import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

dataset = pd.read\_csv("/Housing.csv")

# Printing first 5 records of the dataset

print(dataset.head(5))

dataset.shape

obj = (dataset.dtypes == 'object')

object\_cols = list(obj[obj].index)

print("Categorical variables:",len(object\_cols))

int\_ = (dataset.dtypes == 'int')

num\_cols = list(int\_[int\_].index)

print("Integer variables:",len(num\_cols))

fl = (dataset.dtypes == 'float')

fl\_cols = list(fl[fl].index)

print("Float variables:",len(fl\_cols))

plt.figure(figsize=(12, 6))

sns.heatmap(dataset.corr(),

      cmap = 'BrBG',

      fmt = '.2f',

      linewidths = 2,

      annot = True)

plt.figure(figsize=(18, 36))

plt.title('Categorical Features: Distribution')

plt.xticks(rotation=90)

index = 1

for col in object\_cols:

  y = dataset[col].value\_counts()

  plt.subplot(11, 4, index)

  plt.xticks(rotation=90)

  sns.barplot(x=list(y.index), y=y)

  index += 1

new\_dataset = dataset.dropna()

new\_dataset.isnull().sum()

from sklearn.preprocessing import OneHotEncoder

s = (new\_dataset.dtypes == 'object')

object\_cols = list(s[s].index)

print("Categorical variables:")

print(object\_cols)

print('No. of. categorical features: ',

  len(object\_cols))

OH\_encoder = OneHotEncoder(sparse=False)

OH\_cols = pd.DataFrame(OH\_encoder.fit\_transform(new\_dataset[object\_cols]))

OH\_cols.index = new\_dataset.index

new\_column\_names = OH\_encoder.get\_feature\_names\_out(object\_cols)

OH\_cols.columns = new\_column\_names

df\_final = new\_dataset.drop(object\_cols, axis=1)

df\_final = pd.concat([df\_final, OH\_cols], axis=1)

from sklearn.metrics import mean\_absolute\_error

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

X = df\_final.drop(['price'], axis=1)

Y = df\_final['price']

# Split the training set into

# training and validation set

X\_train, X\_valid, Y\_train, Y\_valid = train\_test\_split(

  X, Y, train\_size=0.8, test\_size=0.2, random\_state=0)

from sklearn import svm

from sklearn.svm import SVC

from sklearn.metrics import mean\_absolute\_percentage\_error

model\_SVR = svm.SVR()

model\_SVR.fit(X\_train,Y\_train)

Y\_pred = model\_SVR.predict(X\_valid)

print(mean\_absolute\_percentage\_error(Y\_valid, Y\_pred))

from sklearn.ensemble import RandomForestRegressor

model\_RFR = RandomForestRegressor(n\_estimators=10)

model\_RFR.fit(X\_train, Y\_train)

Y\_pred = model\_RFR.predict(X\_valid)

mean\_absolute\_percentage\_error(Y\_valid, Y\_pred)

from sklearn.linear\_model import LinearRegression

model\_LR = LinearRegression()

model\_LR.fit(X\_train, Y\_train)

Y\_pred = model\_LR.predict(X\_valid)

print(mean\_absolute\_percentage\_error(Y\_valid, Y\_pred))