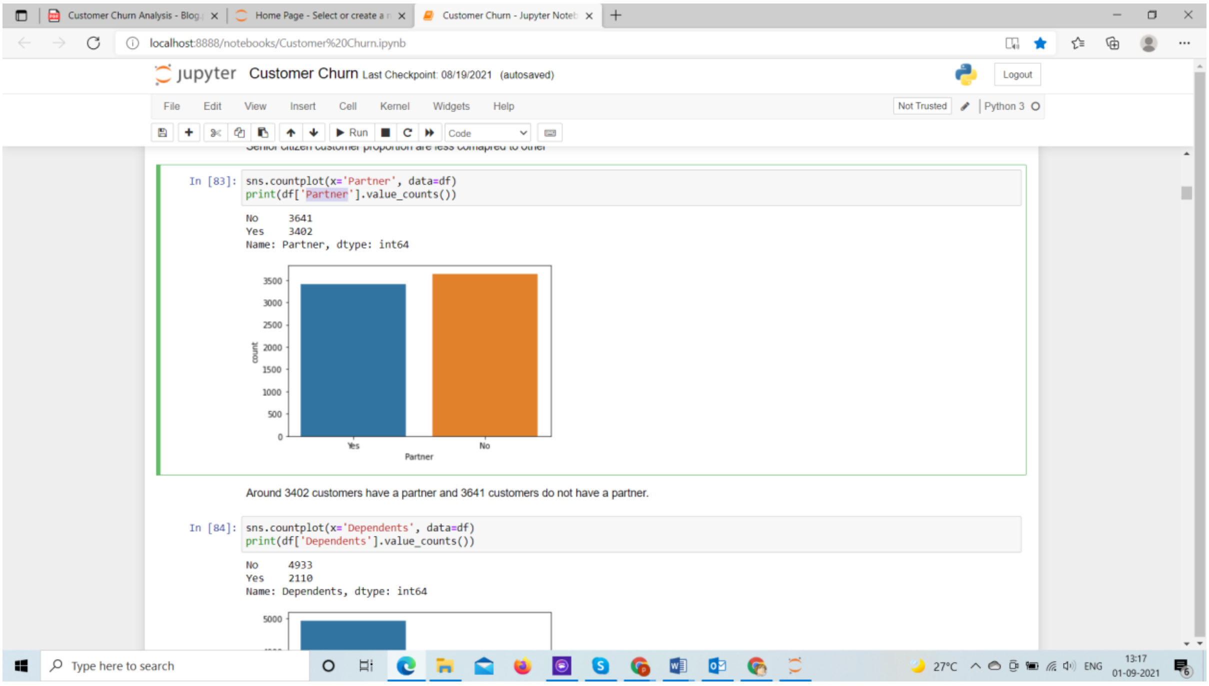
```
In [82]: sns.countplot(x='SeniorCitizen',data=df)
         print(df['SeniorCitizen'].value_counts())

0    5901
1    1142
Name: SeniorCitizen, dtype: int64
```

Senior citizen customer proportion are less comapred to other

```
In [83]: sns.countplot(x='Partner', data=df)
         print(df['Partner'].value_counts())

No     3641
Yes    3402
Name: Partner, dtype: int64
```

Senior citizen customer proportion are less comapred to other

Partner


Customer Churn Analysis - Blog
Home Page - Select or create a
Customer Churn - Jupyter Noteb
localhost:8888/notebooks/Customer%20Churn.ipynb
jupyter Customer Churn Last Checkpoint: 08/19/2021 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
3500
3000
2500
2000
1500
1000
500
0
count
Yes
No
Partner
5000
Type here to search
27°C
ENG
13:17
01-09-2021

Senior citizen customer proportion are less comapred to other

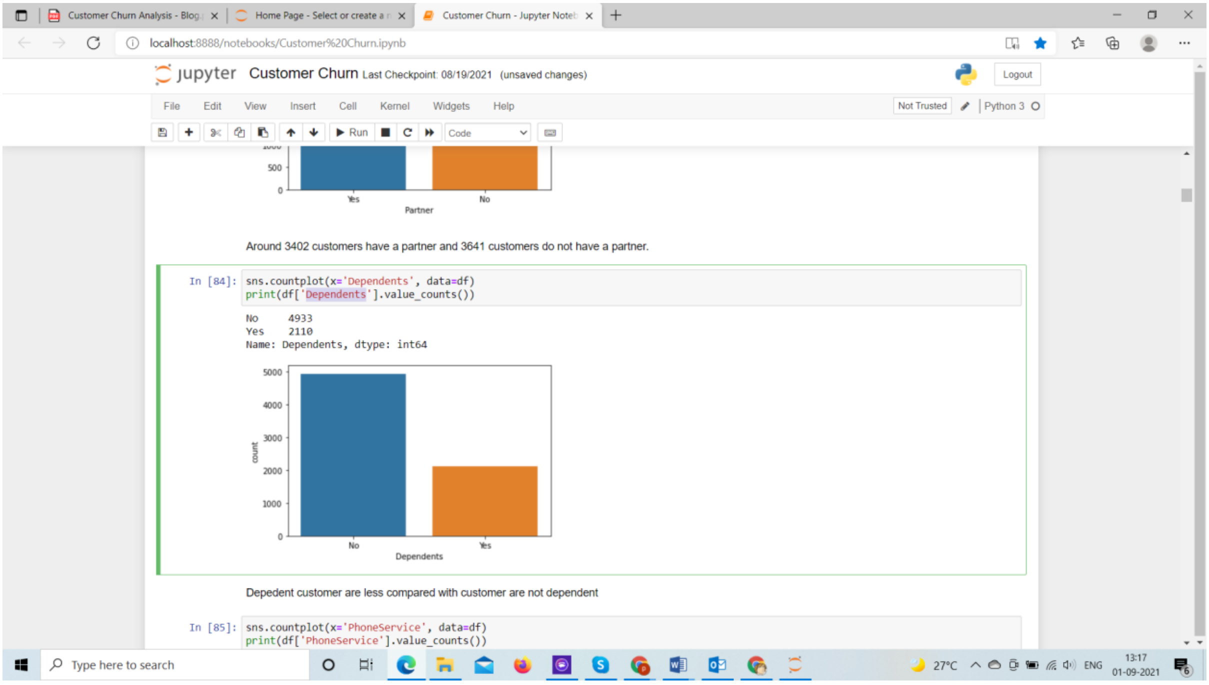
```
In [83]: sns.countplot(x='Partner', data=df)
         print(df['Partner'].value_counts())

No     3641
Yes    3402
Name: Partner, dtype: int64
```

Around 3402 customers have a partner and 3641 customers do not have a partner.

```
In [84]: sns.countplot(x='Dependents', data=df)
         print(df['Dependents'].value_counts())

No     4933
Yes    2110
Name: Dependents, dtype: int64
```

Around 3402 customers have a partner and 3641 customers do not have a partner.

Dependents


Customer Churn Analysis - Blog
Home Page - Select or create a n
Customer Churn - Jupyter Noteb
localhost:8888/notebooks/Customer%20Churn.ipynb
jupyter Customer Churn Last Checkpoint: 08/19/2021 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
500
0
Yes
No
Partner
5000
4000
3000
2000
1000
0
count
No
Yes
Dependents
Type here to search
27°C
ENG
13:17
01-09-2021

Around 3402 customers have a partner and 3641 customers do not have a partner.

```
In [84]: sns.countplot(x='Dependents', data=df)
         print(df['Dependents'].value_counts())
```

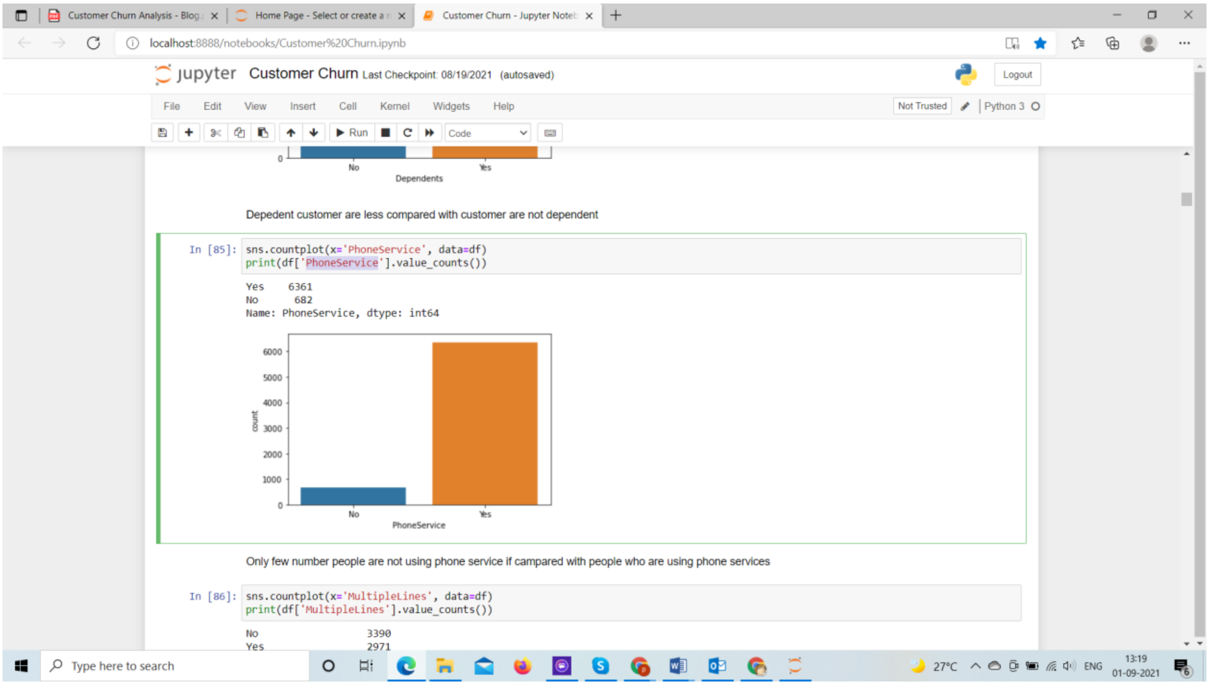
```
No     4933
Yes    2110
Name: Dependents, dtype: int64
```

Depedent customer are less compared with customer are not dependent

```
In [85]: sns.countplot(x='PhoneService', data=df)
         print(df['PhoneService'].value_counts())
```

Telecom Depedent customer are less compared with customer are not dependent

Phone Service


Customer Churn Analysis - Blog
Home Page - Select or create a n
Customer Churn - Jupyter Noteb
localhost:8888/notebooks/Customer%20Churn.ipynb
jupyter Customer Churn Last Checkpoint: 08/19/2021 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
0
No
Yes
Dependents
6000
5000
4000
3000
2000
1000
0
count
No
Yes
PhoneService
Type here to search
27°C
ENG
13:19
01-09-2021

Depedent customer are less compared with customer are not dependent

```
In [85]: sns.countplot(x='PhoneService', data=df)
         print(df['PhoneService'].value_counts())
```

```
Yes    6361
No      682
Name: PhoneService, dtype: int64
```

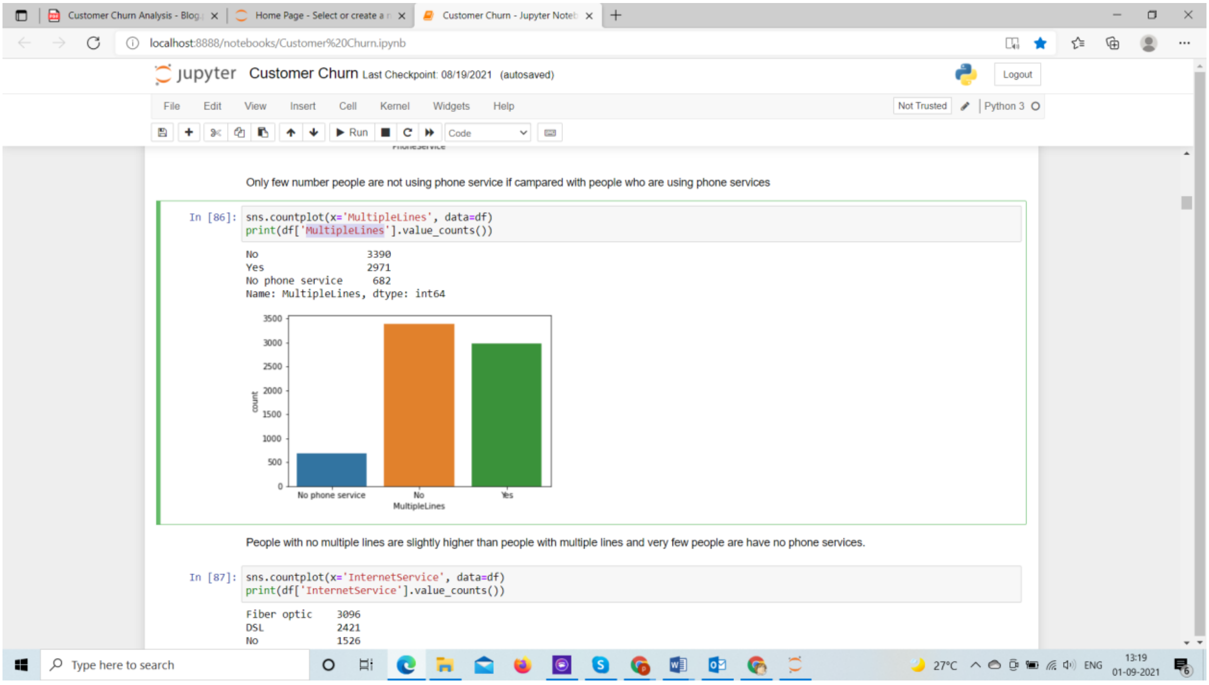
Only few number people are not using phone service if campared with people who are using phone services

```
In [86]: sns.countplot(x='MultipleLines', data=df)
         print(df['MultipleLines'].value_counts())
```

```
No                  3390
Yes                 2971
```

Only few number people are not using phone service if campared with people who are using phone services

Multiple Lines


Customer Churn Analysis - Blog
Home Page - Select or create a
Customer Churn - Jupyter Noteb
localhost:8888/notebooks/Customer%20Churn.ipynb
jupyter Customer Churn Last Checkpoint: 08/19/2021 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
3500
3000
2500
2000
1500
1000
500
0
count
No phone service
No
Yes
MultipleLines
Type here to search
27°C
ENG
13:19
01-09-2021

Only few number people are not using phone service if campared with people who are using phone services

In [86]:
```
sns.countplot(x='MultipleLines', data=df)
print(df['MultipleLines'].value_counts())
```

```
No                  3390
Yes                 2971
No phone service     682
Name: MultipleLines, dtype: int64
```

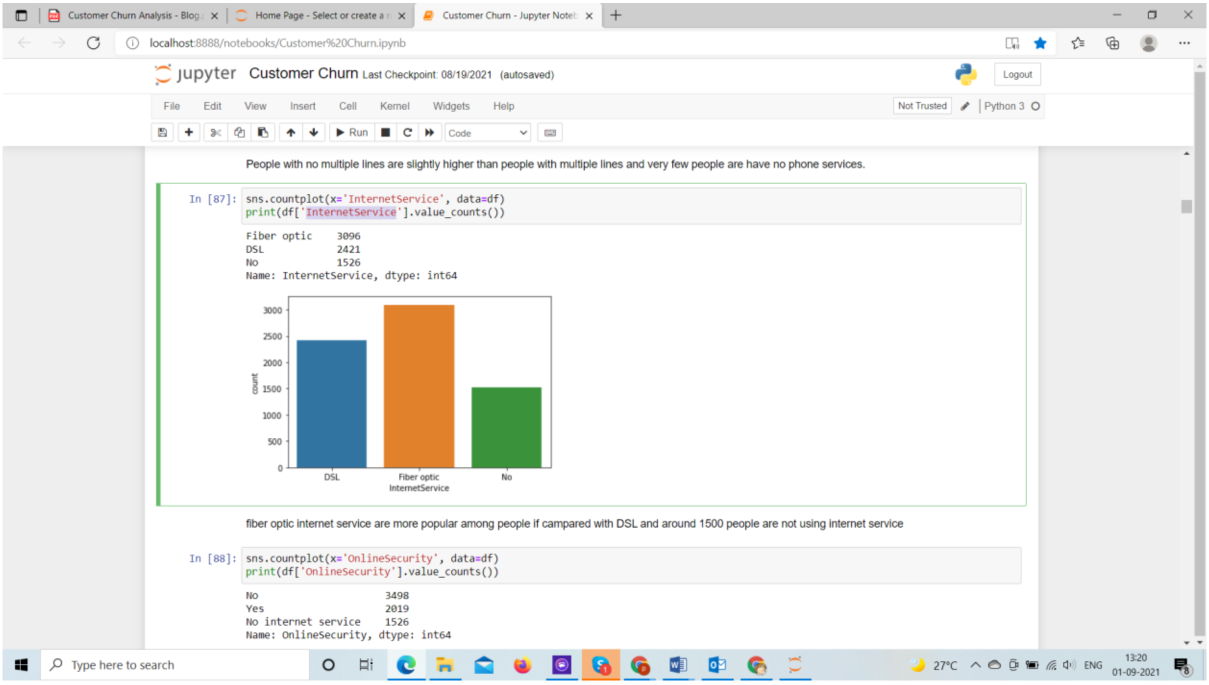
People with no multiple lines are slightly higher than people with multiple lines and very few people are have no phone services.

In [87]:
```
sns.countplot(x='InternetService', data=df)
print(df['InternetService'].value_counts())
```

```
Fiber optic    3096
DSL            2421
No             1526
```

People with no multiple lines are slightly higher than people with multiple lines and very few people are having no phone services.

Internet Service


Customer Churn Analysis - Blog
Home Page - Select or create a
Customer Churn - Jupyter Noteb
localhost:8888/notebooks/Customer%20Churn.ipynb
jupyter Customer Churn Last Checkpoint: 08/19/2021 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
3000
2500
2000
1500
1000
500
0
count
DSL
Fiber optic
No
InternetService
Type here to search
27°C
ENG
13:20
01-09-2021

People with no multiple lines are slightly higher than people with multiple lines and very few people are have no phone services.

In [87]:
```
sns.countplot(x='InternetService', data=df)
print(df['InternetService'].value_counts())
```

```
Fiber optic    3096
DSL            2421
No             1526
Name: InternetService, dtype: int64
```

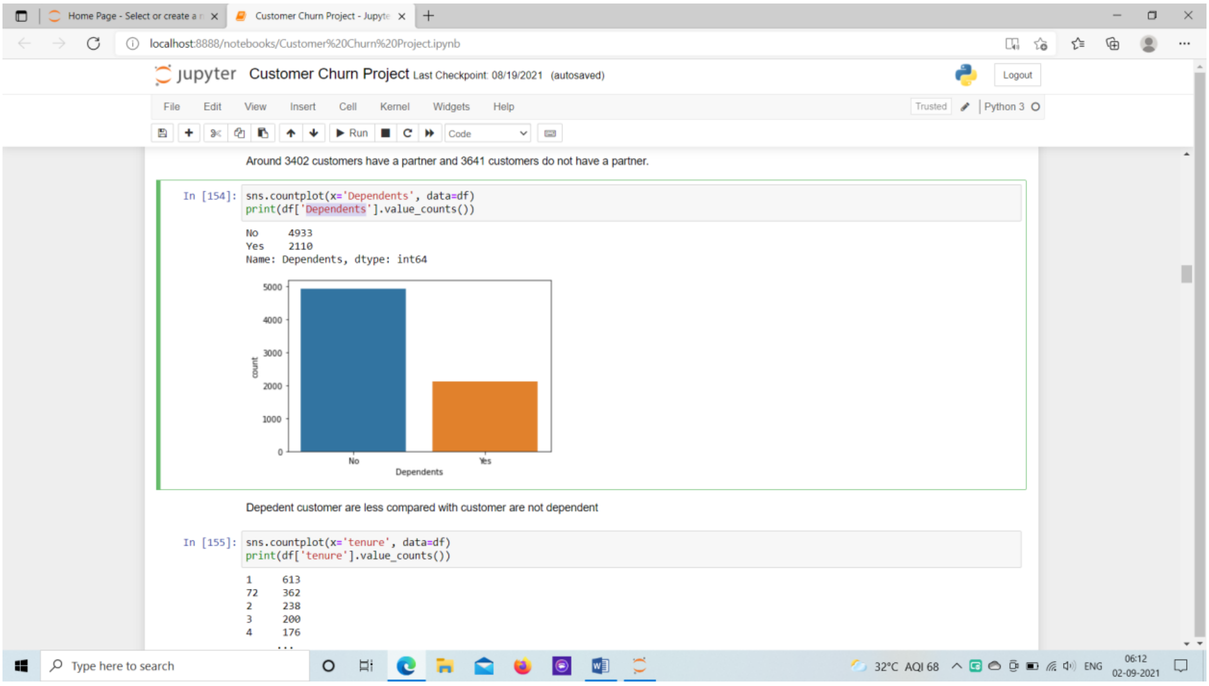
fiber optic internet service are more popular among people if campared with DSL and around 1500 people are not using internet service

In [88]:
```
sns.countplot(x='OnlineSecurity', data=df)
print(df['OnlineSecurity'].value_counts())
```

```
No                     3498
Yes                    2019
No internet service    1526
Name: OnlineSecurity, dtype: int64
```

fiber optic internet service is more popular among people if compared with DSL and around 1500 people are not using internet service

Dependents


Home Page - Select or create a
Customer Churn Project - Jupyter
localhost:8888/notebooks/Customer%20Churn%20Project.ipynb
jupyter Customer Churn Project Last Checkpoint: 08/19/2021 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
5000
4000
3000
2000
1000
0
count
No
Yes
Dependents
Type here to search
32°C AQI 68
ENG
06:12
02-09-2021

Around 3402 customers have a partner and 3641 customers do not have a partner.

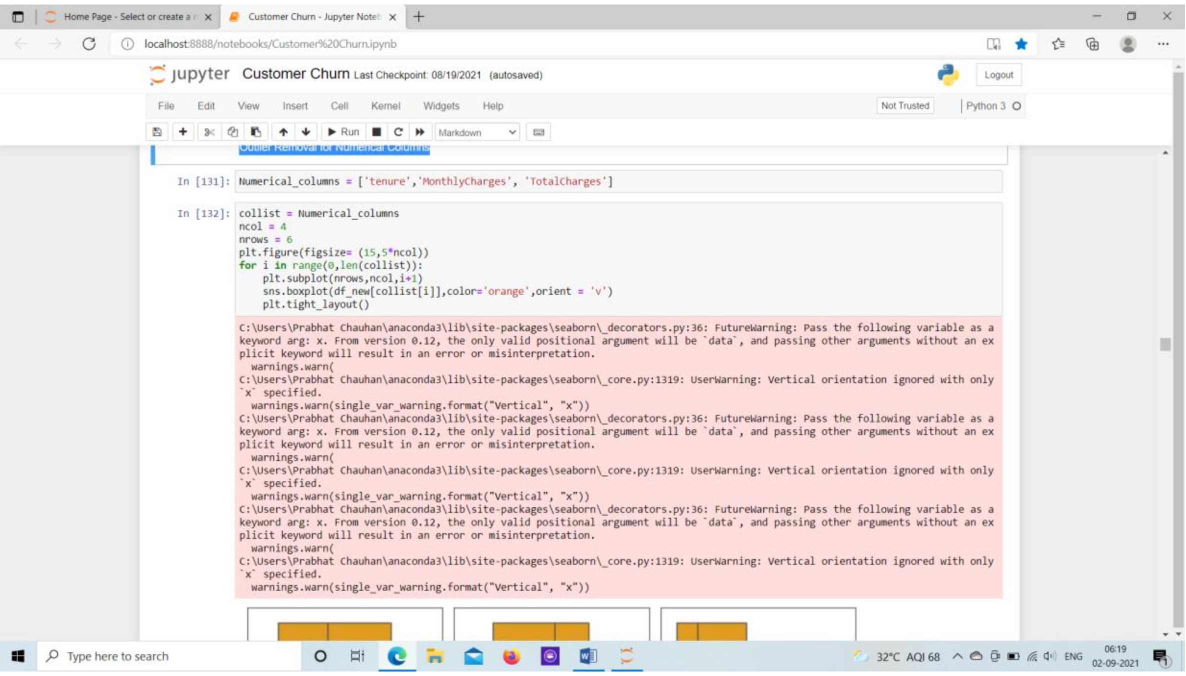
```
In [154]: sns.countplot(x='Dependents', data=df)
          print(df['Dependents'].value_counts())

No     4933
Yes    2110
Name: Dependents, dtype: int64
```

Depedent customer are less compared with customer are not dependent

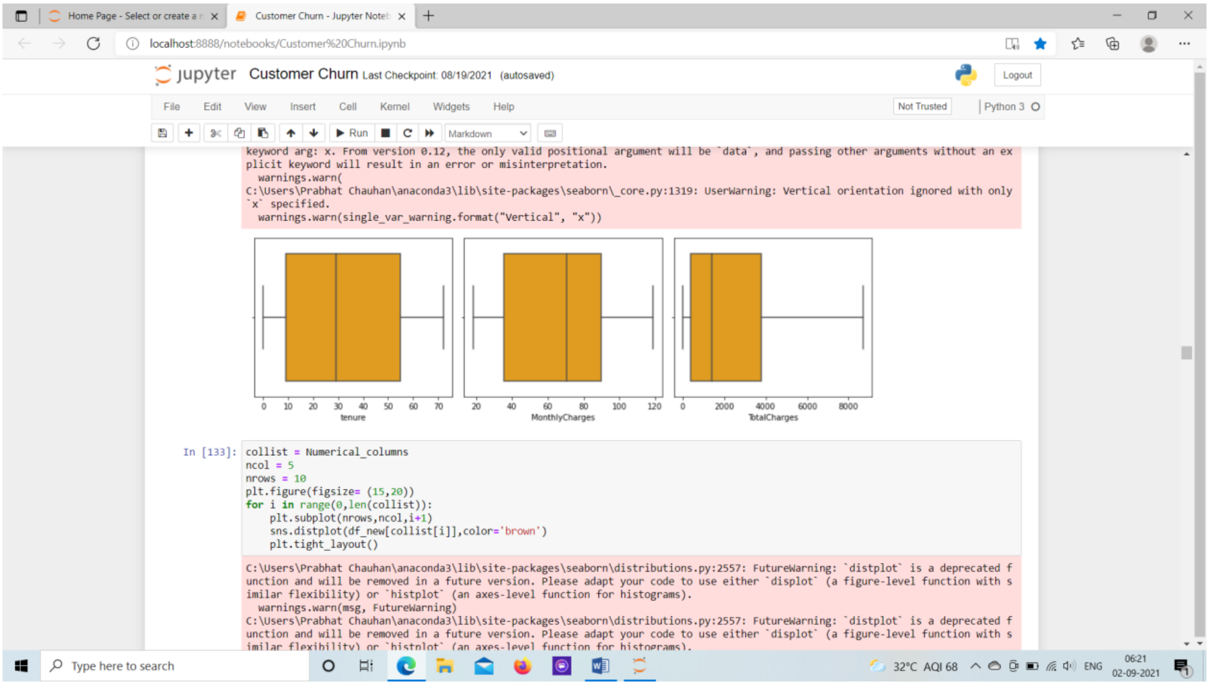
```
In [155]: sns.countplot(x='tenure', data=df)
          print(df['tenure'].value_counts())

1     613
72    362
2     238
3     200
4     176
     ...
```

Depedent customer are less compared with customer are not dependent

Outlier Removal for Numerical Columns


Home Page - Select or create a
Customer Churn - Jupyter Noteb
localhost:8888/notebooks/Customer%20Churn.ipynb
jupyter Customer Churn Last Checkpoint: 08/19/2021 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
Outlier Removal for Numerical Columns
Type here to search
32°C AQI 68
ENG
06:19
02-09-2021

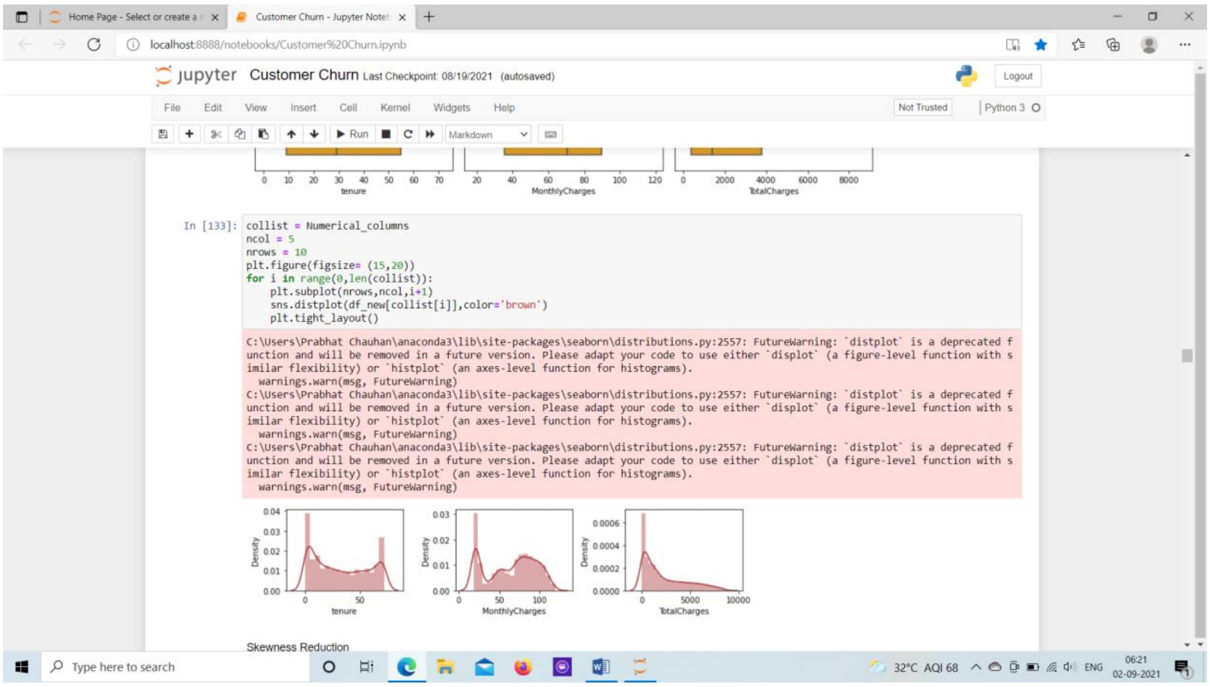
```
In [131]: Numerical_columns = ['tenure','MonthlyCharges', 'TotalCharges']

In [132]: collist = Numerical_columns
          ncol = 4
          nrows = 6
          plt.figure(figsize= (15,5*ncol))
          for i in range(0,len(collist)):
              plt.subplot(nrows,ncol,i+1)
              sns.boxplot(df_new[collist[i]],color='orange',orient = 'v')
              plt.tight_layout()

C:\Users\Prabhat Chauhan\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a
keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an ex
plicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\Prabhat Chauhan\anaconda3\lib\site-packages\seaborn\_core.py:1319: UserWarning: Vertical orientation ignored with only
`x` specified.
  warnings.warn(single_var_warning.format("Vertical", "x"))
C:\Users\Prabhat Chauhan\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a
keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an ex
plicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\Prabhat Chauhan\anaconda3\lib\site-packages\seaborn\_core.py:1319: UserWarning: Vertical orientation ignored with only
`x` specified.
  warnings.warn(single_var_warning.format("Vertical", "x"))
C:\Users\Prabhat Chauhan\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a
keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an ex
plicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\Prabhat Chauhan\anaconda3\lib\site-packages\seaborn\_core.py:1319: UserWarning: Vertical orientation ignored with only
`x` specified.
  warnings.warn(single_var_warning.format("Vertical", "x"))
```



```
keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an ex
plicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\Prabhat Chauhan\anaconda3\lib\site-packages\seaborn\_core.py:1319: UserWarning: Vertical orientation ignored with only
`x` specified.
  warnings.warn(single_var_warning.format("Vertical", "x"))
```

```
In [133]: collist = Numerical_columns
          ncol = 5
          nrows = 10
          plt.figure(figsize= (15,20))
          for i in range(0,len(collist)):
              plt.subplot(nrows,ncol,i+1)
              sns.distplot(df_new[collist[i]],color='brown')
              plt.tight_layout()
```

```
C:\Users\Prabhat Chauhan\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated f
unction and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with s
imilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Prabhat Chauhan\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated f
unction and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with s
imilar flexibility) or `histplot` (an axes-level function for histograms).
```

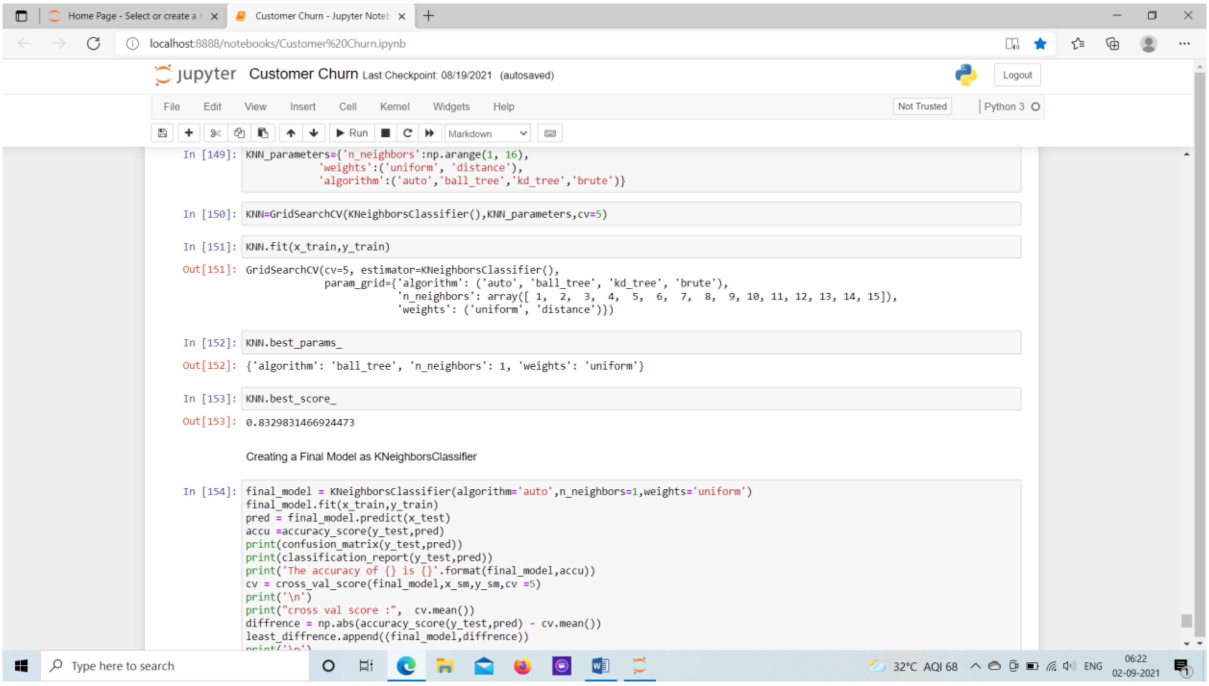


```
In [133]: collist = Numerical_columns
          ncol = 5
          nrows = 10
          plt.figure(figsize= (15,20))
          for i in range(0,len(collist)):
              plt.subplot(nrows,ncol,i+1)
              sns.distplot(df_new[collist[i]],color='brown')
              plt.tight_layout()
```

```
C:\Users\Prabhat Chauhan\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated f
unction and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with s
imilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Prabhat Chauhan\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated f
unction and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with s
imilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Prabhat Chauhan\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated f
unction and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with s
imilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Skewness Reduction

Hyperparameter Tuning

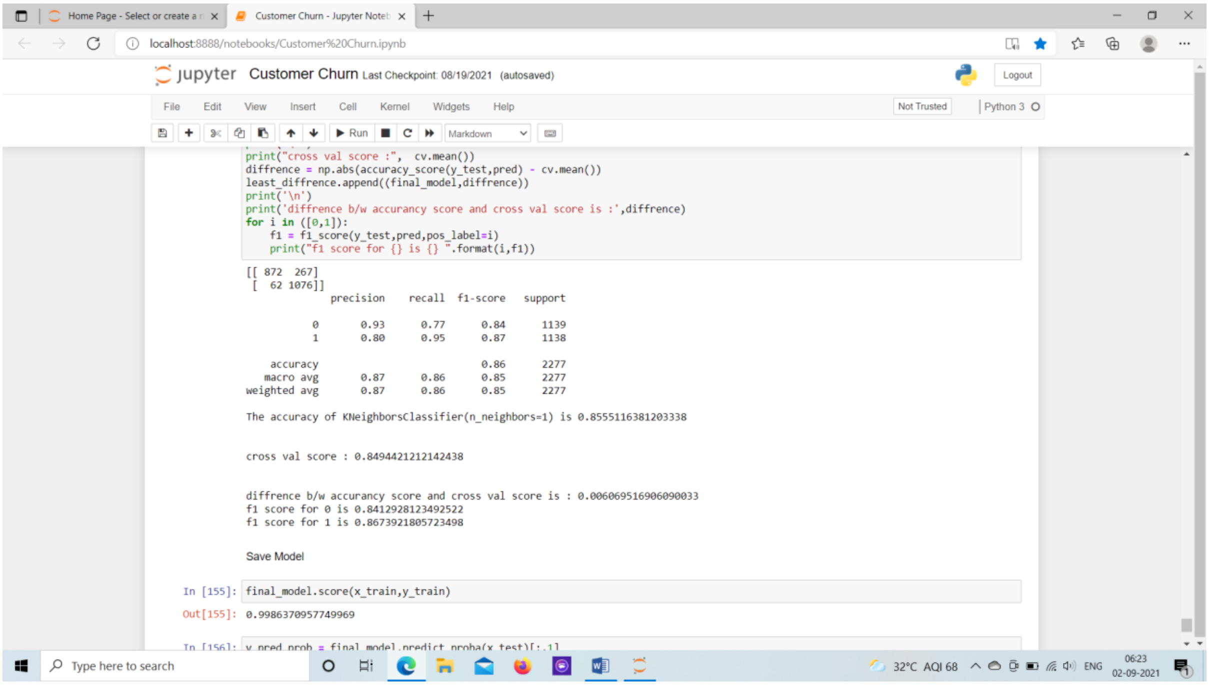

Home Page - Select or create a
Customer Churn - Jupyter Notet
localhost:8888/notebooks/Customer%20Churn.ipynb
jupyter Customer Churn Last Checkpoint: 08/19/2021 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
Type here to search
32°C AQI 68
ENG
06:22
02-09-2021

```
In [149]: KNN_parameters={'n_neighbors':np.arange(1, 16),
                      'weights':('uniform', 'distance'),
                      'algorithm':('auto','ball_tree','kd_tree','brute')}

In [150]: KNN=GridSearchCV(KNeighborsClassifier(),KNN_parameters,cv=5)

In [151]: KNN.fit(x_train,y_train)

Out[151]: GridSearchCV(cv=5, estimator=KNeighborsClassifier(),
                       param_grid={'algorithm': ('auto', 'ball_tree', 'kd_tree', 'brute'),
                                   'n_neighbors': array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15]),
                                   'weights': ('uniform', 'distance')})

In [152]: KNN.best_params_

Out[152]: {'algorithm': 'ball_tree', 'n_neighbors': 1, 'weights': 'uniform'}

In [153]: KNN.best_score_

Out[153]: 0.8329831466924473
```

Creating a Final Model as KNeighborsClassifier

```
In [154]: final_model = KNeighborsClassifier(algorithm='auto',n_neighbors=1,weights='uniform')
          final_model.fit(x_train,y_train)
          pred = final_model.predict(x_test)
          accu =accuracy_score(y_test,pred)
          print(confusion_matrix(y_test,pred))
          print(classification_report(y_test,pred))
          print('The accuracy of {} is {}'.format(final_model,accu))
          cv = cross_val_score(final_model,x_sm,y_sm,cv =5)
          print('\n')
          print("cross val score :",  cv.mean())
          diffrence = np.abs(accuracy_score(y_test,pred) - cv.mean())
          least_diffrence.append((final_model,diffrence))
          print('\n')
```

Creating a Final Model as KNeighborsClassifier


Home Page - Select or create a
Customer Churn - Jupyter Notet
localhost:8888/notebooks/Customer%20Churn.ipynb
jupyter Customer Churn Last Checkpoint: 08/19/2021 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
Type here to search
32°C AQI 68
ENG
06:23
02-09-2021

```
          print("cross val score :",  cv.mean())
          diffrence = np.abs(accuracy_score(y_test,pred) - cv.mean())
          least_diffrence.append((final_model,diffrence))
          print('\n')
          print('diffrence b/w accurancy score and cross val score is :',diffrence)
          for i in ([0,1]):
              f1 = f1_score(y_test,pred,pos_label=i)
              print("f1 score for {} is {} ".format(i,f1))

          [[ 872  267]
           [  62 1076]]
                        precision    recall  f1-score   support

                     0       0.93      0.77      0.84      1139
                     1       0.80      0.95      0.87      1138

              accuracy                           0.86      2277
             macro avg       0.87      0.86      0.85      2277
          weighted avg       0.87      0.86      0.85      2277

          The accuracy of KNeighborsClassifier(n_neighbors=1) is 0.8555116381203338


          cross val score : 0.8494421212142438


          diffrence b/w accurancy score and cross val score is : 0.006069516906090033
          f1 score for 0 is 0.8412928123492522
          f1 score for 1 is 0.8673921805723498
```

Save Model

```
In [155]: final_model.score(x_train,y_train)

Out[155]: 0.9986370957749969

In [156]: y_pred_prob = final_model.predict_proba(x_test)[:,1]
```

We have roc_auc_score of 86

Result

Best Model had been concluded is 83%

- **people having very high tenure or very less tenure are leaving company**
- **people don’t have the phone services aren’t enjoying other services , so**
- **probably customer is leaving , here company can work upon new**
- **schemes so that customer can get attract towards services.**
- **Logistic Regression algorithm looks best for the**
- **telecom customer churn dataset, which will predict the churn analysis**