Name:-Prashant Shekhar Mishra
Roll No: 56
Registration no:- 18010666

Insurance Cost Prediction

Introduction

n In this project we will learn about:

- The basics of Machine learning by going over a short intro.
- Types of Machine learning.
- Understanding the Linear regression algorithm.
- Understanding the Random Forest Regression.
- Understanding the Principle component Analysis.
- Applying Multiple Linear regression and Random forest Regression to create ML

model to Insurance cost dataset to predict future Insurance costs for the individuals.

Machine learning is a method of data analysis which sends instructions(programmable code) to computers so that they can learn from data. Then, based on the learned data, they provide us the predicted results/patterns. With the help of Machine Learning, we can develop intelligent systems that are capable of taking decisions on an autonomous basis.

Types of Machine Learning-

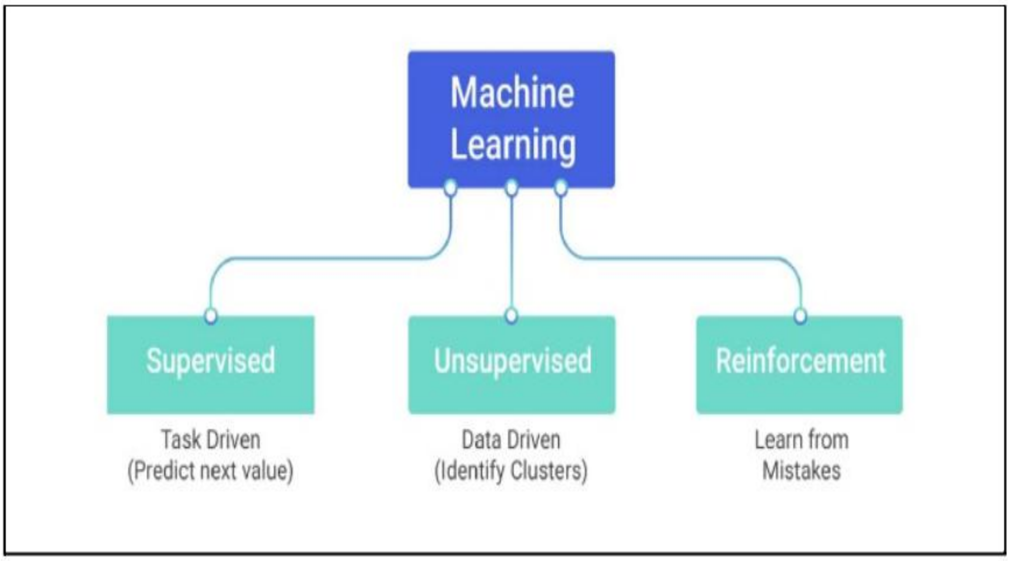
Machine Learning can be classified into 3 types as follows –

1)Supervised Learning algorithms are used when we have labeled data and are trying to predict a label (target) based off of known features(input variables). This is commonly used in applications where historical data predicts likely future events.

For example, it can attempt to predict the price of a product/car/house based on different features for products for which we have historical price data.

2)Unsupervised Learning algorithms are used when we have unlabeled data and are trying to group together similar data points based off of features. This is mainly used to explore the data and find some structure within.
For example, it can identify the image(cat or dog) based on different inputs which groups together similar segments and then attempts to recognize the image correctly. This is unsupervised learning, where a machine is not taught but learns from the data (in this case data about a dog or cat)

3)Reinforcement Learning occurs when a computer system receives data in a specific environment and then learns how to maximize its outcomes. That means this model keeps continues to learn until best possible behavior is met. Reinforcement learning is frequently used for robotics, gaming, and navigation.



Supervised learning problems can be further grouped into:

- **Regression problems and**
- **Classification problems**

In classification, learning algorithms takes the input data and map the output to a discrete output like True or False In regression, learning algorithms maps the input data to continuous output like weight, cost, etc.

In this project I will apply regression techniques of supervised learning to predict the insurance costs.

Methods and Data

To create the claim cost model predictor, we obtained the data set through the project provided. The data set includes seven attributes see below ; the data set is separated into two-part the first part called training data, and the second called test data; training data makes up about 80 percent of the total data used, and the rest for test data The training data set is applied to build a model as a predictor of insurance cost year and the test set will use to evaluate the regression model. The following data below shows the Description of the Dataset.

Columns

- age: age of primary beneficiary

- bmi: Body mass index, providing an understanding of body, weights that are relatively high or low relative to height, objective index of body weight (kg / m ^ 2) using the ratio of height to weight, ideally 18.5 to 24.9
- children: Number of children covered by health insurance / Number of dependents
- Smoker: Smoking
- region: the beneficiary's residential area in the US, northeast, southeast, southwest,
 northwest.

- charges: Individual medical costs billed by health insurance(target – y)

Data Cleaning

This dataset needed some cleanings and modification. Besides some feature representation should be done.

· Some of the features in the dataset are self-reported and they are not the same across subjects. For example, city could be either “Bangalore” or “Bangalor” or “Banglor”. These discrepancies could be fixed manually.

· In case of categorical data, we used one-hot-encoding.
· In case of missing values, values were imputed by mean and variance.

Data Analysis and Visualization
In this project, the following things done are:
- Statistical measure of the dataset
- Age distribution Plot

- Gender column count plot
- Bmi distribution plot
- Children column count plot
- Smoker column count plot
- Region column count plot
- Charges distribution plot

In this project we are going to build these following ML models:

- Multiple Linear Regression
- Random Forest Regression
- Multiple Linear Regression using Principal Component Analysis
- Random Forest Regression using Principal Component Analysis

Input Dataset used are:

1. Age
2. Sex
3. BMI
4. Children
5. Smoker
6. Region
7. Charges

Multiple Linear Regression

Multiple linear regression is simply the extension of simple linear regression, that predicts the value of a dependent variable (sometimes it is called as the outcome, target or criterion variable) on the basis of two or more independent variables (or sometimes, the predictor, explanatory or regressor variables).
The equation of multiple linear regression is expressed as;

yi=ß0+ ß1 xi1+ ß2 xi2 +........+ ßp xip+ Ø

Where for
i=n observations;
yi= dependent variable,

xi= explanatory variables, here we have “p” predictor variables and “p+1” as total regression parameters.
ß0= y-intercept which is a constant term,
ßp= Slope coefficient for each explanatory variable,   and

Ø= residuals (model’s error term), having a normal distribution with mean 0 and constant variance,
In multiple linear regression, the word linear signifies that the model is linear in parameters, ß0, ß1, ß2 and so on.

Work Flow of Multiple linear Regression :

Step 1: First we have to do the Data analysis.
Step 2: We have to pre-process the data in suitable manner for the model preparation.
Step 3: In this step we have to create the linear regression model.
Step 4: We have to build the predictive system which predicts the insurance cost after reading the input data.

In [1]:

```
!pip install seaborn
!pip install sklearn
!pip install matplotlib
```

```
Requirement already satisfied: seaborn in c:\users\prashant mishra\appdata\local\pro
grams\python\python39\lib\site-packages (0.11.2)
Requirement already satisfied: pandas>=0.23 in c:\users\prashant mishra\appdata\loca
l\programs\python\python39\lib\site-packages (from seaborn) (1.3.2)
Requirement already satisfied: scipy>=1.0 in c:\users\prashant mishra\appdata\local
\programs\python\python39\lib\site-packages (from seaborn) (1.7.1)
Requirement already satisfied: numpy>=1.15 in c:\users\prashant mishra\appdata\local
\programs\python\python39\lib\site-packages (from seaborn) (1.21.2)
Requirement already satisfied: matplotlib>=2.2 in c:\users\prashant mishra\appdata\l
ocal\programs\python\python39\lib\site-packages (from seaborn) (3.4.3)
Requirement already satisfied: pillow>=6.2.0 in c:\users\prashant mishra\appdata\loc
al\programs\python\python39\lib\site-packages (from matplotlib>=2.2->seaborn) (8.3.
2)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\prashant mishra\appdata
\local\programs\python\python39\lib\site-packages (from matplotlib>=2.2->seaborn)
(1.3.2)
Requirement already satisfied: cycler>=0.10 in c:\users\prashant mishra\appdata\loca
l\programs\python\python39\lib\site-packages (from matplotlib>=2.2->seaborn) (0.11.
0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\prashant mishra\appdata
\local\programs\python\python39\lib\site-packages (from matplotlib>=2.2->seaborn)
(2.4.7)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\prashant mishra\appd
ata\local\programs\python\python39\lib\site-packages (from matplotlib>=2.2->seaborn)
(2.8.2)
Requirement already satisfied: pytz>=2017.3 in c:\users\prashant mishra\appdata\loca
l\programs\python\python39\lib\site-packages (from pandas>=0.23->seaborn) (2021.1)
Requirement already satisfied: six>=1.5 in c:\users\prashant mishra\appdata\local\pr
ograms\python\python39\lib\site-packages (from python-dateutil>=2.7->matplotlib>=2.2
->seaborn) (1.16.0)
Requirement already satisfied: sklearn in c:\users\prashant mishra\appdata\local\pro
grams\python\python39\lib\site-packages (0.0)
Requirement already satisfied: scikit-learn in c:\users\prashant mishra\appdata\loca
l\programs\python\python39\lib\site-packages (from sklearn) (1.0.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\prashant mishra\appd
ata\local\programs\python\python39\lib\site-packages (from scikit-learn->sklearn)
(3.0.0)
Requirement already satisfied: numpy>=1.14.6 in c:\users\prashant mishra\appdata\loc
al\programs\python\python39\lib\site-packages (from scikit-learn->sklearn) (1.21.2)
Requirement already satisfied: joblib>=0.11 in c:\users\prashant mishra\appdata\loca
l\programs\python\python39\lib\site-packages (from scikit-learn->sklearn) (1.0.1)
Requirement already satisfied: scipy>=1.1.0 in c:\users\prashant mishra\appdata\loca
l\programs\python\python39\lib\site-packages (from scikit-learn->sklearn) (1.7.1)
Requirement already satisfied: matplotlib in c:\users\prashant mishra\appdata\local
\programs\python\python39\lib\site-packages (3.4.3)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\prashant mishra\appdata
\local\programs\python\python39\lib\site-packages (from matplotlib) (2.4.7)
Requirement already satisfied: cycler>=0.10 in c:\users\prashant mishra\appdata\loca
l\programs\python\python39\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\prashant mishra\appd
ata\local\programs\python\python39\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: numpy>=1.16 in c:\users\prashant mishra\appdata\local
\programs\python\python39\lib\site-packages (from matplotlib) (1.21.2)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\prashant mishra\appdata
\local\programs\python\python39\lib\site-packages (from matplotlib) (1.3.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\prashant mishra\appdata\loc
al\programs\python\python39\lib\site-packages (from matplotlib) (8.3.2)
Requirement already satisfied: six>=1.5 in c:\users\prashant mishra\appdata\local\pr
```

```
ograms\python\python39\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.
16.0)
```

In [2]:

```
# Importing Header Files
import numpy as np
import pandas as pd
from numpy import math
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
```

In [3]:

```
#Loading Data
dataset = pd.read_csv('insurance.csv')
```

In [4]:

```
# Printing keys
print(dataset.keys())
```

```
Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges'], dtype='objec
t')
```

In [5]:

```
#First 10 data
dataset.head(10)
```

Out[5]:

	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
5	31	female	25.740	0	no	southeast	3756.62160
6	46	female	33.440	1	no	southeast	8240.58960
7	37	female	27.740	3	no	northwest	7281.50560
8	37	male	29.830	2	no	northeast	6406.41070
9	60	female	25.840	0	no	northwest	28923.13692

In [6]:

```
#Last 10 data
dataset.tail(10)
```

Out[6]:

	age	sex	bmi	children	smoker	region	charges
1328	23	female	24.225	2	no	northeast	22395.74424
1329	52	male	38.600	2	no	southwest	10325.20600
1330	57	female	25.740	2	no	southeast	12629.16560
1331	23	female	33.400	0	no	southwest	10795.93733

	age	sex	bmi	children	smoker	region	charges
1332	52	female	44.700	3	no	southwest	11411.68500
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

In [7]:

```
# Replacing string values to numbers
dataset['sex'] = dataset['sex'].apply({'male':0,'female':1}.get)
dataset['smoker'] = dataset['smoker'].apply({'yes':1, 'no':0}.get)
dataset['region'] = dataset['region'].apply({'southwest':1, 'southeast':2, 'northwes
```

In [8]:

```
#First 10 data
dataset.head(10)
```

Out[8]:

	age	sex	bmi	children	smoker	region	charges
0	19	1	27.900	0	1	1	16884.92400
1	18	0	33.770	1	0	2	1725.55230
2	28	0	33.000	3	0	2	4449.46200
3	33	0	22.705	0	0	3	21984.47061
4	32	0	28.880	0	0	3	3866.85520
5	31	1	25.740	0	0	2	3756.62160
6	46	1	33.440	1	0	2	8240.58960
7	37	1	27.740	3	0	3	7281.50560
8	37	0	29.830	2	0	4	6406.41070
9	60	1	25.840	0	0	3	28923.13692

In [9]:

```
#Last 10 data
dataset.tail(10)
```

Out[9]:

	age	sex	bmi	children	smoker	region	charges
1328	23	1	24.225	2	0	4	22395.74424
1329	52	0	38.600	2	0	1	10325.20600
1330	57	1	25.740	2	0	2	12629.16560
1331	23	1	33.400	0	0	1	10795.93733
1332	52	1	44.700	3	0	1	11411.68500
1333	50	0	30.970	3	0	3	10600.54830
1334	18	1	31.920	0	0	4	2205.98080
1335	18	1	36.850	0	0	2	1629.83350

	age	sex	bmi	children	smoker	region	charges
1336	21	1	25.800	0	0	1	2007.94500
1337	61	1	29.070	0	1	3	29141.36030

In [10]:

```
# Checking for NULL values
dataset.isnull().sum()
```

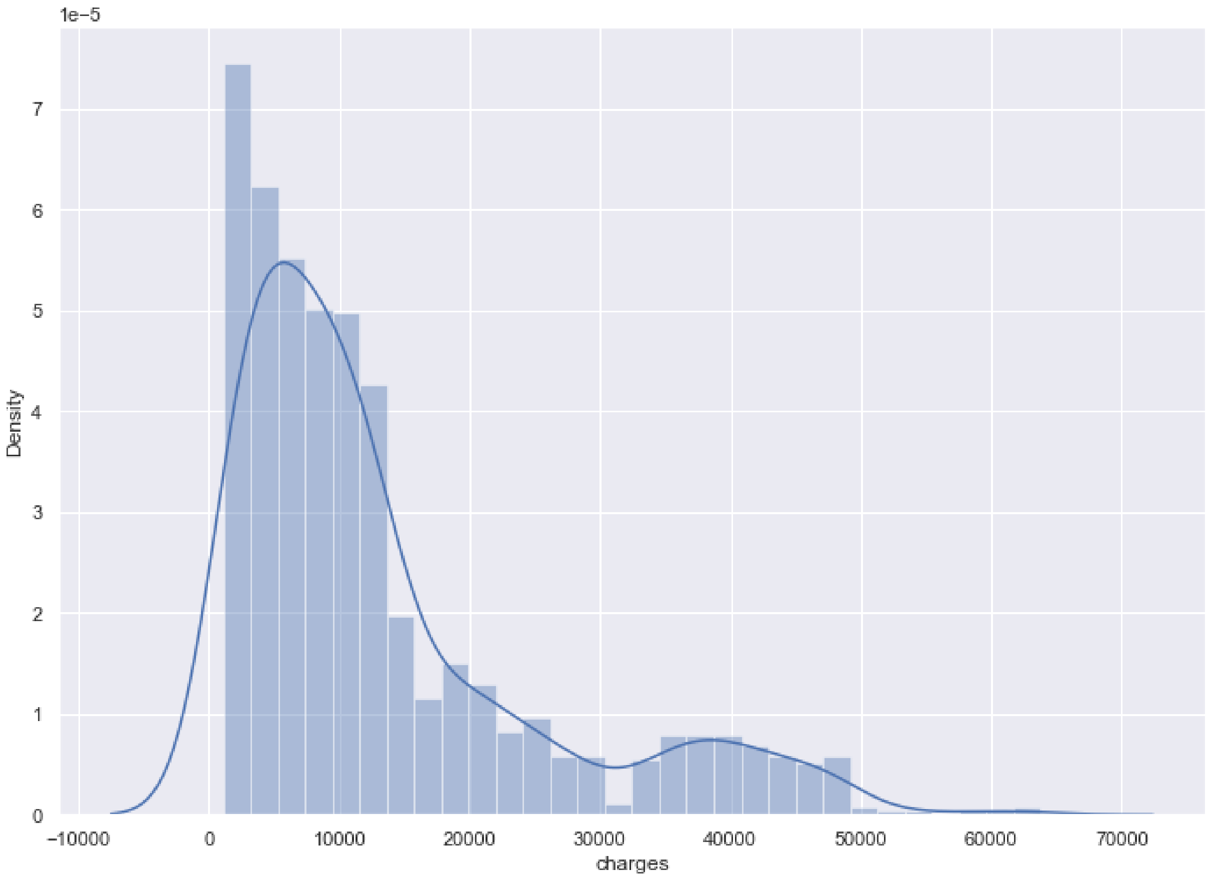
Out[10]:

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

In [11]:

```
# EDA
sns.set(rc={'figure.figsize':(11.7,8.27)})
sns.distplot(dataset['charges'], bins=30)
plt.show()
```

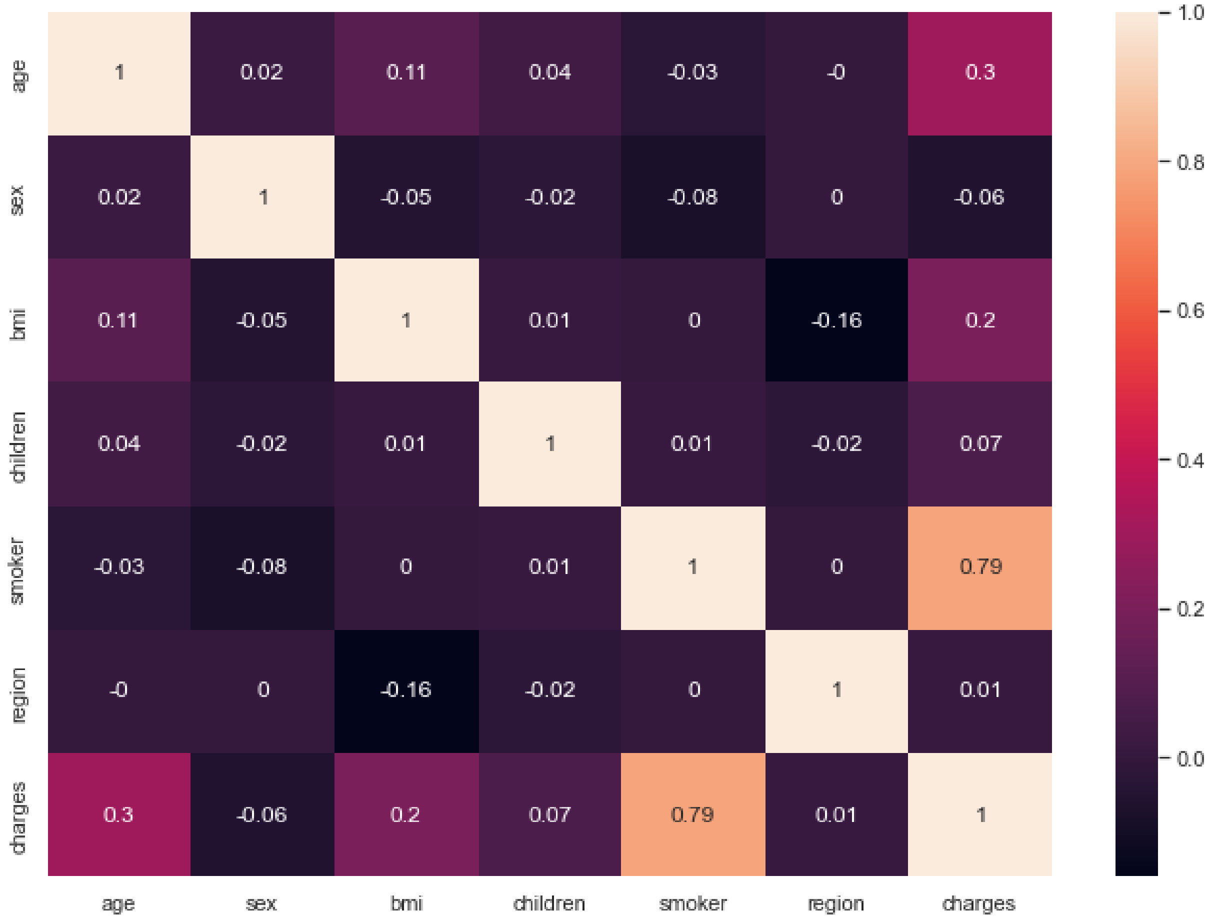
```
c:\users\prashant mishra\appdata\local\programs\python\python39\lib\site-packages\se
aborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and
will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `histplot` (an axes-level func
tion for histograms).
  warnings.warn(msg, FutureWarning)
```



In [12]:

```
# Correlation Matrix
correlation_matrix = dataset.corr().round(2)
# annot = True to print the values inside the square
sns.heatmap(data=correlation_matrix, annot=True)
```

Out[12]:
```
<AxesSubplot:>
```



There is really good relation between smoker and charges. So we will go with that.

In [13]:
```python
# features
X = dataset[['age','sex','bmi','children','smoker','region']]

# predicted variable
Y = dataset['charges']
```

In [14]:
```python
#Values in our X
X.head()
```

Out[14]:

	age	sex	bmi	children	smoker	region
0	19	1	27.900	0	1	1
1	18	0	33.770	1	0	2
2	28	0	33.000	3	0	2
3	33	0	22.705	0	0	3
4	32	0	28.880	0	0	3

In [15]:
```python
#Values in our y
Y.head()
```

Out[15]:
```
0     16884.92400
1      1725.55230
2      4449.46200
```

```
3    21984.47061
4     3866.85520
Name: charges, dtype: float64
```

In [16]:

```python
# Splitting the dataset into training and testing
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.33)
print(X_train.shape)
print(X_test.shape)
print(Y_train.shape)
print(Y_test.shape)
```

```
(896, 6)
(442, 6)
(896,)
(442,)
```

In [17]:

```python
# Building the Linear Regression Model
lin_model = LinearRegression()
lin_model.fit(X_train, Y_train)
```

Out[17]:

```
LinearRegression()
```

In [18]:

```python
# Model Evaluation

y_train_predict = lin_model.predict(X_train)
mse = mean_squared_error(Y_train, y_train_predict)
rmse = (np.sqrt(mse))
r2 = r2_score(Y_train, y_train_predict)

print("The model performance for training set:\n")
print('MSE is {}'.format(mse))
print('RMSE is {}'.format(rmse))
print('R2 score is {}'.format(r2))
print("\n")
```

```
The model performance for training set:

MSE is 36670720.942923464
RMSE is 6055.635469785435
R2 score is 0.7369587975707715
```

In [19]:

```python
# Predict charges for new customer : Name- prashant
data = {'age' : 40,
        'sex' : 1,
        'bmi' : 45.50,
        'children' : 4,
        'smoker' : 1,
        'region' : 3}

index = [1]
prashant_df = pd.DataFrame(data,index)
print(prashant_df.head())

prediction_prashant = lin_model.predict(prashant_df)
print("\n\nMedical Insurance cost for prashant is : ",prediction_prashant)
```

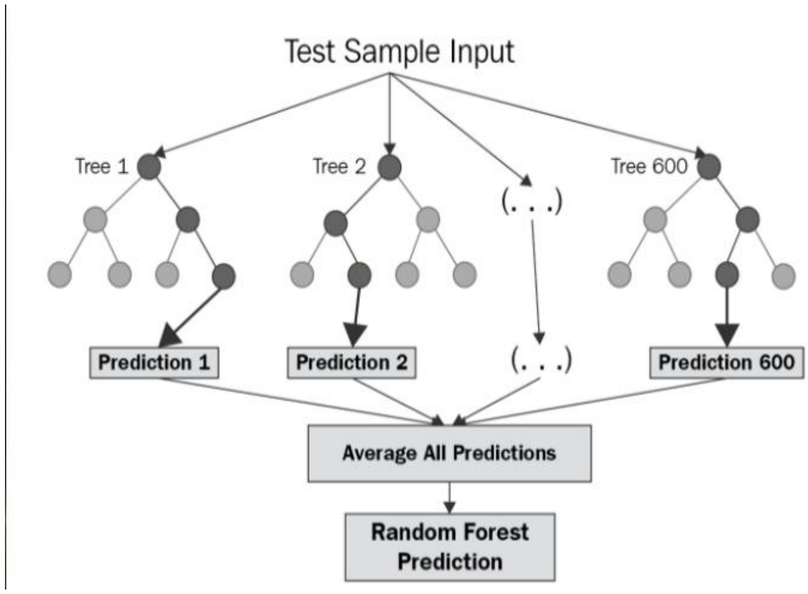
```
   age  sex   bmi  children  smoker  region
1   40    1  45.5         4       1       3
```

```
Medical Insurance cost for prashant is :  [38399.25421277]
```

In [ ]:

2. Random Forest Regression

Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.



The diagram above shows the structure of a Random Forest. You can notice that the trees run in parallel with no interaction amongst them. A Random Forest operates by constructing several decision trees during training time and outputting the mean of the classes as the prediction of all the trees. To get a better understanding of the Random Forest algorithm, let's walk through the steps:

Pick at random k data points from the training set.
Build a decision tree associated to these k data points.
Choose the number N of trees you want to build and repeat steps 1 and 2.
For a new data point, make each one of your N-tree trees predict the value of y for the data point in question and assign the new data point to the average across all of the predicted y values.

A Random Forest Regression model is powerful and accurate. It usually performs great on many problems, including features with non-linear relationships. Disadvantages, however, include the following: there is no interpretability, overfitting may easily occur, we must choose the number of trees to include in the model.

Working or random forest regression:

Step 1: Identify your dependent (y) and independent variables (X)
Step 2: Split the dataset into the Training set and Test set
Step 3: Training the Random Forest Regression model on the whole dataset
Step 4: Predicting the Test set results

In [1]:

```
!pip install seaborn
!pip install sklearn
!pip install matplotlib
```

```
Requirement already satisfied: seaborn in c:\users\prashant mishra\appdata\local\pro
grams\python\python39\lib\site-packages (0.11.2)
Requirement already satisfied: scipy>=1.0 in c:\users\prashant mishra\appdata\local
\programs\python\python39\lib\site-packages (from seaborn) (1.7.1)
Requirement already satisfied: numpy>=1.15 in c:\users\prashant mishra\appdata\local
\programs\python\python39\lib\site-packages (from seaborn) (1.21.2)
Requirement already satisfied: matplotlib>=2.2 in c:\users\prashant mishra\appdata\l
ocal\programs\python\python39\lib\site-packages (from seaborn) (3.4.3)
Requirement already satisfied: pandas>=0.23 in c:\users\prashant mishra\appdata\loca
l\programs\python\python39\lib\site-packages (from seaborn) (1.3.2)
Requirement already satisfied: cycler>=0.10 in c:\users\prashant mishra\appdata\loca
l\programs\python\python39\lib\site-packages (from matplotlib>=2.2->seaborn) (0.11.
0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\prashant mishra\appdata
\local\programs\python\python39\lib\site-packages (from matplotlib>=2.2->seaborn)
(1.3.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\prashant mishra\appd
ata\local\programs\python\python39\lib\site-packages (from matplotlib>=2.2->seaborn)
(2.8.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\prashant mishra\appdata\loc
al\programs\python\python39\lib\site-packages (from matplotlib>=2.2->seaborn) (8.3.
2)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\prashant mishra\appdata
\local\programs\python\python39\lib\site-packages (from matplotlib>=2.2->seaborn)
(2.4.7)
Requirement already satisfied: pytz>=2017.3 in c:\users\prashant mishra\appdata\loca
l\programs\python\python39\lib\site-packages (from pandas>=0.23->seaborn) (2021.1)
Requirement already satisfied: six>=1.5 in c:\users\prashant mishra\appdata\local\pr
ograms\python\python39\lib\site-packages (from python-dateutil>=2.7->matplotlib>=2.2
->seaborn) (1.16.0)
Requirement already satisfied: sklearn in c:\users\prashant mishra\appdata\local\pro
grams\python\python39\lib\site-packages (0.0)
Requirement already satisfied: scikit-learn in c:\users\prashant mishra\appdata\loca
l\programs\python\python39\lib\site-packages (from sklearn) (1.0.1)
Requirement already satisfied: scipy>=1.1.0 in c:\users\prashant mishra\appdata\loca
l\programs\python\python39\lib\site-packages (from scikit-learn->sklearn) (1.7.1)
Requirement already satisfied: numpy>=1.14.6 in c:\users\prashant mishra\appdata\loc
al\programs\python\python39\lib\site-packages (from scikit-learn->sklearn) (1.21.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\prashant mishra\appd
ata\local\programs\python\python39\lib\site-packages (from scikit-learn->sklearn)
(3.0.0)
Requirement already satisfied: joblib>=0.11 in c:\users\prashant mishra\appdata\loca
l\programs\python\python39\lib\site-packages (from scikit-learn->sklearn) (1.0.1)
Requirement already satisfied: matplotlib in c:\users\prashant mishra\appdata\local
\programs\python\python39\lib\site-packages (3.4.3)
Requirement already satisfied: pillow>=6.2.0 in c:\users\prashant mishra\appdata\loc
al\programs\python\python39\lib\site-packages (from matplotlib) (8.3.2)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\prashant mishra\appdata
\local\programs\python\python39\lib\site-packages (from matplotlib) (2.4.7)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\prashant mishra\appdata
\local\programs\python\python39\lib\site-packages (from matplotlib) (1.3.2)
Requirement already satisfied: cycler>=0.10 in c:\users\prashant mishra\appdata\loca
l\programs\python\python39\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: numpy>=1.16 in c:\users\prashant mishra\appdata\local
\programs\python\python39\lib\site-packages (from matplotlib) (1.21.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\prashant mishra\appd
ata\local\programs\python\python39\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\prashant mishra\appdata\local\pr
```

```
ograms\python\python39\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.
16.0)
```

In [2]:

```
# Importing the Libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
%matplotlib inline
```

In [3]:

```
# Importing and printing the dataset
dataset = pd.read_csv('insurance.csv')
```

In [4]:

```
# Printing keys
print(dataset.keys())
```

```
Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges'], dtype='objec
t')
```

In [5]:

```
#First 10 data
dataset.head(10)
```

Out[5]:

	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
5	31	female	25.740	0	no	southeast	3756.62160
6	46	female	33.440	1	no	southeast	8240.58960
7	37	female	27.740	3	no	northwest	7281.50560
8	37	male	29.830	2	no	northeast	6406.41070
9	60	female	25.840	0	no	northwest	28923.13692

In [6]:

```
#Last 10 data
dataset.tail(10)
```

Out[6]:

	age	sex	bmi	children	smoker	region	charges
1328	23	female	24.225	2	no	northeast	22395.74424
1329	52	male	38.600	2	no	southwest	10325.20600
1330	57	female	25.740	2	no	southeast	12629.16560

	age	sex	bmi	children	smoker	region	charges
1331	23	female	33.400	0	no	southwest	10795.93733
1332	52	female	44.700	3	no	southwest	11411.68500
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

In [7]:

```
# Replacing string values to numbers
dataset['sex'] = dataset['sex'].apply({'male':0,'female':1}.get)
dataset['smoker'] = dataset['smoker'].apply({'yes':1, 'no':0}.get)
dataset['region'] = dataset['region'].apply({'southwest':1, 'southeast':2, 'northwes
```

In [8]:

```
#First 10 data
dataset.head(10)
```

Out[8]:

	age	sex	bmi	children	smoker	region	charges
0	19	1	27.900	0	1	1	16884.92400
1	18	0	33.770	1	0	2	1725.55230
2	28	0	33.000	3	0	2	4449.46200
3	33	0	22.705	0	0	3	21984.47061
4	32	0	28.880	0	0	3	3866.85520
5	31	1	25.740	0	0	2	3756.62160
6	46	1	33.440	1	0	2	8240.58960
7	37	1	27.740	3	0	3	7281.50560
8	37	0	29.830	2	0	4	6406.41070
9	60	1	25.840	0	0	3	28923.13692

In [9]:

```
#Last 10 data
dataset.tail(10)
```

Out[9]:

	age	sex	bmi	children	smoker	region	charges
1328	23	1	24.225	2	0	4	22395.74424
1329	52	0	38.600	2	0	1	10325.20600
1330	57	1	25.740	2	0	2	12629.16560
1331	23	1	33.400	0	0	1	10795.93733
1332	52	1	44.700	3	0	1	11411.68500
1333	50	0	30.970	3	0	3	10600.54830
1334	18	1	31.920	0	0	4	2205.98080

	age	sex	bmi	children	smoker	region	charges
1335	18	1	36.850	0	0	2	1629.83350
1336	21	1	25.800	0	0	1	2007.94500
1337	61	1	29.070	0	1	3	29141.36030

In [10]:

```
# Checking for NULL values
dataset.isnull().sum()
```

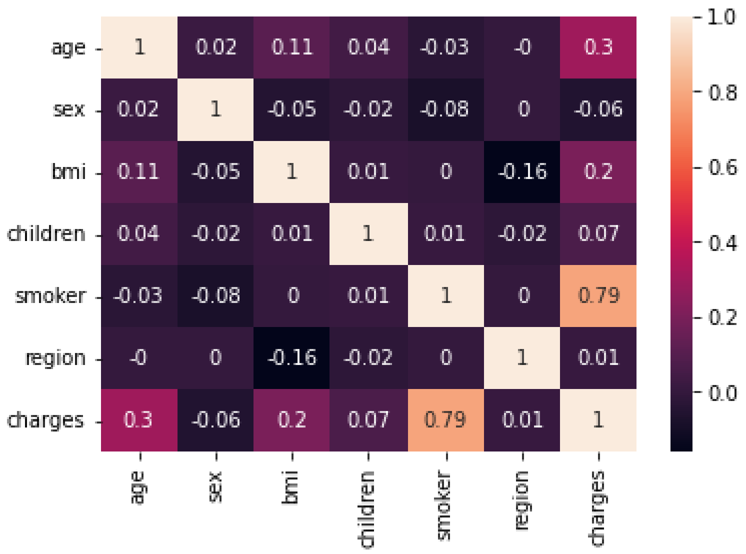
Out[10]:

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

In [11]:

```
# Correlation Matrix
correlation_matrix = dataset.corr().round(2)
# annot = True to print the values inside the square
sns.heatmap(data=correlation_matrix, annot=True)
```

Out[11]: <AxesSubplot:>



In [12]:

```
# features
X = dataset[['age','bmi','smoker']]

# predicted variable
Y = dataset['charges']

# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.33) # 70% trai
```

In [13]:

```
# Create a Gaussian Classifier
clf=RandomForestClassifier(n_estimators=100)

# Train the model using the training sets y_pred=clf.predict(X_test)
clf.fit(X_train.astype('int'),y_train.astype('int'))
```

```python
y_pred=clf.predict(X_test.astype('int'))
```

In [14]:

```python
# Model Evaluation

y_pred = clf.predict(X_train)
mse = mean_squared_error(y_train, y_pred)
rmse = (np.sqrt(mse))
r2 = r2_score(y_train, y_pred)

print("The model performance for training set:\n")
print('MSE is {}'.format(mse))
print('RMSE is {}'.format(rmse))
print('R2 score is {}'.format(r2))
print("\n")
```

```
The model performance for training set:

MSE is 21084291.457086332
RMSE is 4591.7634365335425
R2 score is 0.8556200339934718
```

In [15]:

```python
# Predict charges for new customer : Name- Prashant
data = {'age' : 40,
        'bmi' : 45.50,
        'smoker' : 1}

index = [1]
prashant_df = pd.DataFrame(data,index)
print(prashant_df.head())

prediction_prashant = clf.predict(prashant_df)
print("\n\nMedical Insurance cost for prashant is : ",prediction_prashant)
```

```
   age   bmi  smoker
1   40  45.5       1


Medical Insurance cost for prashant is :  [46113]
```