Python Project Submission – Insurance

Submitted by: Mohit Rohilla

Problem Statement:

An insurance agency, ABC Insurance, has a large dataset containing information about their policyholders and claims. They want to perform exploratory data analysis (EDA) on this dataset to gain insights that can help them make better business decisions and improve their operations.

The agency wants to analyze the different body types and the environment that affect the premium. The disease's effect or the cost of treatment differs depending on the circumstances. For example, a smoker's medical insurance premium may be higher than that of a healthy person, because smokers are more likely to develop chronic diseases. The agency wants to analyze the data to research healthcare premium costs.

Objective: To analyze the dataset that will help to create a model that will predict the cost of medical insurance based on various input features

Domain: Healthcare

Dataset: insurance dataset (insurance.csv)

Steps to Be Followed:

1. Import libraries such as Pandas, matplotlib, NumPy, and seaborn and load the insurance dataset
2. Check the shape of the data along with the data types of the column
3. Check missing values in the dataset and find the appropriate measures to fill in the missing values
4. Explore the relationship between the feature and target column using a count plot of categorical columns and a scatter plot of numerical columns
5. Perform data visualization using plots of feature vs feature
6. Check if the number of premium charges for smokers or non-smokers is increasing as they are aging
7. After each step, specify the observations

Solutions:

Task 1: Importing libraries as mentioned in task 1

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

# Task 1: checking default file location
import os
os.getcwd()
# Task 1: changing default file location
os.chdir('C:\\Users\\ml30r\\Downloads')

# Task 1: importing insurance file from downloads folder
insurance = pd.read_csv('insurance.csv')

[6]:
```
# Task 1: Importing libraries as mentioned in task 1
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

[9]:
```
# Task 1: checking default file location
import os
os.getcwd()
```

[9]:
```
'C:\\Users\\ml30r'
```

[10]:
```
# Task 1: changing default file location
os.chdir('C:\\Users\\ml30r\\Downloads')
```

[11]:
```
# Task 1:  importing insurance file from downloads folder
insurance = pd.read_csv('insurance.csv')
```

[9]:
```
insurance
```

[9]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

Task 2: checking shape of the data
insurance.shape

```
[12]: # Task 2: checking shape of the data
      insurance.shape
```

```
[12]: (1338, 7)
```

# Task 2: checking the datatype the data in one formula
print(insurance.dtypes)

```
[20]: # Task 2: checking the datatype the data in one formula
      print(insurance.dtypes)
```

```
age         int64
sex        object
bmi       float64
children    int64
smoker     object
region     object
charges   float64
dtype: object
```

# Task 3: identidfication of missing values
insurance.isnull()

```
[21]: # Task 3: identidfication of missing values
      insurance.isnull()
```

[21]:

	age	sex	bmi	children	smoker	region	charges
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...
1333	False	False	False	False	False	False	False
1334	False	False	False	False	False	False	False
1335	False	False	False	False	False	False	False
1336	False	False	False	False	False	False	False
1337	False	False	False	False	False	False	False

1338 rows × 7 columns

# Task 3: total number of missing values in columns
insurance.isnull().sum(axis=0)

```
[22]: #  Task 3: total number of missing values in columns
      insurance.isnull().sum(axis=0)
```

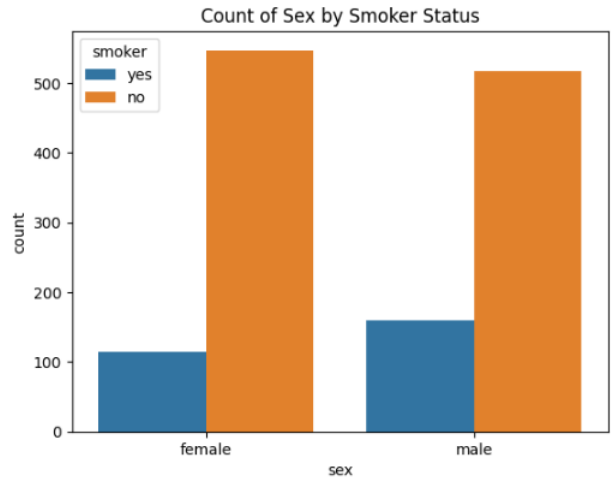
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

Task 4: Explore the relationship between the feature and target column using
A count plot of categorical columns and a scatter plot of numerical columns
features variables are the targets and the rest are feature variables

import matplotlib.pyplot as plt
import seaborn as sns
sns.countplot(x='sex', data=insurance, hue='smoker')
plt.title('Count of Sex by Smoker Status')
plt.show()

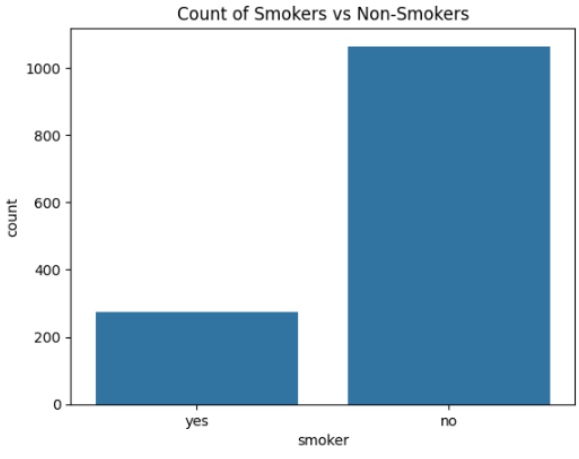
```
# Task 4: Explore the relationship between the feature and target column using
# a count plot of categorical columns and a scatter plot of numerical columns
# features variables are the targets and the rest are feature variables

import matplotlib.pyplot as plt
import seaborn as sns
sns.countplot(x='sex', data=insurance, hue='smoker')
plt.title('Count of Sex by Smoker Status')
plt.show()
```



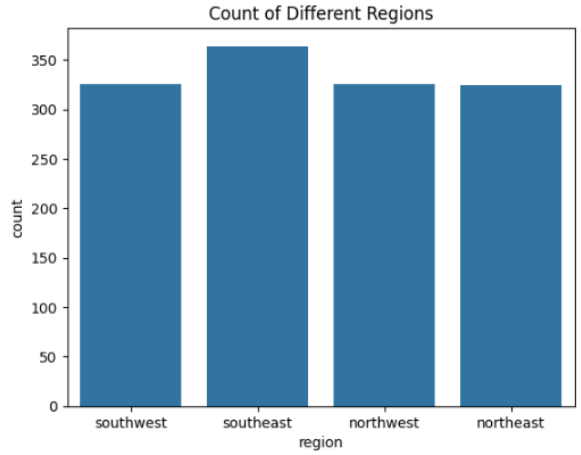
# count of smokers vs non smokers
sns.countplot(x='smoker', data=insurance)
plt.title('Count of Smokers vs Non-Smokers')
plt.show()

```
# count of smokers vs non smokers
sns.countplot(x='smoker', data=insurance)
plt.title('Count of Smokers vs Non-Smokers')
plt.show()
```



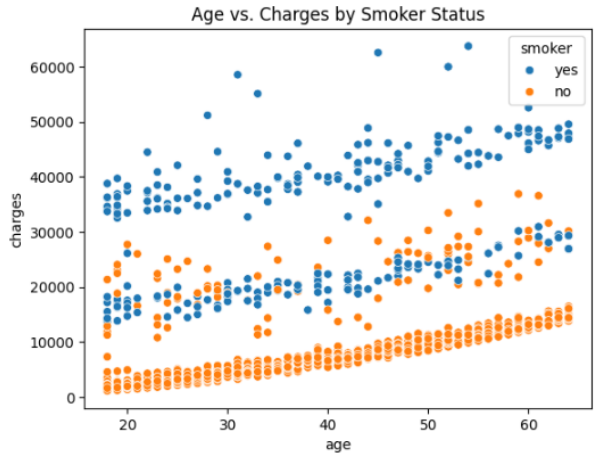
# count of different regions
sns.countplot(x='region', data=insurance)
plt.title('Count of Different Regions')
plt.show()

```
[26]: # count of different regions
sns.countplot(x='region', data=insurance)
plt.title('Count of Different Regions')
plt.show()
```



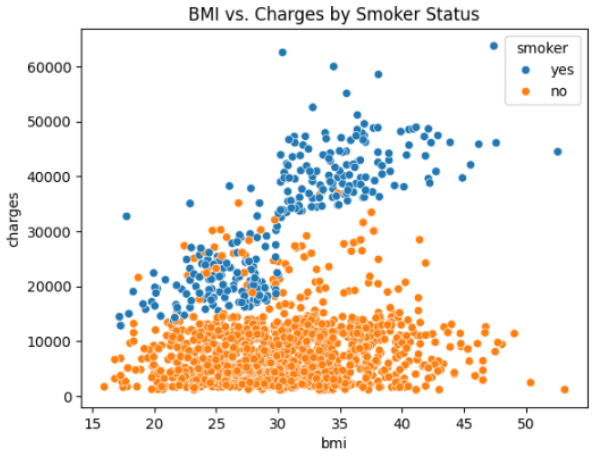
# Scatter plot for age vs. charges
sns.scatterplot(x='age', y='charges', data=insurance, hue='smoker')
plt.title('Age vs. Charges by Smoker Status')
plt.show()

```
[27]: # Scatter plot for age vs. charges
sns.scatterplot(x='age', y='charges', data=insurance, hue='smoker')
plt.title('Age vs. Charges by Smoker Status')
plt.show()
```



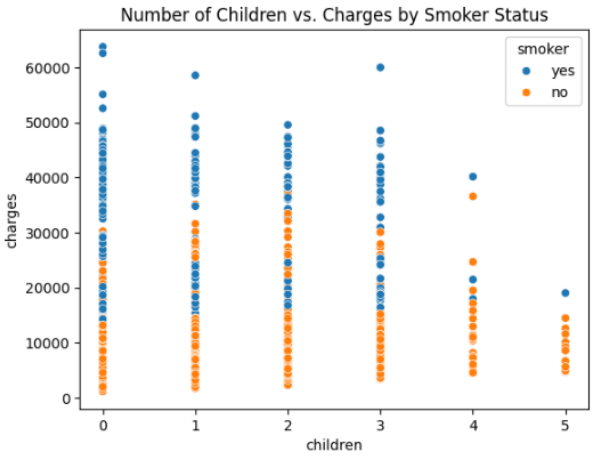
# Scatter plot for BMI vs. charges
sns.scatterplot(x='bmi', y='charges', data=insurance, hue='smoker')
plt.title('BMI vs. Charges by Smoker Status')
plt.show()

[28]:
```
# Scatter plot for BMI vs. charges
sns.scatterplot(x='bmi', y='charges', data=insurance, hue='smoker')
plt.title('BMI vs. Charges by Smoker Status')
plt.show()
```



# Scatter plot for children vs. charges
sns.scatterplot(x='children', y='charges', data=insurance, hue='smoker')
plt.title('Number of Children vs. Charges by Smoker Status')
plt.show()

[29]:
```
# Scatter plot for children vs. charges
sns.scatterplot(x='children', y='charges', data=insurance, hue='smoker')
plt.title('Number of Children vs. Charges by Smoker Status')
plt.show()
```

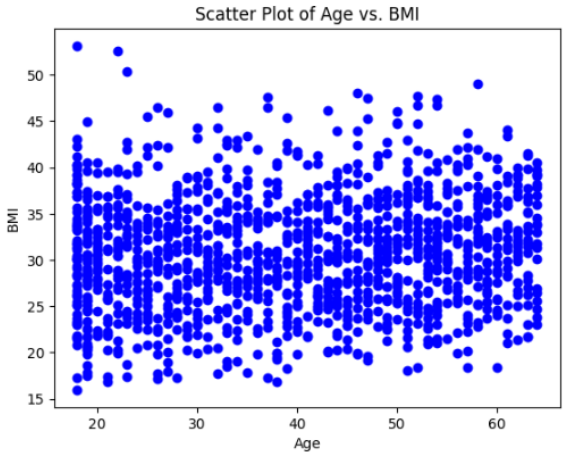


Task 5: Perform data visualization using plots of feature vs feature
Scatter plot for age vs. bmi

```
plt.scatter(insurance.age, insurance.bmi, c='blue')
plt.title('Scatter Plot of Age vs. BMI')
plt.xlabel('Age')
plt.ylabel('BMI')
plt.show()
```

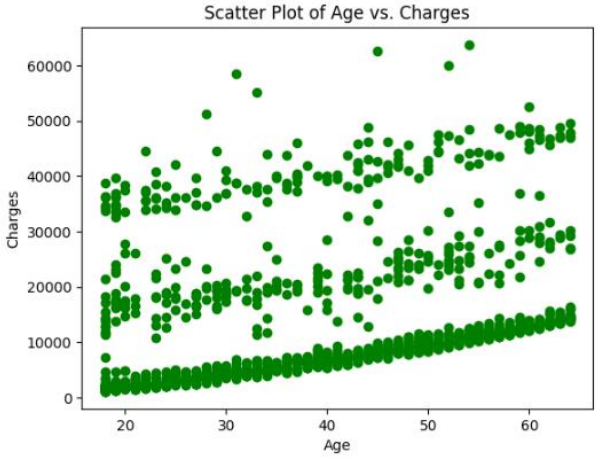
[35]:
```
# Task 5: Perform data visualization using plots of feature vs feature
# Scatter plot for age vs. bmi

plt.scatter(insurance.age, insurance.bmi, c='blue')
plt.title('Scatter Plot of Age vs. BMI')
plt.xlabel('Age')
plt.ylabel('BMI')
plt.show()
```



Scatter plot for age vs. charges
```
plt.scatter(insurance.age, insurance.charges, c='green')
plt.title('Scatter Plot of Age vs. Charges')
plt.xlabel('Age')
plt.ylabel('Charges')
plt.show()
```

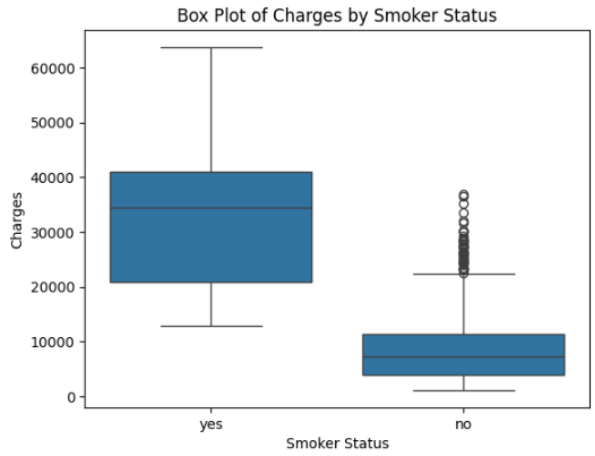
[37]:
```
# Scatter plot for age vs. charges
plt.scatter(insurance.age, insurance.charges, c='green')
plt.title('Scatter Plot of Age vs. Charges')
plt.xlabel('Age')
plt.ylabel('Charges')
plt.show()
```



Box plot of charges by smoker status

```
sns.boxplot(x='smoker', y='charges', data=insurance)
plt.title('Box Plot of Charges by Smoker Status')
plt.xlabel('Smoker Status')
plt.ylabel('Charges')
plt.show()
```

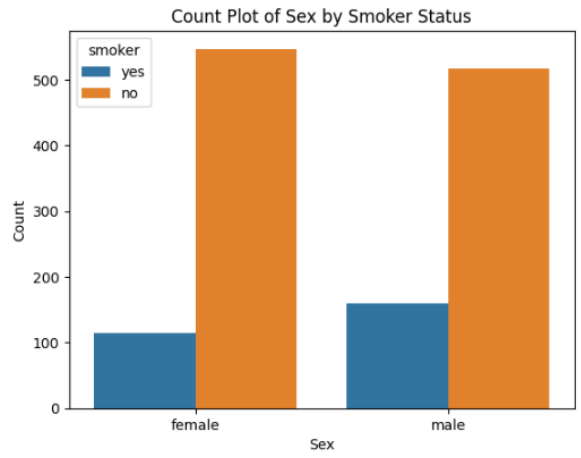
[41]:
```
# Box plot of charges by smoker status
sns.boxplot(x='smoker', y='charges', data=insurance)
plt.title('Box Plot of Charges by Smoker Status')
plt.xlabel('Smoker Status')
plt.ylabel('Charges')
plt.show()
```



Count plot of sex by smoker status

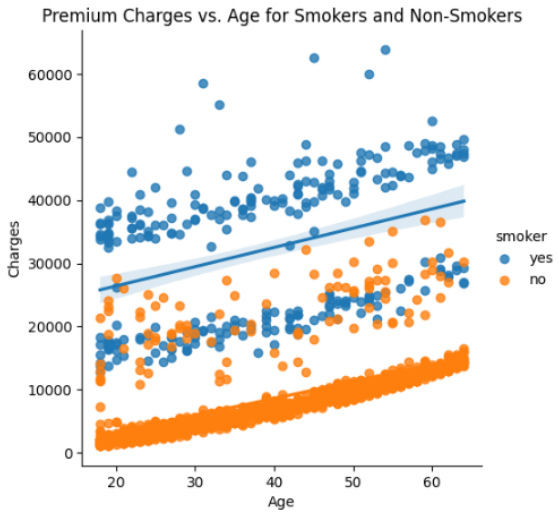
```
sns.countplot(x='sex', hue='smoker', data=insurance)
plt.title('Count Plot of Sex by Smoker Status')
plt.xlabel('Sex')
plt.ylabel('Count')
plt.show()
```

[43]:
```
# Count plot of sex by smoker status
sns.countplot(x='sex', hue='smoker', data=insurance)
plt.title('Count Plot of Sex by Smoker Status')
plt.xlabel('Sex')
plt.ylabel('Count')
plt.show()
```



# Task 6: Check if the number of premium charges for smokers or non-smokers is increasing as they are aging
# Scatter plot with regression lines for smokers and non-smokers
sns.lmplot(x='age', y='charges', hue='smoker', data=insurance)
plt.title('Premium Charges vs. Age for Smokers and Non-Smokers')
plt.xlabel('Age')
plt.ylabel('Charges')
plt.show()

```
•[44]: # Task 6: Check if the number of premium charges for smokers or non-smokers is increasing as they are aging
       # Scatter plot with regression lines for smokers and non-smokers
       sns.lmplot(x='age', y='charges', hue='smoker', data=insurance)
       plt.title('Premium Charges vs. Age for Smokers and Non-Smokers')
       plt.xlabel('Age')
       plt.ylabel('Charges')
       plt.show()
```



# Scatter plot for both smokers and non-smokers
sns.scatterplot(x='age', y='charges', hue='smoker', data=insurance, palette='coolwarm', marker='o')
sns.regplot(x='age', y='charges', data=insurance[insurance['smoker'] == 'yes'], scatter=False, color='red', label='Smokers')
sns.regplot(x='age', y='charges', data=insurance[insurance['smoker'] == 'no'], scatter=False, color='blue', label='Non-Smokers')

plt.title('Premium Charges vs. Age')
plt.xlabel('Age')
plt.ylabel('Charges')
plt.legend()
plt.grid(True)
plt.show()

[47]:
```
# Scatter plot for both smokers and non-smokers
sns.scatterplot(x='age', y='charges', hue='smoker', data=insurance, palette='coolwarm', marker='o')
sns.regplot(x='age', y='charges', data=insurance[insurance['smoker'] == 'yes'], scatter=False, color='red', label='Smokers')
sns.regplot(x='age', y='charges', data=insurance[insurance['smoker'] == 'no'], scatter=False, color='blue', label='Non-Smokers')

plt.title('Premium Charges vs. Age')
plt.xlabel('Age')
plt.ylabel('Charges')
plt.legend()
plt.grid(True)
plt.show()
```