#import required libraries

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from scipy.stats import chi2\_contingency

#read data

data=pd.read\_csv('train.csv')

#get column names from dataframe

col\_list=list(data)

#drop rows with empty values

data.dropna(

axis=0,

how='any',

thresh=None,

subset=None,

inplace=True

)

#replace string data with numbers

uni\_lc\_lis=data['Loyalty\_customer'].unique()

uni\_dic=dict(zip(uni\_lc\_lis, range(len(uni\_lc\_lis))))

data.replace({'Loyalty\_customer':uni\_dic},inplace=True)

uni\_pc\_dic=data['Product\_Category'].unique()

uni\_pc\_dic=dict(zip(uni\_pc\_lis,range(len(uni\_pc\_lis))))

data.replace({'Product\_Category':uni\_pc\_dic},inplace=True)

#drop unnecessary columns

data.drop(['Customer\_name'],axis=1,inplace=True)

data.drop(['Product\_id'],axis=1,inplace=True)

data.drop(['instock\_date'],axis=1,inplace=True)

data.drop( data[ data['Selling\_Price']<0].index , inplace=True)

#plot dependent variable against each independent variable

plt.style.use('ggplot')

fig = plt.figure(figsize = (18, 18))

for i in range(len(col\_list)):

ax = fig.add\_subplot(4, 4, i + 1)

ax.scatter(data[col\_list[i]],data['Selling\_Price'])

ax.set\_ylabel('Sell Price', size = 12)

ax.set\_xlabel(col\_list[i], size = 12)

#find chi score of all features

for i in range(len(col\_list)-1):

del data\_crosstab

data\_crosstab = pd.crosstab(data[col\_list[i]],

data['Selling\_Price'],

margins = False)

chi\_scores\_sn=chi2\_contingency(data\_crosstab)

t.append(chi\_scores\_sn[1])

#convert data

val=data.values

x=val[:,:-1]

y=val[:,-1]

#perform random forest regression

from sklearn.ensemble import RandomForestRegressor

# create regressor object

regressor = RandomForestRegressor(n\_estimators = 100, random\_state = 0)

# fit the regressor with x and y data

regressor.fit(x, y)

#load test data

test=pd.read\_csv('test.csv')

#Perform encoding and dropping

test.drop(['instock\_date'],axis=1,inplace=True)

test.drop(['Customer\_name'],axis=1,inplace=True)

test.replace({'Product\_Category':uni\_pc\_dic},inplace=True)

test.replace({'Loyalty\_customer':uni\_dic},inplace=True)

test\_val=test.values

z=test\_val[:,1:]

#perform prediction on test data

z\_pred=regressor.predict(z)

#write to output

output=pd.DataFrame(data={"Product\_id":test['Product\_id'],"Selling\_Price":z\_pred})

output.to\_csv('Final\_Submission.csv')