Health insurance premium prediction using Machine Learning

*Background

Health insurance premiums refer to the amount of money individuals or employers pay on a regular basis to secure health insurance coverage for themselves, their families, or their employees. These premiums serve as a primary source of revenue for insurance companies, which use the funds to cover medical expenses and administrative costs. The cost of health insurance premiums has been a longstanding concern due to several factors, including the complexity of the healthcare system, the rising costs of medical care, and the structure of the insurance market.

*Problem Statement

To build a model that will predict the health insurance premium.

*Data Source

The dataset was downloaded from Kaggle which contains the following:

- the age of the person

- gender of the person

- Body Mass Index of the person

- how many children the person is having

- whether the person smokes or not

- the region where the person lives

- and the charges of the insurance premium

In [65]:
```
# Import libraries

import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
```

In [66]:
```
# Load dataset

data = pd.read_csv("Health_insurance.csv")
data.head()
```

Out[66]:

	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

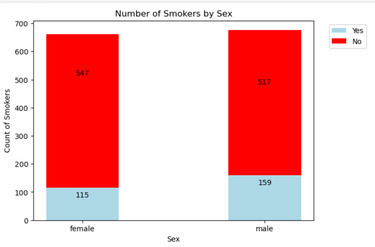
*Exploratory Data Analysis (EDA)

In [67]:
```
# Check for null values (this is important because random forest regression algorithm does not accept NaN values)

data.isnull().sum()
```

Out[67]:
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

After getting the first impressions of this data, I noticed the "smoker" column, which indicates whether the person smokes or not. This is an important feature of the dataset because a person who smokes is more likely to have major health problems compared to a person who does not smoke

In [68]:
```
# Check distribution of smokers and non-smokers

import matplotlib.pyplot as plt

# Filter the data for only "yes" and "no" answers
filtered_data = data[data["smoker"].isin(["yes", "no"])]

# Group the filtered data by 'sex' and 'smoker' columns and count the occurrences
grouped_data = filtered_data.groupby(['sex', 'smoker']).size().reset_index(name='count')

# Pivot the data to have 'sex' as the index, 'smoker' as the columns, and 'count' as values
pivot_data = grouped_data.pivot(index='sex', columns='smoker', values='count')

# Create a figure and axes
fig, ax = plt.subplots(figsize=(7, 5))

# Define the bars for male and female
bars = ax.bar(pivot_data.index, pivot_data["yes"], width=0.4, color='lightblue', label='Yes')
bars += ax.bar(pivot_data.index, pivot_data["no"], bottom=pivot_data["yes"], width=0.4, color='red', label='No')

# Set the title and axis labels
ax.set_title("Number of Smokers by Sex")
ax.set_xlabel("Sex")
ax.set_ylabel("Count of Smokers")

# Add value labels inside the bars
for bar in bars:
    height = bar.get_height()
    ax.annotate('{}'.format(height), xy=(bar.get_x() + bar.get_width() / 2, height),
                xytext=(0, -15), textcoords="offset points", ha='center', va='bottom')

# Move the legend outside and set its position
ax.legend(loc='upper right', bbox_to_anchor=(1.2, 1))

# Show the plot
plt.show()
```



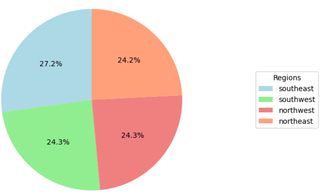
As per the above, 547 females, 517 males don't smoke, and 115 females, 159 males do smoke.Therefore, this is an important feature to use while training a machine learning model.

*Feature Selection

In [69]:
```
# Replace the "sex" and "smoker" column values with 0 and 1 as both these columns contain string values

data["sex"] = data["sex"].map({"female": 0, "male": 1})
data["smoker"] = data["smoker"].map({"no": 0, "yes": 1})
print(data.head())
```

```
   age  sex     bmi  children  smoker     region      charges
0   19    0  27.900         0       1  southwest  16884.92400
1   18    1  33.770         1       0  southeast   1725.55230
2   28    1  33.000         3       0  southeast   4449.46200
3   33    1  22.705         0       0  northwest  21984.47061
4   32    1  28.880         0       0  northwest   3866.85520
```

In [70]:
```
# Check the distribution of the regions where people are living

import matplotlib.pyplot as plt

# Calculate the value counts for each region
pie = data["region"].value_counts()

# Get the region names and population counts
regions = pie.index
population = pie.values

# Create a pie chart
fig, ax = plt.subplots(figsize=(8, 5))
wedges, _, _ = ax.pie(population, colors=['lightblue', 'lightgreen', 'lightcoral', 'lightsalmon'],
                      autopct='%1.1f%%', startangle=90)

# Set the title
ax.set_title("Population Distribution by Region")

# Equal aspect ratio ensures that pie is drawn as a circle
ax.axis('equal')

# Create a legend outside the pie chart
ax.legend(wedges, regions, title="Regions", loc="center left", bbox_to_anchor=(1, 0, 0.5, 1))

# Show the pie chart
plt.show()
```



In [71]:
```
# Check the correlation between the features

print(data.corr())
```

```
               age       sex       bmi  children    smoker   charges
age       1.000000 -0.020856  0.109272  0.042469 -0.025019  0.299008
sex      -0.020856  1.000000  0.046371  0.017163  0.076185  0.057292
bmi       0.109272  0.046371  1.000000  0.012759  0.003750  0.198341
children  0.042469  0.017163  0.012759  1.000000  0.007673  0.067998
smoker   -0.025019  0.076185  0.003750  0.007673  1.000000  0.787251
charges   0.299008  0.057292  0.198341  0.067998  0.787251  1.000000
```

*Model Selection and Training

In [72]:
```
# Create x and y arrays from the "age", "sex", "bmi", and "smoker" columns from the dataset

x = np.array(data[["age", "sex", "bmi",  "smoker"]])
y = np.array(data["charges"])
```

Split data to training and testing

In [73]:
```
# Split the data into 80% training and 20% test sets

from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2, random_state=42)
```

Train the model by using the random forest regression algorithm

In [74]:
```
# Create a random forest regressor model and fit it to the training data

from sklearn.ensemble import RandomForestRegressor
forest = RandomForestRegressor()
forest.fit(xtrain, ytrain)
```

Out[74]:
```
RandomForestRegressor()
```

Results

Predict the health insurance premium

In [75]:
```
# Predict the premium amounts using the trained random forest regressor model

ypred = forest.predict(xtest)

# Create pandas DataFrame to display the predicted values

data = pd.DataFrame(data={"Predicted Premium Amount": ypred})
print(data.head())
```

```
   Predicted Premium Amount
0              10098.016703
1               5704.560068
2              28238.800447
3               9718.891256
4              34795.246376
```

*Summary

Using the train_test_split, we split the inputs and the output into 2 parts containing 80% (to train the model) and 20% (to test the model) data. With the random forest regression algorithm, the resulting model predicted the health insurance premium amount of each individual based on their demographic and lifestyle profiles.

End