consumer-churn

September 14, 2025

1 Customer Chrun Project

I developed a comprehensive Customer Churn Prediction project involving end-to-end data analysis and modeling. The process included data manipulation (feature extraction, filtering, and sampling), data visualization (bar plots, histograms, scatter plots, and box plots) to uncover customer behavior patterns, and implementation of multiple predictive models. I applied Linear Regression to analyze relationships, Logistic Regression for churn classification, and advanced models like Decision Trees and Random Forests to enhance accuracy. This project demonstrates proficiency in data wrangling, visualization, and machine learning techniques to derive actionable insights for reducing customer churn.

1.1 Data Manipulation:

a. Extract the 5th column & store it in 'customer_5'

b. Extract the 15th column & store it in 'customer_15'

c. Extract all the male senior citizens whose Payment Method is Electronic check & store the result in 'senior_male_electronic'

d. Extract all those customers whose tenure is greater than 70 months or their Monthly charges is more than 100$ & store the result in 'customer_total_tenure'

e. Extract all the customers whose Contract is of two years, payment method is Mailed

f. check & the value of Churn is 'Yes' & store the result in 'two_mail_yes'
Extract 333 random records from the customer_churndataframe& store the result in 'customer_333'

g. Get the count of different levels from the 'Churn' column

[2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[3]:
```
customer_churn = pd.read_csv("customer_churn.csv")
```

[4]:
```
customer_churn.head()
```

[4]:
```
   customerID  gender  SeniorCitizen Partner Dependents  tenure PhoneService  \
0  7590-VHVEG  Female              0     Yes         No       1           No
1  5575-GNVDE    Male              0      No         No      34          Yes
2  3668-QPYBK    Male              0      No         No       2          Yes
3  7795-CFOCW    Male              0      No         No      45           No
4  9237-HQITU  Female              0      No         No       2          Yes

      MultipleLines InternetService OnlineSecurity  … DeviceProtection  \
0  No phone service             DSL             No  …               No
1                No             DSL            Yes  …              Yes
2                No             DSL            Yes  …               No
3  No phone service             DSL            Yes  …              Yes
4                No     Fiber optic             No  …               No

  TechSupport StreamingTV StreamingMovies        Contract PaperlessBilling  \
0          No          No              No  Month-to-month              Yes
1          No          No              No        One year               No
2          No          No              No  Month-to-month              Yes
3         Yes          No              No        One year               No
4          No          No              No  Month-to-month              Yes

               PaymentMethod MonthlyCharges  TotalCharges Churn
0           Electronic check          29.85         29.85    No
1               Mailed check          56.95        1889.5    No
2               Mailed check          53.85        108.15   Yes
3  Bank transfer (automatic)          42.30       1840.75    No
4           Electronic check          70.70        151.65   Yes

[5 rows x 21 columns]
```

[5]:
```
c_15=customer_churn.iloc[:,14]
c_15.head()
```

[5]:
```
0    No
1    No
2    No
3    No
4    No
Name: StreamingMovies, dtype: object
```

[6]:
```
c_random=customer_churn[(customer_churn["gender"]=="Male") &␣
 ↪(customer_churn["SeniorCitizen"]==1) &␣
 ↪(customer_churn["PaymentMethod"]=="Electronic check")]
```

[7]:
```
c_random.head()
```

[7]:
```
    customerID gender  SeniorCitizen Partner Dependents  tenure PhoneService  \
20  8779-QRDMV   Male              1      No         No       1           No
55  1658-BYGOY   Male              1      No         No      18          Yes
57  5067-XJQFU   Male              1     Yes        Yes      66          Yes
78  0191-ZHSKZ   Male              1      No         No      30          Yes
91  2424-WVHPL   Male              1      No         No       1          Yes

       MultipleLines InternetService OnlineSecurity  … DeviceProtection  \
20  No phone service             DSL             No  …               Yes
55               Yes     Fiber optic             No  …                No
57               Yes     Fiber optic             No  …               Yes
78                No             DSL            Yes  …                No
91                No     Fiber optic             No  …                No

   TechSupport StreamingTV StreamingMovies        Contract PaperlessBilling  \
20          No          No              Yes  Month-to-month               Yes
55          No         Yes              Yes  Month-to-month               Yes
57         Yes         Yes              Yes        One year               Yes
78          No         Yes              Yes  Month-to-month               Yes
91         Yes          No               No  Month-to-month                No

       PaymentMethod MonthlyCharges  TotalCharges Churn
20  Electronic check          39.65         39.65   Yes
55  Electronic check          95.45       1752.55   Yes
57  Electronic check         108.45       7076.35    No
78  Electronic check          74.75        2111.3    No
91  Electronic check          74.70          74.7    No

[5 rows x 21 columns]
```

[8]:
```
c_random=customer_churn[(customer_churn["tenure"]>70) |␣
↪(customer_churn["MonthlyCharges"]>100)]
```

[9]:
```
c_random.head()
```

[9]:
```
    customerID  gender  SeniorCitizen Partner Dependents  tenure PhoneService  \
8   7892-POOKP  Female              0     Yes         No      28          Yes
12  8091-TTVAX    Male              0     Yes         No      58          Yes
13  0280-XJGEX    Male              0      No         No      49          Yes
14  5129-JLPIS    Male              0      No         No      25          Yes
15  3655-SNQYZ  Female              0     Yes        Yes      69          Yes

   MultipleLines InternetService OnlineSecurity  … DeviceProtection  \
8            Yes     Fiber optic             No  …               Yes
12           Yes     Fiber optic             No  …               Yes
13           Yes     Fiber optic             No  …               Yes
14            No     Fiber optic            Yes  …               Yes
```

```
15           Yes     Fiber optic            Yes  …              Yes

   TechSupport StreamingTV StreamingMovies        Contract PaperlessBilling  \
8          Yes         Yes              Yes  Month-to-month              Yes
12          No         Yes              Yes        One year               No
13          No         Yes              Yes  Month-to-month              Yes
14         Yes         Yes              Yes  Month-to-month              Yes
15         Yes         Yes              Yes        Two year               No

                PaymentMethod MonthlyCharges  TotalCharges Churn
8            Electronic check         104.80       3046.05   Yes
12   Credit card (automatic)         100.35        5681.1    No
13 Bank transfer (automatic)         103.70        5036.3   Yes
14           Electronic check         105.50       2686.05    No
15   Credit card (automatic)         113.25       7895.15    No

[5 rows x 21 columns]
```

[10]:
```
c_random=customer_churn[(customer_churn["Contract"]=="Two year") &␣
↪(customer_churn["PaymentMethod"]=="Mailed check") &␣
↪(customer_churn["Churn"]=="Yes")]
c_random.head()
```

[10]:
```
      customerID  gender  SeniorCitizen Partner Dependents  tenure  \
268   6323-AYBRX    Male              0      No         No      59
5947  7951-QKZPL  Female              0     Yes        Yes      33
6680  9412-ARGBX  Female              0      No        Yes      48

     PhoneService MultipleLines InternetService       OnlineSecurity  …  \
268           Yes            No              No  No internet service  …
5947          Yes           Yes              No  No internet service  …
6680          Yes            No     Fiber optic                  No  …

        DeviceProtection          TechSupport          StreamingTV  \
268   No internet service  No internet service  No internet service
5947  No internet service  No internet service  No internet service
6680                  Yes                  Yes                  Yes

          StreamingMovies  Contract PaperlessBilling PaymentMethod  \
268   No internet service  Two year               No  Mailed check
5947  No internet service  Two year              Yes  Mailed check
6680                   No  Two year              Yes  Mailed check

     MonthlyCharges  TotalCharges Churn
268           19.35        1099.6   Yes
5947          24.50         740.3   Yes
6680          95.50       4627.85   Yes
```

```
[3 rows x 21 columns]
```

[11]:
```
c333=customer_churn.sample(n=333)
c333.head()
```

[11]:
```
      customerID  gender  SeniorCitizen Partner Dependents  tenure  \
2182  2530-FMFXO    Male              0     Yes        Yes      56
3195  0570-BFQHT  Female              0      No         No       9
5409  6005-OBZPH  Female              1      No         No      26
2991  2207-QPJED  Female              1     Yes         No      37
1816  1663-MHLHE    Male              0      No         No       1

     PhoneService MultipleLines InternetService       OnlineSecurity  …  \
2182          Yes           Yes     Fiber optic                   No  …
3195          Yes            No     Fiber optic                   No  …
5409          Yes           Yes     Fiber optic                   No  …
2991          Yes            No     Fiber optic                   No  …
1816          Yes            No              No  No internet service  …

         DeviceProtection          TechSupport          StreamingTV  \
2182                  Yes                  Yes                  Yes
3195                  Yes                   No                   No
5409                   No                   No                  Yes
2991                  Yes                  Yes                  Yes
1816  No internet service  No internet service  No internet service

          StreamingMovies        Contract PaperlessBilling     PaymentMethod  \
2182                  Yes        Two year              Yes  Electronic check
3195                   No  Month-to-month               No  Electronic check
5409                   No  Month-to-month              Yes  Electronic check
2991                   No  Month-to-month               No  Electronic check
1816  No internet service  Month-to-month               No      Mailed check

     MonthlyCharges  TotalCharges Churn
2182         103.20       5873.75    No
3195          80.55         653.9    No
5409          89.15       2277.65   Yes
2991          90.00       3371.75    No
1816          19.20          19.2    No

[5 rows x 21 columns]
```

[12]:
```
customer_churn['Churn'].value_counts()
```

[12]:
```
Churn
No     5174
```

```
Yes    1869
Name: count, dtype: int64
```

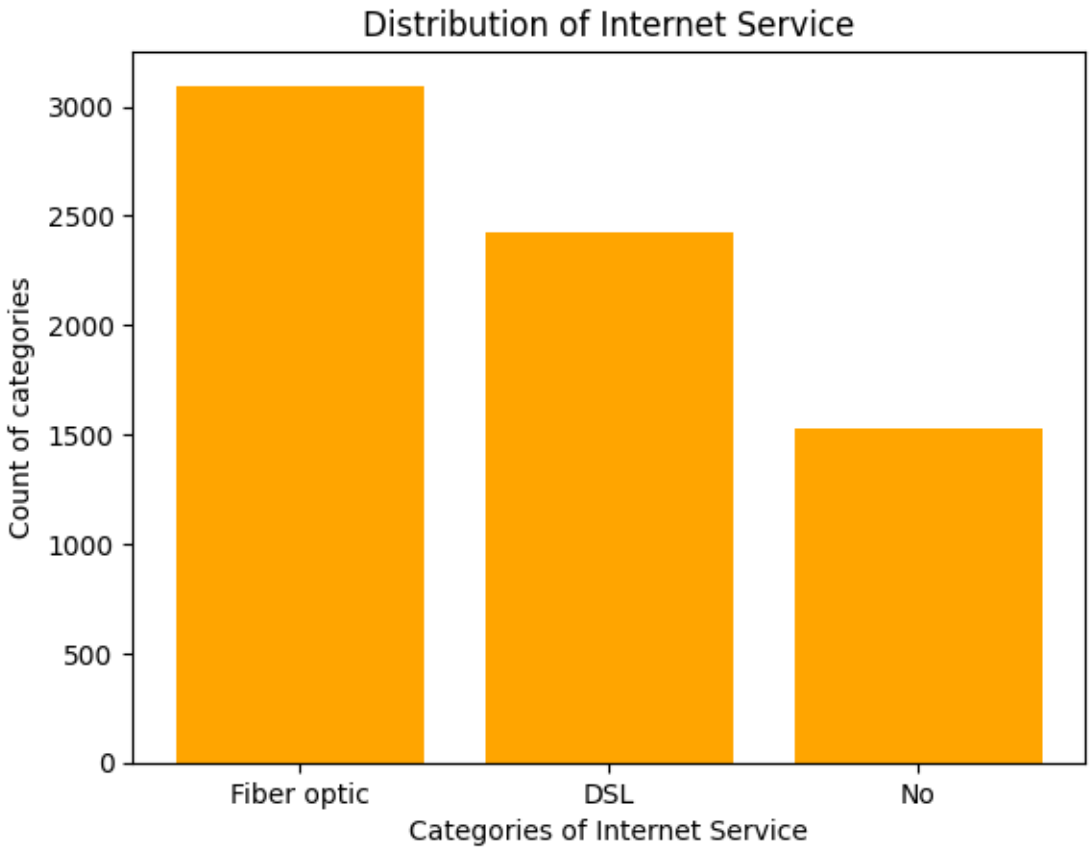
1.2 Data Visualization

1.2.1 a. Build a bar-plot for the 'InternetService' column:

i. Set x-axis label to 'Categories of Internet Service'
ii. Set y-axis label to 'Count of Categories'
iii. Set the title of plot to be 'Distribution of Internet Service'
iv. Set the color of the bars to be 'orange'

[13]:
```
#bar-plot for 'InternetService' column
plt.bar(customer_churn['InternetService'].value_counts().keys().
 ↪tolist(),customer_churn['InternetService'].value_counts().
 ↪tolist(),color='orange')
plt.xlabel('Categories of Internet Service')
plt.ylabel('Count of categories')
plt.title('Distribution of Internet Service')
```

[13]:
```
Text(0.5, 1.0, 'Distribution of Internet Service')
```

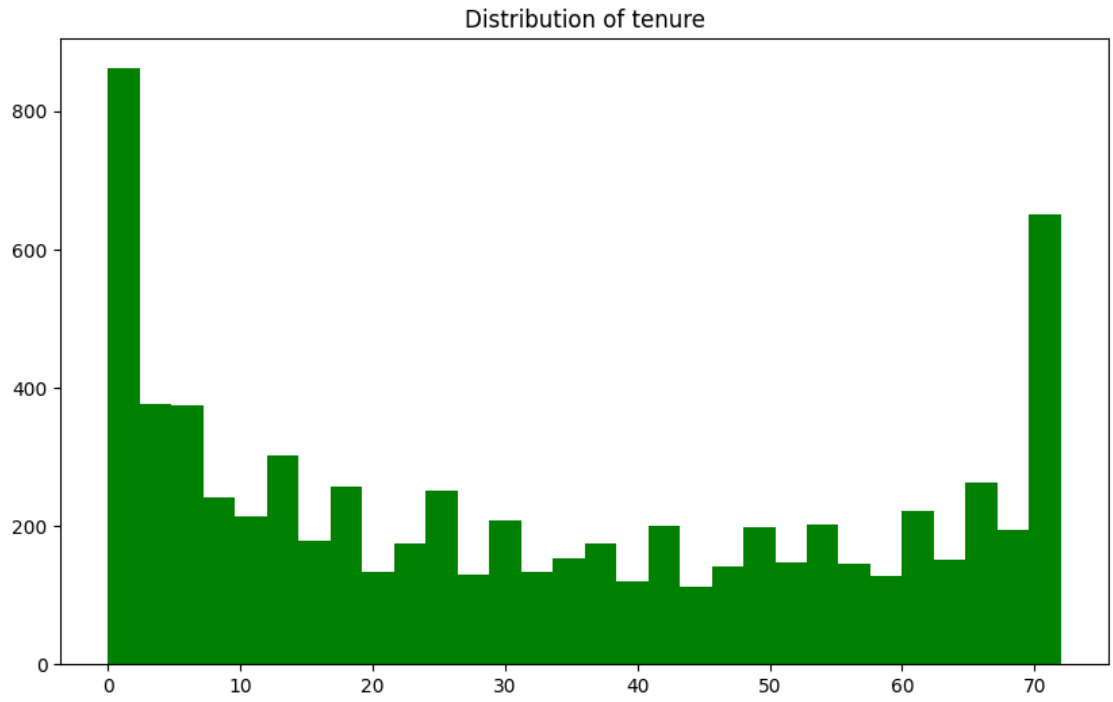


1.2.2 b. Build a histogram for the 'tenure' column:

i. Set the number of bins to be 30
ii. Set the color of the bins to be 'green'
iii. Assign the title 'Distribution of tenure'

[18]:
```
#histogram for 'tenure' column
plt.figure(figsize=(10,6))
plt.hist(customer_churn['tenure'],color='green',bins=30)
plt.title('Distribution of tenure')
```

[18]:
```
Text(0.5, 1.0, 'Distribution of tenure')
```



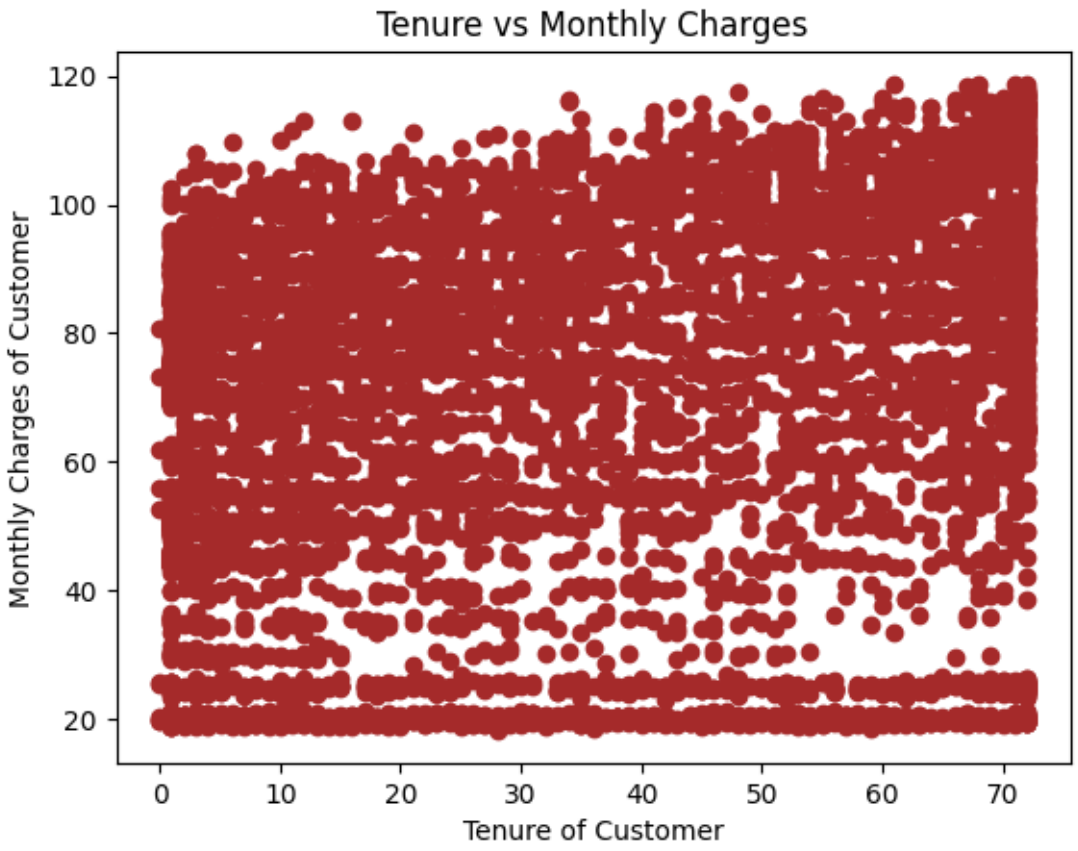
1.2.3 c. Build a scatter-plot between 'MonthlyCharges' & 'tenure'. Map 'MonthlyCharges' to the y-axis & 'tenure' to the 'x-axis':

i. Assign the points a color of 'brown'
ii. Set the x-axis label to 'Tenure of customer'
iii. Set the y-axis label to 'Monthly Charges of customer'
iv. Set the title to 'Tenure vs Monthly Charges'

[109]:
```
#scatterplot
```

```
plt.
  ↪scatter(x=customer_churn['tenure'],y=customer_churn['MonthlyCharges'],color='brown')
plt.xlabel('Tenure of Customer')
plt.ylabel('Monthly Charges of Customer')
plt.title('Tenure vs Monthly Charges')
```

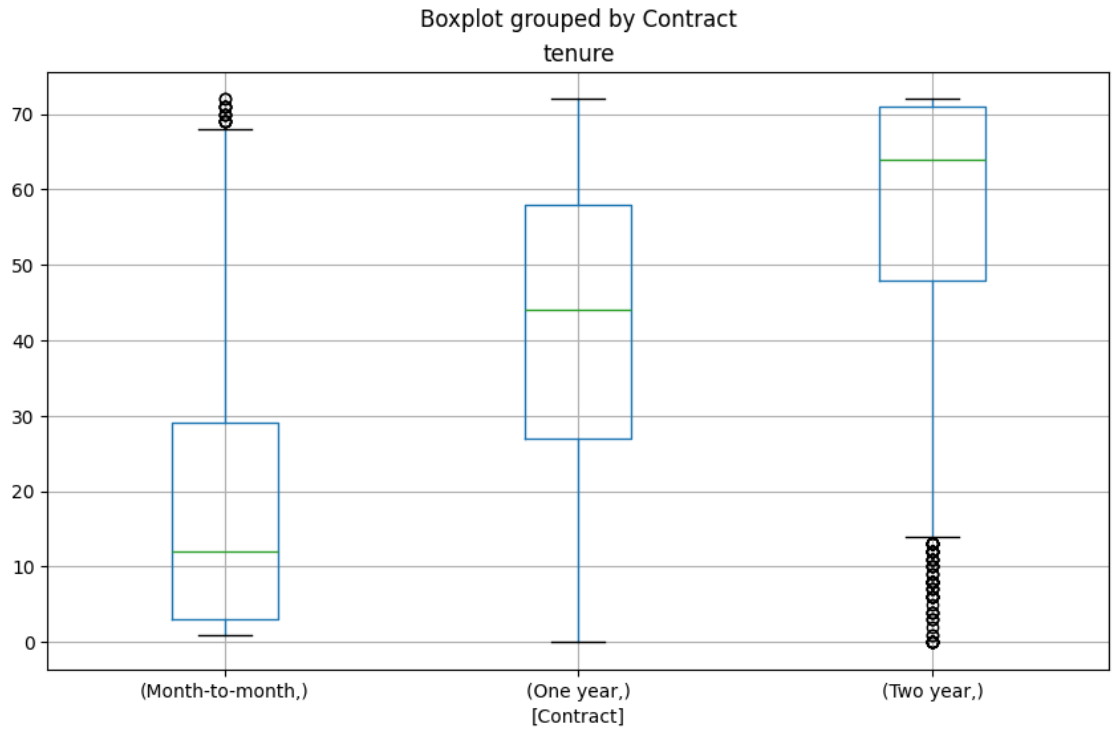
[109]: Text(0.5, 1.0, 'Tenure vs Monthly Charges')



1.2.4 d. Build a box-plot between 'tenure' & 'Contract'. Map 'tenure' on the y-axis & 'Contract' on the x-axis.

[20]:
```
#Box-plot
customer_churn.boxplot(column='tenure',by=['Contract'],figsize=(10,6))
```

[20]: <Axes: title={'center': 'tenure'}, xlabel='[Contract]'>



1.3 Linear Regression Model

1.3.1 a. Build a simple linear model where dependent variable is 'MonthlyCharges' and independent variable is 'tenure'

i. Divide the dataset into train and test sets in 70:30 ratio.
ii. Build the model on train set and predict the values on test set
iii. After predicting the values, find the root mean square error
iv. Find out the error in prediction & store the result in 'error'
v. Find the root mean square error

[27]:
```
from sklearn import linear_model
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

[37]:
```
x=pd.DataFrame(customer_churn['tenure'])
y=customer_churn['MonthlyCharges']
```

[38]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)
```

[40]:
```
simpleLinearRegression = LinearRegression()
simpleLinearRegression.fit(x_train,y_train)
```

[40]:
```
LinearRegression()
```

[45]:
```
#predicting the values
y_pred = simpleLinearRegression.predict(x_test)
y_pred[:5],y_test[:5]
```

[45]:
```
(array([60.95089608, 72.98096699, 59.1903979 , 55.66940154, 71.51388517]),
 2200     58.20
 4627    116.60
 3225     71.95
 2828     20.45
 3768     77.75
 Name: MonthlyCharges, dtype: float64)
```

[42]:
```
from sklearn.metrics import mean_squared_error
```

[43]:
```
np.sqrt(mean_squared_error(y_test,y_pred))
```

[43]:
```
29.394584027273893
```

1.4 Logistic Regression Model

1.4.1 a. Build a simple logistic regression modelwhere dependent variable is 'Churn' & independent variable is 'MonthlyCharges'

i. Divide the dataset in 65:35 ratio
ii. Build the model on train set and predict the values on test set
iii. Build the confusion matrix and get the accuracy score

[46]:
```
x=customer_churn[["MonthlyCharges"]]
y=customer_churn[["Churn"]]
```

[48]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.
↪35,random_state=0)
```

[49]:
```
from sklearn.linear_model import LogisticRegression

log_model = LogisticRegression()
```

[ ]:
```
log_model.fit(x_train,y_train)
```

```
c:\Users\usymr\AppData\Local\Programs\Python\Python312\Lib\site-
packages\sklearn\utils\validation.py:1183: DataConversionWarning: A column-
vector y was passed when a 1d array was expected. Please change the shape of y
to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

[ ]:
```
LogisticRegression()
```

[52]:
```
#predicting the values
y_pred = log_model.predict(x_test)
```

[53]:
```
from sklearn.metrics import confusion_matrix,accuracy_score
```

[54]:
```
confusion_matrix(y_test,y_pred),accuracy_score(y_test,y_pred)
```

[54]:
```
(array([[1815,    0],
        [ 651,    0]], dtype=int64),
 0.7360097323600974)
```

[107]:
```
# Manual Representation of above matrix used for accuracy calculation
(1815)/(1815+651)
```

[107]:
```
0.7360097323600974
```

1.5 Logistic Regression Modle with different Train Test size

1.5.1 b. Build a multiple logistic regression model where dependent variable is 'Churn' & independent variables are 'tenure' & 'MonthlyCharges'

i. Divide the dataset in 80:20 ratio
ii. Build the model on train set and predict the values on test set
iii. Build the confusion matrix and get the accuracy score

[58]:
```
x=customer_churn[["MonthlyCharges","tenure"]]
y=customer_churn[["Churn"]]
```

[59]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.
↪35,random_state=0)
```

[60]:
```
from sklearn.linear_model import LogisticRegression

log_model = LogisticRegression()
```

[61]:
```
log_model.fit(x_train,y_train)
```

```
c:\Users\usymr\AppData\Local\Programs\Python\Python312\Lib\site-
packages\sklearn\utils\validation.py:1183: DataConversionWarning: A column-
vector y was passed when a 1d array was expected. Please change the shape of y
to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

[61]:
```
LogisticRegression()
```

[62]:
```
#predicting the values
y_pred = log_model.predict(x_test)
```

[64]:
```
from sklearn.metrics import confusion_matrix,accuracy_score
confusion_matrix(y_test,y_pred),accuracy_score(y_test,y_pred)
```

[64]:
```
(array([[1634,  181],
        [ 364,  287]], dtype=int64),
 0.7789943227899432)
```

[106]:
```
# Manual Representation of above matrix used for accuracy calculation
(1634+287)/(364+181+1634+287)
```

[106]:
```
0.7789943227899432
```

1.6 Decission Tree Model

1.6.1 a. Build a decision tree model where dependent variable is 'Churn' & independent variable is 'tenure'

i. Divide the dataset in 80:20 ratio
ii. Build the model on train set and predict the values on test set
iii. Build the confusion matrix and calculate the accuracy

[73]:
```
x=customer_churn[['tenure']]
y=customer_churn[['Churn']]
```

[74]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20)
```

[75]:
```
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier()
classifier.fit(x_train, y_train)
```

[75]:
```
DecisionTreeClassifier()
```

[76]:
```
y_pred = classifier.predict(x_test)
```

[77]:
```
from sklearn.metrics import classification_report,␣
 ↪confusion_matrix,accuracy_score
print(confusion_matrix(y_test, y_pred))
print(accuracy_score(y_test, y_pred))
```

```
[[941  77]
 [294  97]]
0.7366926898509581
```

1.7 Random Forest Model

1.7.1 a. Build a Random Forest model where dependent variable is 'Churn' & independent variables are 'tenure' and 'MonthlyCharges'

i. Divide the dataset in 70:30 ratio

ii. Build the model on train set and predict the values on test set
iii. Build the confusion matrix and calculate the accuracy

[101]:
```
x=customer_churn[['tenure','MonthlyCharges']]
y=customer_churn['Churn']
```

[102]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20)
```

[103]:
```
from sklearn.ensemble import RandomForestClassifier
clf=RandomForestClassifier(n_estimators=100)
clf.fit(x_train,y_train)
```

[103]:
```
RandomForestClassifier()
```

[104]:
```
y_pred=clf.predict(x_test)
```

[105]:
```
from sklearn import metrics
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.7466288147622427
```