DataTrained
Keep Skilling, Keep Growing

Customer churn analysis

Name of the project:

Submitted by:

Pooja Rajpal(batch0522)

Problem Statement:

Customer churn is when a customer stops doing business with that company.Customer churn is very biggest expenditure of any organization.As keeping an existing is less expensive than acquiring new customer.If the company could figure out why the customer is leaving and when they leave ,it would help them for strategize their retention initiatives.

Customer retention can be achieved with good products and services.Along with it they must know the reason why are they leaving to prevent attrition.

This dataset is about the customer churn prevention from telecommunication sector.

Lets attempt to solve some of the challenges pertaining to customer attrition .say for eg

1) What is the reason that the active customer is leaving an organization?
2) Key indicators of customer churn.
3) What strategies should be followed by the company to diminish customer churn.

In real-world we have to go through seven major

Stages for the prediction of customer churn

1)Data preprocessing

2)Data evaluation

3)Model selection

4)Model evaluation

5)Model improvement

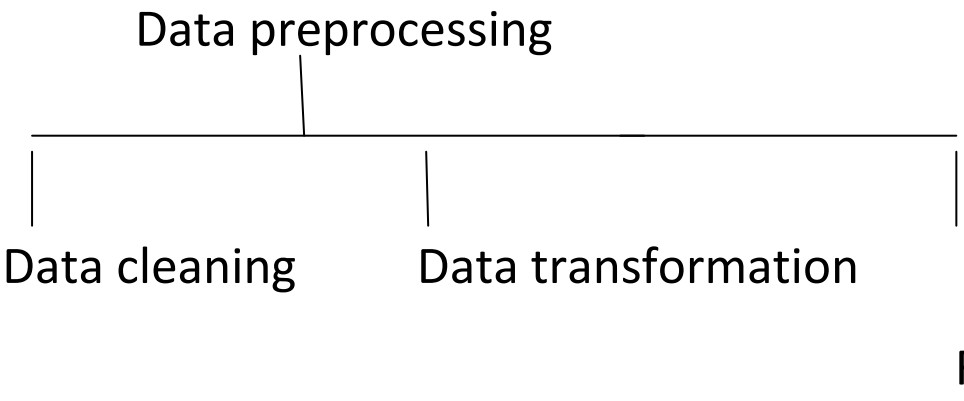
6)Future predictions

7)Model deployment

1)Data preprocessing:

It refers to the cleaning of data to enhance the performance and it is very necessary step to avoid misleading results.Data preprocessing includes cleaning ,normalization,encoding,feature selection.

Data pre processing affects the outcome of the project



- Importing necessary libraries for data analysis such as:

  Import numpy as np

  Import pandas as pd(to load the dataset)

  Import matplotlib.pyplot as plt(for data visualization)

  Import warnings

  Warnings.filterwarnings('ignore')(to ignore the warnings while executing the code)

  After importing the libraries ,we will load the dataset as:

  df=pd.read_csv('path of the dataset')

  (_csv because the dataset present is in this format ie comma separated values)

Evaluation of Data structure:

In this section we have to closely look at the dataset and need to know the each

and every column in detail for understanding of the input data .visualising the dataset using (df.head)and (df.tail) I observe this is a telecommunication churn dataset.And by the code(df.columns)we can see the names of all the columns and (df.shape)gives the number of rows and columns .

The describe method gives the description of the dataset (count,mean,standard deviation,min value,maximum value,and quartiles(25%,50%,75%)).By executing (df.dtypes)we presume that there are so many categorical columns along with numerical columns.

It is very important to keep an eye on the missing values because it mess up the model building and accuracy of that model.So it is necessary to deal with

missing values before selecting the model.By executing (df.isnull().sum()) we can check for nulls .Threre seems no null values in this dataset.Lets move further:

Identifying unique values:

As I observed that column (Total Charges) has continuous data but it is showing object data type .lets handle by seeing unique values by df['TotalCharges'].unique()

So nunique returned the number of unique values for each column.

So checked for white spaces in this column by: df.loc[df['TotalCharges']==" "].

And replaced the white space with nullvalue so that we can treat them using any imputer method .I did that using mean.

Convert the object column datatype ie TotalCharges to integer.

Label encoding:isn pivotal as it encode categorical columns into numeric one as computer only understands numeric values.

2)Data Evaluation:

Exploratory Data Analysis: exploring and visualising our data by distributing the dataset into categorical,ordinal and continuous and visualising it by plotting count plot as:

```
ax=sns.countplot(x='gender',data=df_nominal
)

print(df_nominal["gender"].value_counts())
```

dataframe for categorical columns

df[['gender','SeniorCitizen','Partner','Dependents','PhoneService','MultipleLines','InternetService','OnlineSecurity','OnlineBackup','DeviceProtection','Tec

hSupport','StreamingTV','StreamingMovies','Contr
act','PaperlessBilling','PaymentMethod','Churn']]

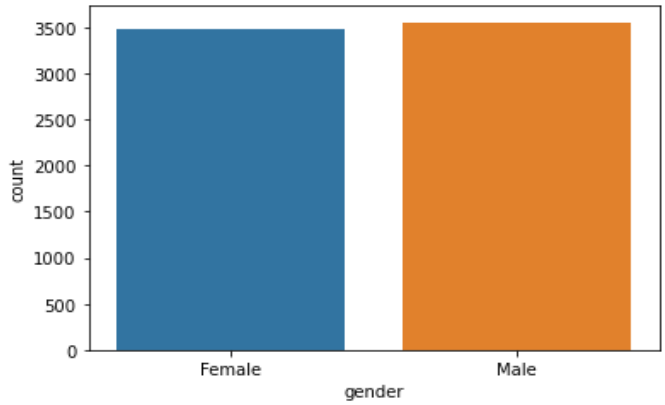
File Edit View Insert Cell Kernel Widgets Help

Run Code

```
In [17]: #visualiation of data for categorical data

In [18]: ax=sns.countplot(x='gender',data=df_nominal)
         print(df_nominal["gender"].value_counts())
```

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



Few observations:

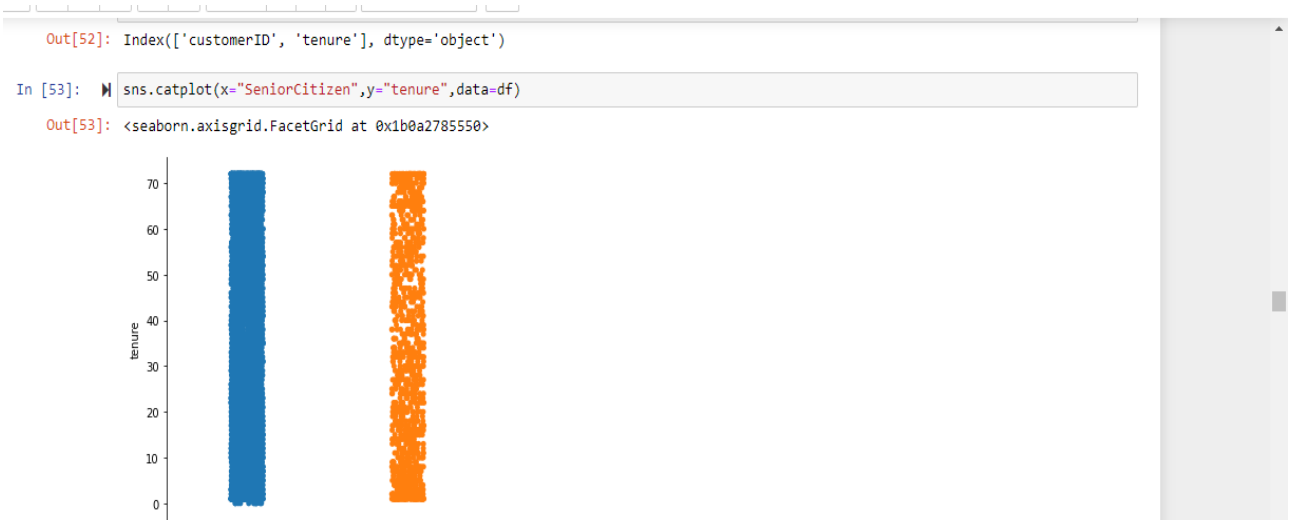
- Gender distribution shows that there are almost equal number of males and females.
- Most of the customers in the dataset are younger people.
- Both singles and those having partners use telecommunication services.
- Not many customers are dependents.

- Most of the customers seems to have phone service and 75% of them have opted for paperless billing.
- Very less people shift from one network to another.
- Most of the people prefer to pay month to month and some people choose two year at once might be because of some discount scheme.
- More than half of people use fiber optic internet service and most of them use DSL and very less doesn’t use internet services.
- The persons who are not using internet services can not take benefits like(online security,online backup,techsupprt,streaming tv,streaming movies)as only those people use these features who uses internet service.
- It is observed that very less peoples move from one phone service to another .This shows that peoples might be satisfied with that phone service.
- It is concluded that majority of the customers in the customers in the dataset are non-senior citizens.

- Out of all senior citizens customers,more than 40% churned.while among younger customers,the churn percentage is less than one-fourth.
- Hence ,senior citizens tend to churn more than younger customers.
- Customers who have availed online backup, device protection,technical support and online security features are a minority.

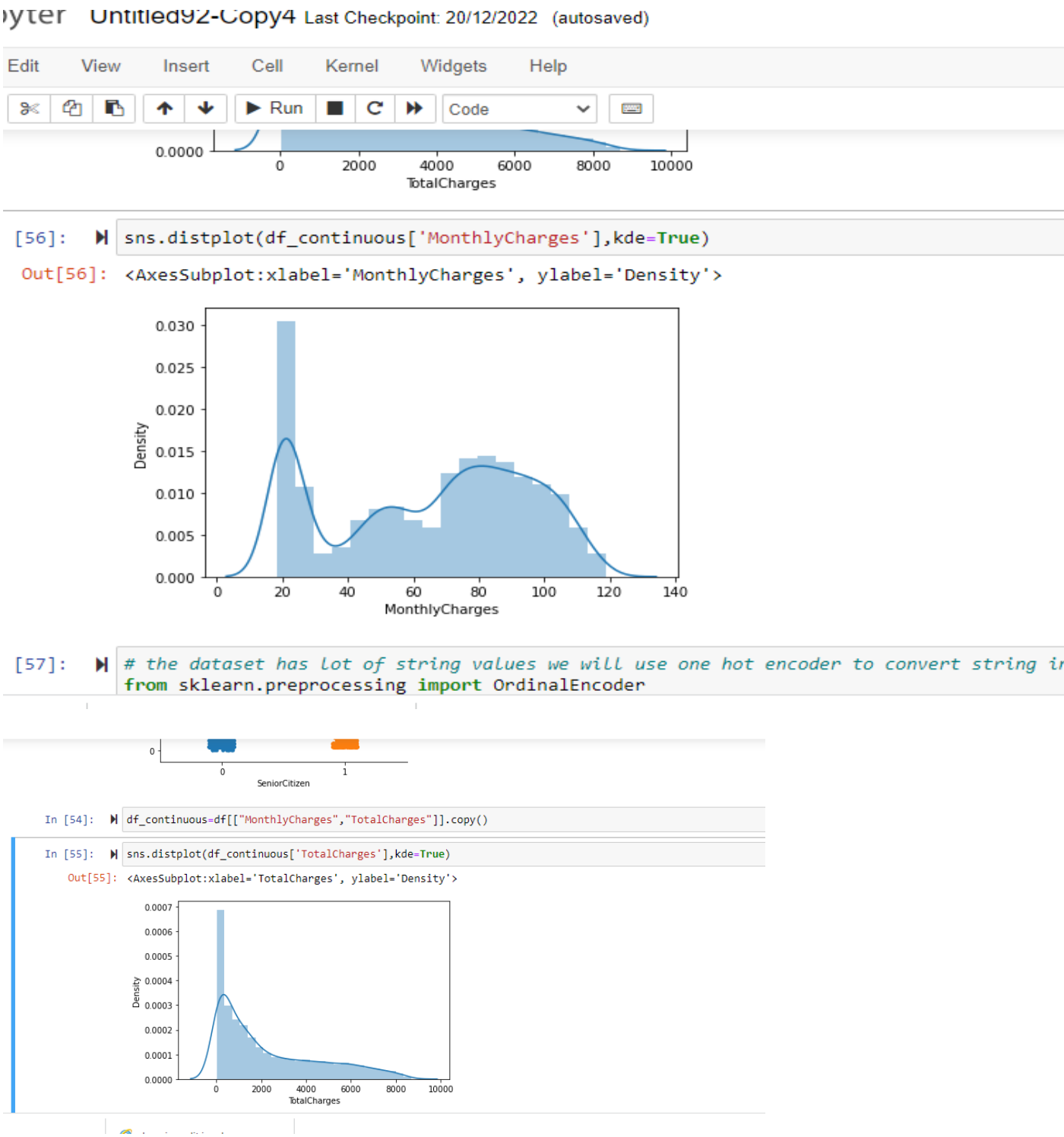
=>Data visualisation for ordinal data

[cat plot for tenure and senior citizen]



Data visualisation for continuous data

Dist plot for two continuous data columns[Total Charges and Monthly charges]



- It is observed that majority of the customers have lower total charges.

Encoding of columns with object datatype into numeric data type using ordinal encoder:

This step is essential because it encodes string columns into numeric.Because computer only understands numeric data.

Correlation:

Lets see the correlation among the features.

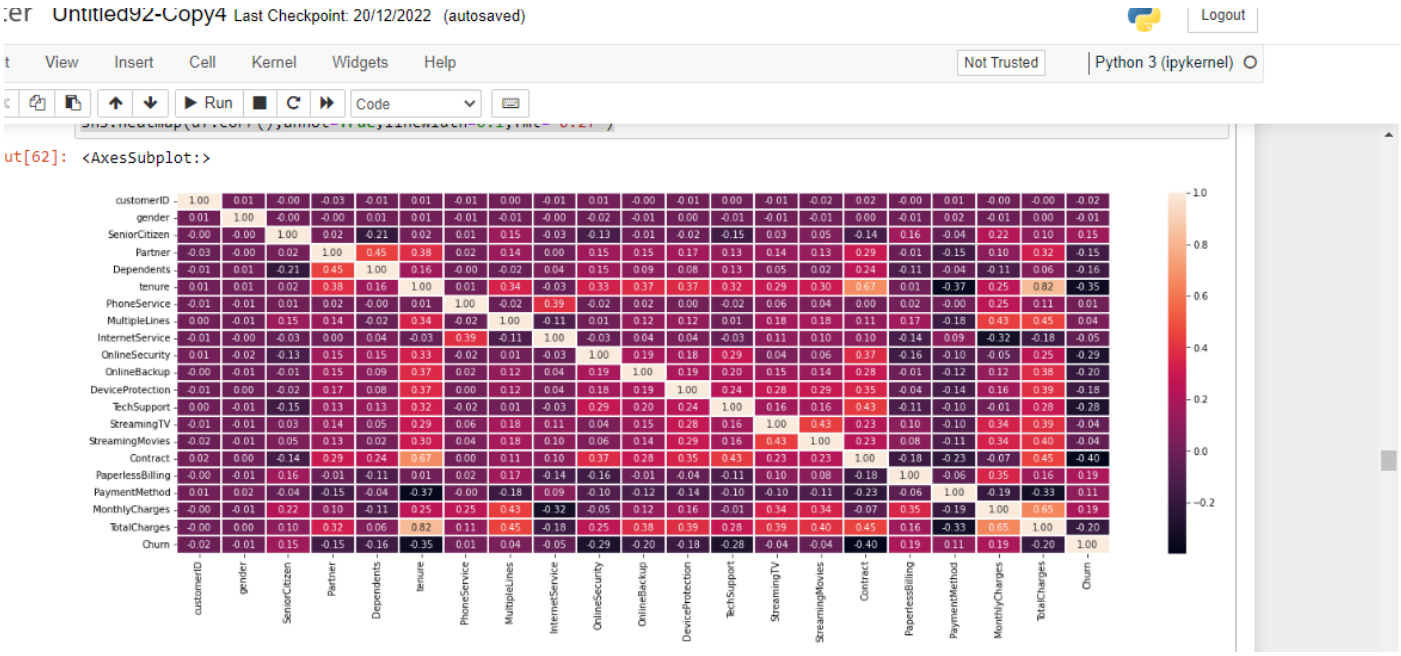
Lets see the correlation value of the column Total Charges using:

df.corr()['TotalCharges'].sort_values()

This will give all the values of the column Total Charges and .sort _values will sort the value in ascending order.

Plotting the heatmap to check the correlation among the features by the code:

sns.heatmap(df.corr(),annot=True,linewidth=0.1,fmt="0.2f")



Correlation matrix helps us to discover the relationship between

Independent variables in the dataset

Observations:

- It is observed that there seems strong correlation between tenure and total charges .
- But I am not deleting any column.

Remarks of Exploratory data analysis:

- The dataset does not have any missing values
- There is strong correlation between tenure and total charges.
- Imbalanced dataset.

- Most of the customers in the dataset are younger people.
- Most of the customers have month to month subscription so have a high probability of churn.

Checking the skewness in the dataset:

Skewness gives the direction of the outliers if it is right skewed,most of the outliers are present on the right side while if it is left skewed ,most of the outliers will be on the left side of the distribution.

- Df.skew():

  Givesthe skewness in numbers.

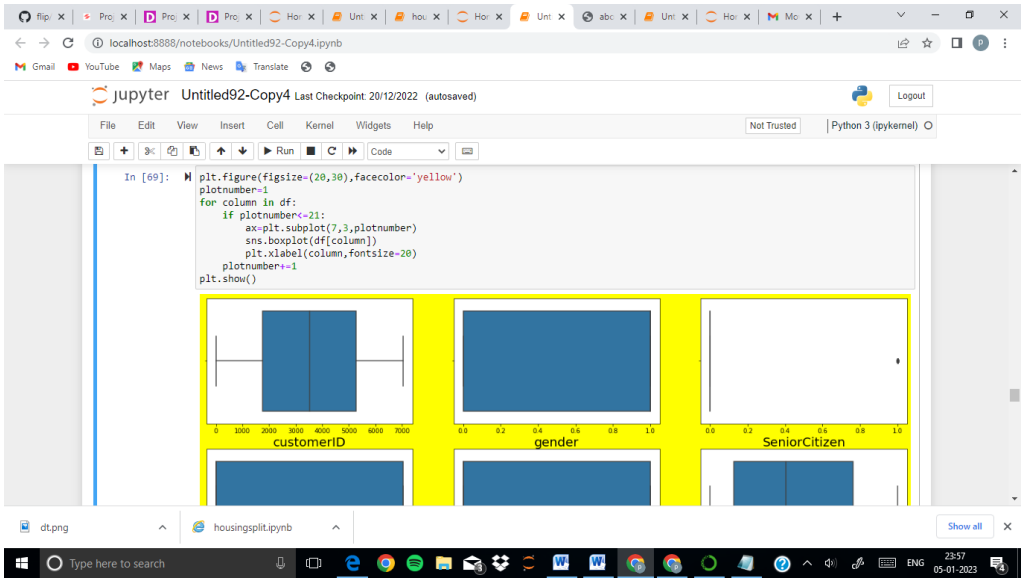
  keeping +/-0.5 as the range for skewness,we will treat the skewness only for int and float datatype originally.

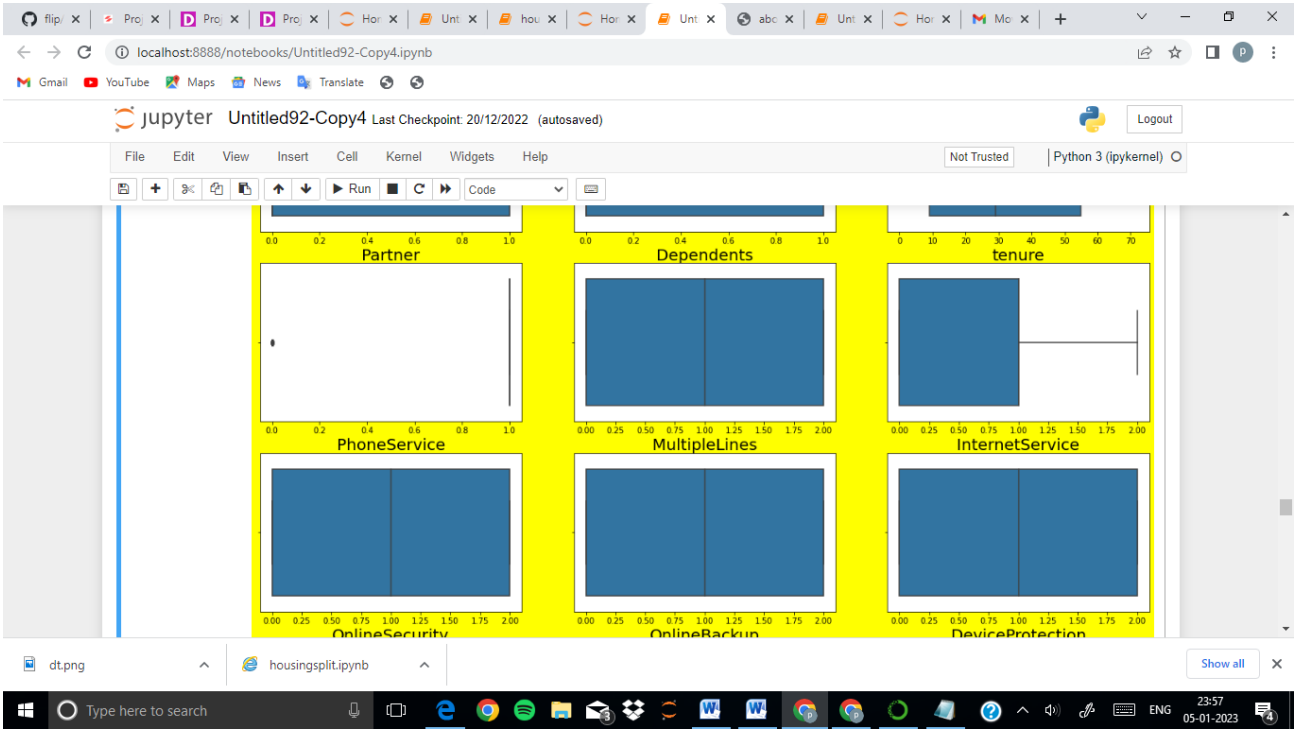
```
customerID          0.000000
gender             -0.019031
SeniorCitizen       1.833633
Partner             0.067922
Dependents          0.875199
tenure              0.239540
PhoneService       -2.727153
MultipleLines       0.118719
InternetService     0.205423
OnlineSecurity      0.416985
OnlineBackup        0.182930
DeviceProtection    0.186847
TechSupport         0.402365
StreamingTV         0.028486
StreamingMovies     0.014657
Contract            0.630959
PaperlessBilling   -0.375396
PaymentMethod      -0.170129
```

```
MonthlyCharges      -0.220524
TotalCharges         0.962394
Churn                1.063031
```

The above values are the skewness values in the dataset.

Plotting the boxplot to check for outliers





since the column senior citizen is nominal type we cannot consider it as outliers.and not treat it.

- Treating the outliers:

  Treating the outliers using zscore

  Z_score measures how many standard deviations a data point is from the mean in the distribution.

  Mean is always 0and standard deviation is1

  For using z score we will import library as :

  [from scipy.stats import zscore]

  It is important to normalize the variables before conducting any machine learning.

- Splitting the dataset into x and y values(features and label)

  x=df.drop("TotalCharges",axis=1)

  y=df["TotalCharges"]

  3)Model selection:

  Model selection is a process to compare the relative value

Of different models and determine which one is best fit for the data analysed.

- Factors for the selection of model

  1)Accuracy of the models : selecting the model according to the available data .

  2)The relevance of the methods:the problem statement must be solved with a suitable method.

  3)Accuracy of the data:the accuracy of the data and the accuracy of methods must match.

  4)Data availability:what data is available in due time.

How to choose a good model:

- Performance of the model
- Explainability of results
- Models complexity
- The size of the dataset
- Training time
- Inference time

How to choose which type of the algorithm is this:

The regression problem is when the output variable is real or continuous value such as salary etc .The difference between the regression and classification algogorithm is that regression algorithm are used to determine continuous values such as price ,income etc whereas classification algorithm are used to forecast discrete class labels.

This is a regression problem because as the label(Total charges) is a continuous data column .There are so many regression algorithms such as

- Linear regression
- Decision tree
- Support vector regression
- Lasso regression
- Random forest regressor

Generate training and test datasets

Train/test is a method to measure the accuracy of the model.It is called train/test because we divide the dataset into two set .One is the training dataset and another is

testing dataset .80% for training and 20% for testing . The main difference between training data is that data which is used to train the model whereas testing data is used to check the accuracy of the model. For training the dataset importing necessary library.

from sklearn.model_selection import train_test_split

⇨ Model choosen in this dataset

For this we need to import necessary libraries

from sklearn.linear_model import LinearRegression

⇨ Initialising the model is necessary

lr=LinearRegression()

from sklearn.linear_model import Lasso

4)Model evaluation:

Lets fit the selected model(linear regression) in this case on the training dataset and evaluate the results.

1)[for i in range(0,50):

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=i)

```
    lr.fit(x_train,y_train)
    pred_train=lr.predict(x_train)
    pred_test=lr.predict(x_test)
    print(f"at random state{i},the training score
is:{r2_score(y_train,pred_train)}")
    print(f"at random state{i},the testing score
is:{r2_score(y_test,pred_test)}")
    print("\n")]
```

2)
```
parameters={'alpha':[.0001,.001,.01,.01,1],
        'random_state':list(range(0,10))}
ls=Lasso()
clf=GridSearchCV(ls,parameters)
clf.fit(x_train,y_train)
print(clf.best_params_)
```

3)
```
from sklearn.ensemble import RandomForestRegressor
```

```
rf=RandomForestRegressor(criterion='mse',max_features='au
to')

rf.fit(x_train,y_train)

rf.score(x_train,y_train)

pred_decision=rf.predict(x_test)

rfs=r2_score(y_test,pred_decision)

print('r2score',rfs*100)

cvs=cross_val_score(rf,x,y,cv=5)

cvs_score=cvs.mean()

print("cross val score",cvs_score*100)
```

Cross validation:

Model evaluation is commonly done through cross validation .It is used to test the ability of a model to predict new data.Also we can solve the problems like overfitting .

In cross validation the whole dataset is splitted into parts.This reduces bias as we are using most of the data for fitting .Variance problem occurs when we got good accuracy .

(from sklearn.model_selection import cross_val_score)

The above code is used to import the library for cross validation.

```
for i in range(2,10):
    cv_score=cross_val_score(lr,x,y,cv=i)
    cv_mean=cv_score.mean()
    print(f"at cross fold{i}the cv score is{cv_mean}and accuracy_score for training is {train_accuracy}and accuracy for testing is {test_accuracy}")
    print("\n")
```

Regularization technique:

```
ls=Lasso(alpha=1,random_state=0)
ls.fit(x_train,y_train)
ls_training=ls.score(x_train,y_train)
pred_ls=ls.predict(x_test)
ls_training*100
pred_ls=ls.predict(x_test)
lss=r2_score(y_test,pred_ls
```

lss*100

regularization is the technique that prevents overfitting.

=>Accuracy score

It is used to measure the model performance.It is an evaluation metric that measures the number of predictions .For this importing the necessary library.

5)model improvement:

Model improvement basically involveschoosing the best parameters for the model.

So hyperparameter tuning is used to increase the accuracy of the model.

Grid searchcv is one of the technique used for hyper parameter tuning.

We need to feed the required parameters .Best parameters are extracted and predictions are made.

Following is the code for that:

```
parameters={'criterion':['mse','mae'],
        'max_features':["auto","sqrt","log2"]}
rf=RandomForestRegressor()
```

clf=GridSearchCV(rf,parameters)

clf.fit(x_train,y_train)

print(clf.best_params_)

6)Future predictions :

Comparing the predictions against test dataie comparing the train and test score .

7)saving of model:

So model deployment is the last step of model.This makes the model prediction available to users .

For this it is necessary to import the library PICKLE.

[pickle.dump(rf,open('churn','wb'))

loaded_model=pickle.load(open('churn','rb'))]

Displaying the final results usingn the below code:

result=loaded_model.score(x_test,y_test)

print(result*100)

⇨ Conclusion

I tried to make a customer churn dataset article with a score of 99%

==========thank you==========