import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn import metrics

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

from sklearn.linear\_model import LinearRegression

import colorama

from colorama import Fore, Style

## loading the data from csv file to a Pandas DataFrame

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

## first 5 rows of the dataframe

dataset.head()

## number of rows and columns

dataset.shape

## getting some informations about the dataset

dataset.info()

## check the unique values in catagorical columns(object Dtype)

print(dataset["gender"].unique())

print(dataset["smoker"].unique())

print(dataset["state"].unique())

## checking for missing values

dataset.isnull().sum()

## statistical Measures of the dataset

dataset.describe()

## distribution of age value

sns.set()

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

sns.displot(dataset['age'])

plt.title('Age Distribution')

plt.show()

## Gender column

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

sns.countplot(x = 'gender', data = dataset)

plt.title('Gender Distribution')

plt.show()

dataset['gender'].value\_counts()

## bmi distribution

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

sns.displot(dataset['bmi'])

plt.title('BMI Distribution')

plt.show()

## children column

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

sns.countplot(x = 'children', data = dataset)

plt.title('Children')

plt.show()

## smoker column

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

sns.countplot(x = 'smoker', data = dataset)

plt.title('smoker')

plt.show()

# state column

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

sns.countplot(x='state', data = dataset)

plt.title('state')

plt.show()

## distribution of charges value

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

sns.displot(dataset['charges'])

plt.title('Charges Distribution(Rupees)')

plt.show()

dataset.columns

final\_dataset = dataset[['charges', 'age', 'gender', 'bmi', 'children', 'smoker', 'state']]

final\_dataset.head()

final\_dataset = pd.get\_dummies(final\_dataset, drop\_first = True)

final\_dataset.head()

final\_dataset.corr()

%matplotlib inline

corrmat=final\_dataset.corr()

top\_corr\_features=corrmat.index

plt.figure(figsize=(20,20))

#plot heat map

g=sns.heatmap(final\_dataset[top\_corr\_features].corr(),annot=True,cmap="RdYlGn")

## independent and dependent features

X = final\_dataset.iloc[:,1:]

y = final\_dataset.iloc[:,0]

X.head()

y.head()

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,test\_size = 0.2,random\_state = 2)

print(X.shape, X\_train.shape, X\_test.shape)

## loading the Linear Regression model

regressor = LinearRegression()

regressor.fit(X\_train.values, y\_train)

## prediction on training data

training\_data\_prediction = regressor.predict(X\_train.values)

## R squared value

r2\_train = metrics.r2\_score(y\_train, training\_data\_prediction)

print('R squared value : ', r2\_train)

## prediction on test data

test\_data\_prediction = regressor.predict(X\_test.values)

## R squared value

r2\_test = metrics.r2\_score(y\_test, test\_data\_prediction)

print('R squared value : ', r2\_test)

## age bmi children gender\_male smoker\_yes state\_karnataka state\_tamilnadu state\_telangana

age = int(input("enter the age: "))

gender = input("enter the gender: (male/female) --> ")

bmi = float(input("enter the bmi: "))

children = int(input("enter how many children: "))

smoker = input("is Smoker: (yes/no) --> ")

state = input("enter the state:(andhrapradesh/telangana/karnataka/tamilnadu) --> ")

if gender == "male":

int\_gender = 1

elif gender == "female":

int\_gender = 0

if smoker == "yes":

int\_smoker = 1

elif smoker == "no":

int\_smoker = 0

int\_karnataka = 0

int\_tamilnadu = 0

int\_telangana = 0

if state == "telangana":

int\_telangana = 1

elif state == "karnataka":

int\_karnataka = 1

elif state == "tamilnadu":

int\_tamilnadu = 1

else: # "andhrapradesh"

pass

input\_data = (age,bmi,children,int\_gender,int\_smoker,int\_karnataka,int\_tamilnadu,int\_telangana)

## changing input\_data to a numpy array

input\_data\_as\_numpy\_array = np.asarray(input\_data)

## reshape the array

input\_data\_reshaped = input\_data\_as\_numpy\_array.reshape(1,-1)

prediction = regressor.predict(input\_data\_reshaped)

print(Fore.RED + Style.BRIGHT +'The medical cost is Rs.', prediction[0])