1. Introduction:

Health insurance is a kind of insurance product that mainly focuses on or guarantees the health costs or meet the cost of healthcare of the insurance members if they have an accident or fall ill. In today’s scenario, especially in this pandemic, it’s crucial that each avail insurance scheme, i.e. is affordable and meets their taste and requirements. For this insurance company to understand the factors that impact a user’s health insurance premium would be essential to make the accurate charge, premium always is a user’s priority consideration to make appropriate decisions. Thus from the customer/user perspective, it’s essential to choose and decide what schemes suits them the most in terms of money, requirement, current health issues, etc.

a. Overview:

The health care sector is the primary area that all countries in the world focus on. Nowadays there are a good amount of people who draws a relation between insurance and good health. There is no doubt that health insurance can improve health measures when people require better treatment options. But is difficult for common people to decide on what policy/scheme of health insurance as there several factors that determine the cost/expense of the insurance. For example, just eliminating smoking and lowering your BMI by a few points might mean shaving thousands of dollars from your premium charges. By leveraging artificial intelligence (AI) and machine learning, we can help customers understand just how much age, smoking, and other factors increase their premium by predicting how much they will have to pay within seconds.

b. Purpose:

This project aims to explore the use of machine learning algorithms to predict the prices of annual health insurance premiums given the specifications of the contract and the company's demographics. This helps users identify the factors that influence the health insurance cost. According to the output, which demonstrated that the majority of factors contributing to health insurance premiums cost are BMI, smoke status, age, and children, these four factors have a significant correlation impact on health insurance premiums. Given health insurance information about a company, we accurately predict how much it will cost per year? Using Software like IBM Auto AI and Machine Algorithms like Multiple linear regression, Random forest, Decision tree Regressor, we can try to predict the premium costs of an insurance policy.

2. Literature Survey:

a. Existing problem:

The most crucial problem associated with what exists today is the expense of health insurance premiums. Health insurance or medical claim is opted for by most individuals with various insurance companies to pay for hospitalization or medical expenses. The amount of premium that needs to be paid towards this type of insurance is decided upon by calculations various parameter that includes: Pre-existing medical conditions:, age, gender, children, Injurious substances like the habit of smoking and consuming alcohol. Hence rates of premium for their insurance plans increase/decreases as per these parameters.

So it’s difficult for people to calculate or get an idea of the expense or cost of insurance; how does the cost vary from individual to individual, and they keep searching for plans or policies when they are in need.

b. Proposed solution:

In this application, we study the effects of age, smoking, BMI, gender, and region to determine how much of a difference these factors can make on your insurance premium.

This project aims at building a web App that automatically estimates premium costs by taking the input values. By using our application, customers see the radical difference their lifestyle choices make on their insurance charges. The following are the methodologies adopted for data analysis, prediction and display of results by integrating with the system:

- IBM Auto AI
- Machine Learning algorithm using python
- FLASK

Create a model from a dataset that includes the age, gender, BMI, number of children, smoking preferences, region, and expenses to predict the health insurance premium cost that an individual pays. Using IBM AutoAI and Machine Learning algorithms, we automate all of the tasks involved in building predictive models for different requirements. FLASK is a framework that helps build web apps that could act as the interface to the user for input and output in the front and integration of values in the backend.

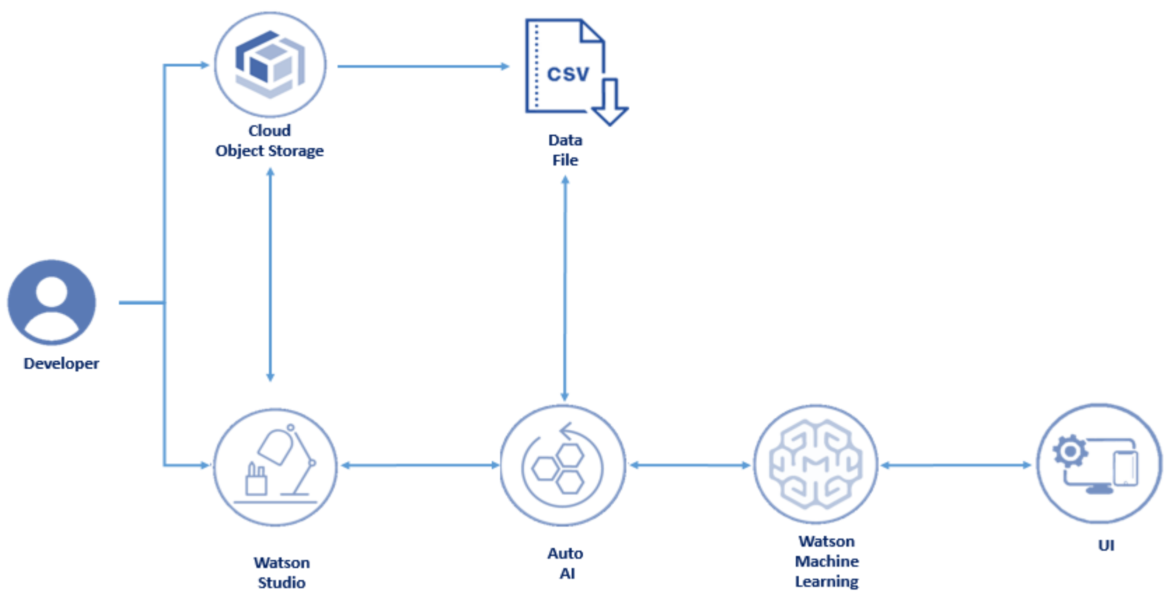
3. Theoretical Analysis:

1. Block Diagram:

We have carried out or used two approaches for Data analysis for Health Insurance Prediction which are :

- **Prediction using IBM Auto AI**
- **Prediction using Machine Learning Models**

Prediction using IBM Auto AI:

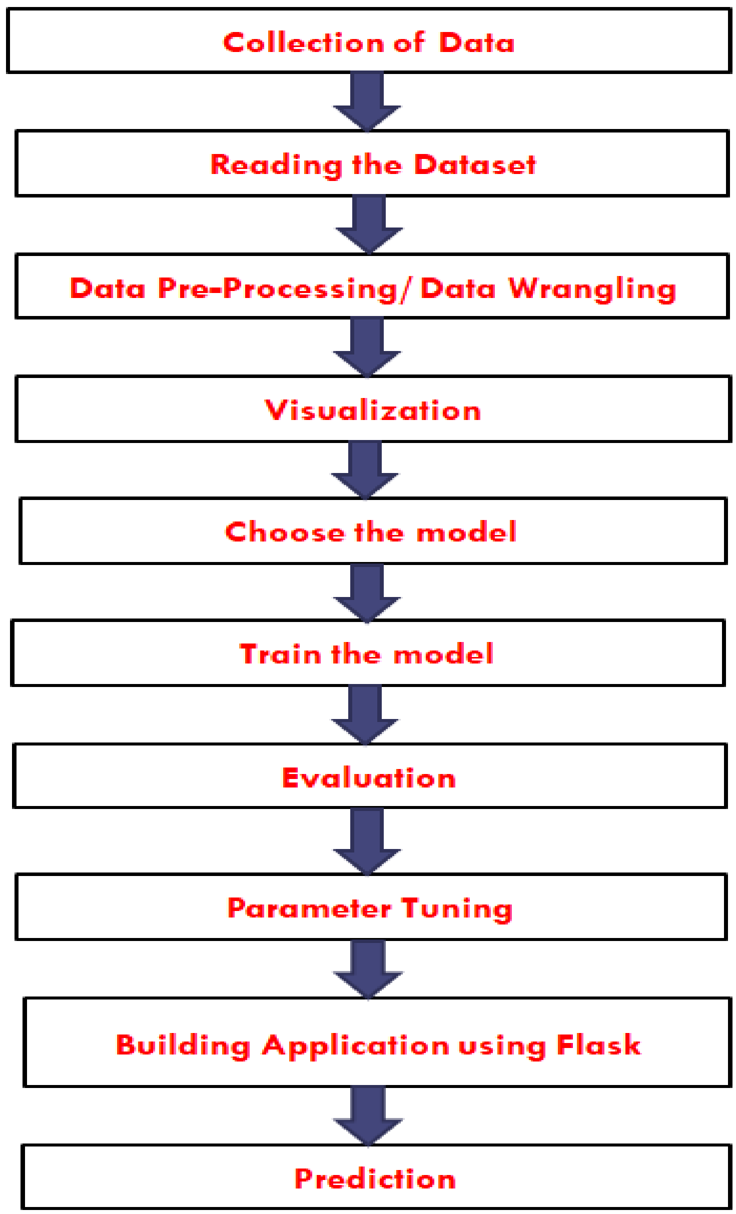


The user creates an IBM Watson Studio Service, IBM Cloud Object Storage Service on IBM Cloud.

- The user uploads the insurance premium data file into Watson Studio.
- The user creates an AutoAI Experiment to predict an insurance premium on Watson Studio.

- Auto AI uses Machine Learning services to create several models, and the user deploys the best performing model.
- We use the FLASK web application to connect to the deployed model and predict insurance.

Prediction using Machine Learning Models:



b. Hardware / Software designing:

Hardware:

- Lenovo Ideapad-500/8Gb RAM/64 bit/Windows 10.

Software:

- IBM Cloud, Spider Python(Anaconda), Notebook(Anaconda), FLASK.

4. Experimental Investigations:

Dataset:

SmartInterns provided the data source link, and the source of data for this project was from Kaggle. The dataset is comprised of 1338 records with 7 attributes. Attributes are as follow age, gender, bmi, children, smoker, region and expenses. The data was in a structured format and was stores in a CSV file. In a dataset, not every attribute has an impact on the prediction. Whereas some attributes even decline the accuracy, so it becomes necessary to remove these attributes from the features of the code. Removing such attributes not only help in improving accuracy but also the overall performance and speed.

Machine Learning

Machine learning can be defined as the process of teaching a computer system that allows it to make accurate predictions after the data is fed.

Regression:

Since the data in the dataset is continuos-continuous, the best suited Machine Learning model is regression. So cleaning of dataset becomes vital for using the data under various regression algorithms. Regression analysis allows us to quantify the relationship between outcome and associated variables.

Many techniques for performing statistical predictions have been developed, but, in this project, three models – Multiple Linear Regression (MLR), Decision tree regression and Gradient Boosting Regression were tested and compared.

Multiple Linear Regression:

Multiple linear regression can be defined as extended simple linear regression. It comes under usage when we want to predict a single output depending upon multiple inputs or we can say that the predicted value of a variable is based upon the value of two or more different variables. The predicted variable or the variable we want to predict is called the dependent variable (or sometimes, the outcome, target or criterion variable) and the variables being used in predict of the value of the dependent variable are called the independent variables (or sometimes, the predictor, explanatory or regressor variables).

Random Forest Regressor:

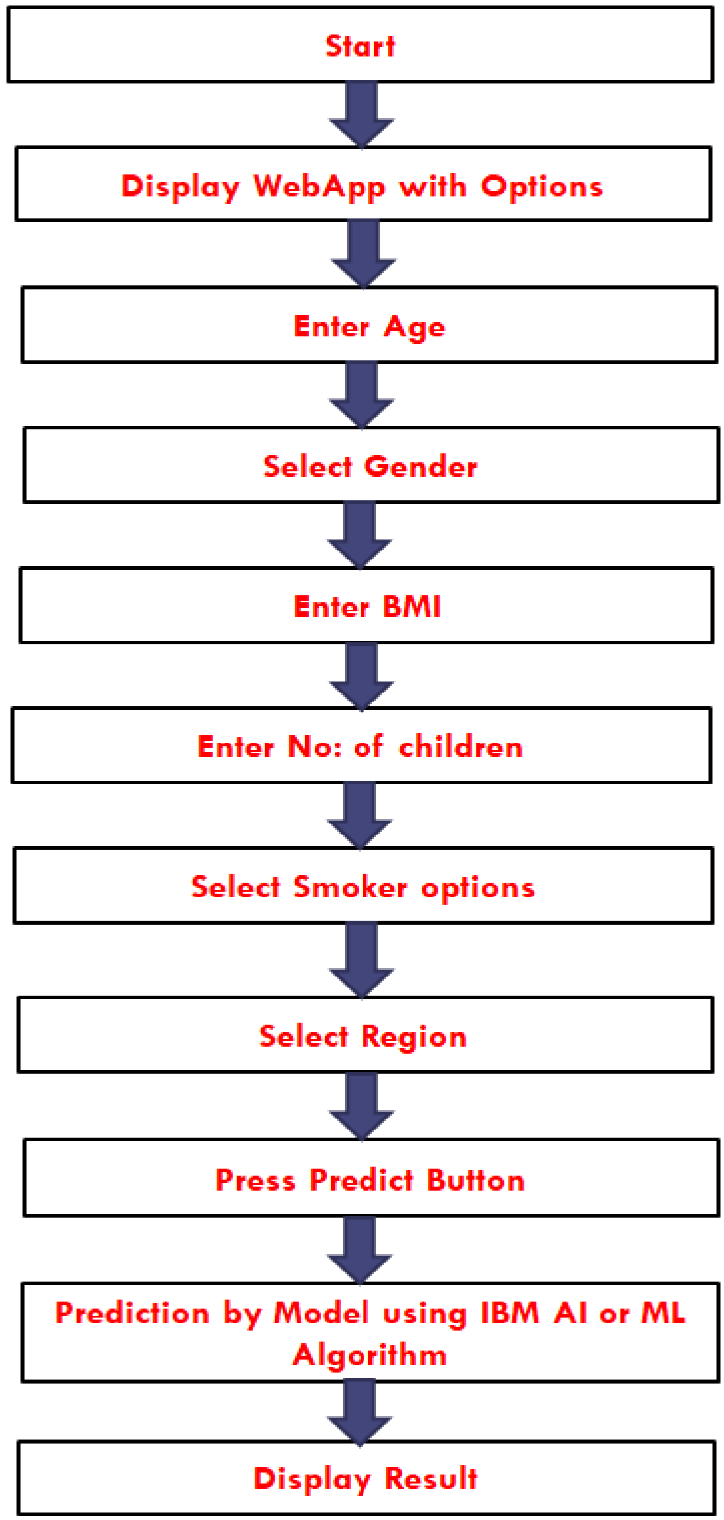
Random forest is a supervised learning algorithm which is used for both classification as well as regression. Random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting.

Decision Tree regressor:

Regression or classification models in decision tree regression build in the form of a tree structure. The dataset is divided or segmented into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. A decision tree with decision nodes and leaf nodes is obtained as a final result. These decision nodes have two or more branches, each representing values for the attribute tested.

The leaf node represents the decision on the numerical target. The topmost decision node corresponds to the best predictor in the tree called the root node. Decision trees can handle numerical data along with categorical data.

5. Flowchart:



Start
Display WebApp with Options
Enter Age
Select Gender
Enter BMI
Enter No: of children
Select Smoker options
Select Region
Press Predict Button
Prediction by Model using IBM AI or ML Algorithm
Display Result

6. Result:

- **Prediction using IBM Auto AI:**

  With Linear Regression model using **age, gender, BMI, children, smoker, region as Independent variables** and **expense as Dependent variable**, **Root Mean Square Error** (**RMSE**) value in "Pipeline 7" is the top performer as the RMSE is 4903.651. So the model or Pipeline 7 gives you less RMSE, which is the best model by the prediction using IBM Auto AI.

- **Prediction using Machine Learning Models:**

  Among the models with various ie Multiple linear Regression, Random forest Regressor, Decision Tree Regressor used with **age, gender, bmi, children, smoker, region as Independent variables** and **expense as Dependent variable**, it has been found that **Decision Tree Regressor** model, which is built upon a decision tree, is the best performing model with an accuracy of **88%**.

7. Advantages & Disadvantages:

Advantages:

- It is not practical for a medical insurance company to do analysis and interpretation on an enormous amount of data without Machine learning.
- Machine learning models can perform these calculations in minimal effort, time and investment.
- This can result in the profitability of such insurance companies as they could make decisions with proper background finding and also save time and money by engaging

human resources that would take a lot of time.

Disadvantages:

- Insufficient data cant result in miss interpretation and wrong prediction. For e.g. in the same dataset, pre-existing body condition, family medical history, current diseases, marital status, location, past insurances, etc. are some other missing attributes that can contribute or make a change in accuracy and prediction.
- The most important in the building of models would be data preprocessing. If this initial stage is not carried out or if the data is not preprocessed properly by a developer, this can lead to a poor prediction model and the business that uses it.

8. Applications:

This concept could be mainly applicable and beneficial to the health insurance industry and the public as it would be used for:

- Predicting Risk Scores For Healthcare Insurers.
- Applications that help doctors or the patient directly suggesting preventative healthy behaviours and habits to patients.
- Applications that used by dieticians, instructors or individuals healthcare expenditures that unhealthy habits could cause.

9. Conclusion:

In ou data analysis we have used IBM Ato AI and three regression models namely Multiple linear Regression, Random forest Regressor and Decision tree Regressor for **Prediction using Machine Learning Models** to evaluate health insurance data.

Prediction using IBM Auto AI:

It has been found that the Linear Regression model, **Root Mean Square Error** (**RMSE**) has obtained “Pipeline 7” as the top performer as the RMSE 4575.693. RMSE tells you how concentrated the data is around the line of best fit. So the model which gives you the less RMSE that will be taken into consideration.

Prediction using Machine Learning Models:

It has been found that Decision Tree Regressor model, which is built upon a decision tree, is the best performing model with an accuracy of 88%. Various factors were used, and their effect on predicted amounts was examined. It was observed that a persons age and smoking status affects the prediction but it is to be noted that other parameters were also significant as the difference of values as per the correlation matrix was comparatively less. So we have built the model using all the parameters for our prediction of expense.

10. Future Scope:

Premium amount prediction focuses on persons own health rather than other company’s insurance terms and conditions. The models can be applied to the data collected in the coming years to predict the premium. This can help people and insurance companies to work in tandem for better and more health-centre insurance amounts.

11. Bibliography:

- Yang, Y., Qian, W., & Zou, H. (2018). Insurance premium prediction via gradient tree-boosted Tweedie compound Poisson models. *Journal of Business & Economic Statistics*, *36*(3), 456-470.

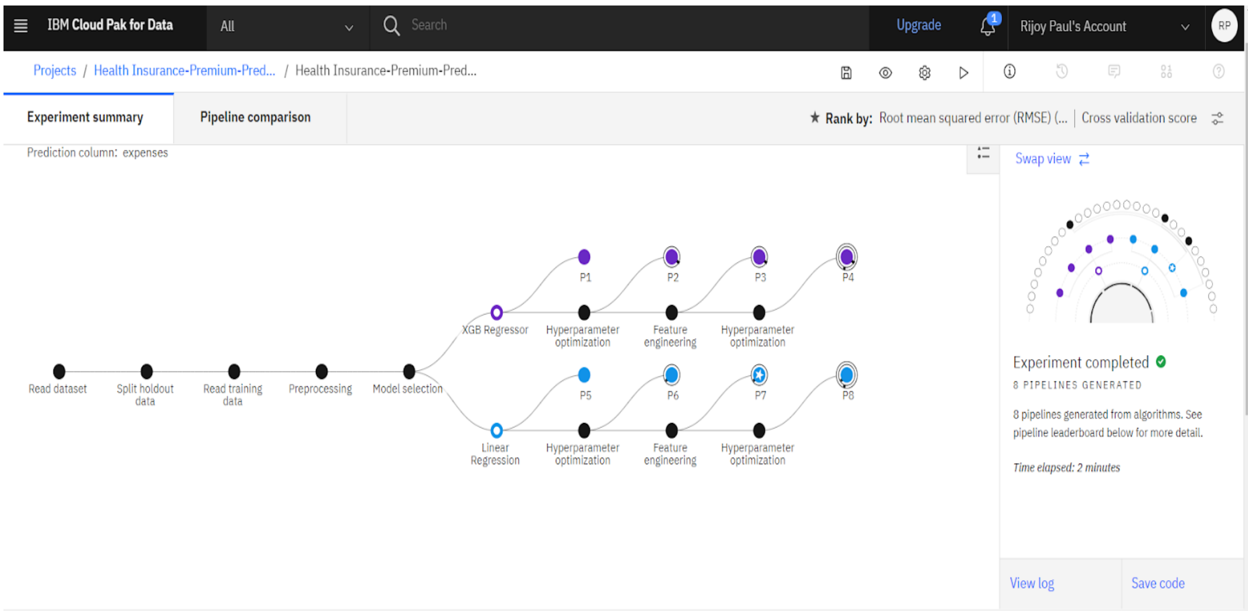
- Lui, E. Employer Health Insurance Premium Prediction.

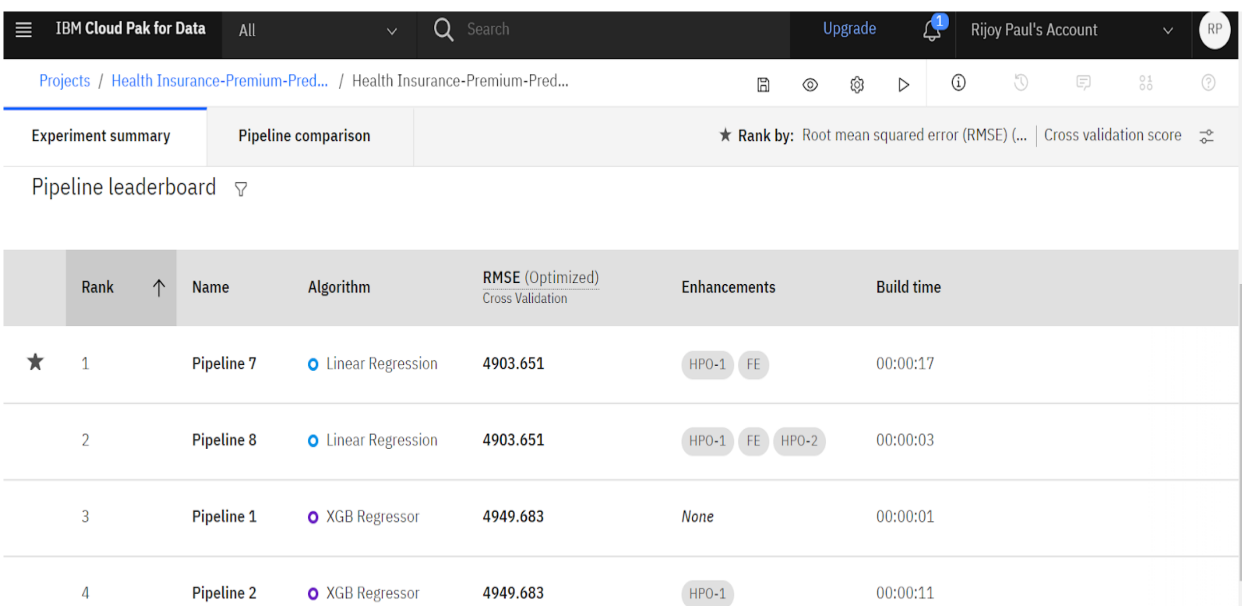
- Sun, J. J. (2020). *Identification and Prediction of Factors Impact America Health Insurance Premium* (Doctoral dissertation, Dublin, National College of Ireland).

12. Appendix:

a. Source code

- **Prediction using IBM Auto AI:**





Rank
1

Pipeline 7

Holdout RMSE *(Optimized)*
4575.693

Algorithm
Linear Regression

Enhancements
HPO-1 FE

Model viewer

Model information

Feature transformations

Feature importance

Evaluation

Model evaluation

Model information

Experiment parameters

Prediction column	expenses
Algorithm	Linear Regression
Number of features	14
Created on	6/27/2021, 1:18:50 AM

Rank
1

Pipeline 7

Holdout RMSE *(Optimized)*
4575.693

Algorithm
Linear Regression

Enhancements
HPO-1 FE

Model viewer

Model information

Feature transformations

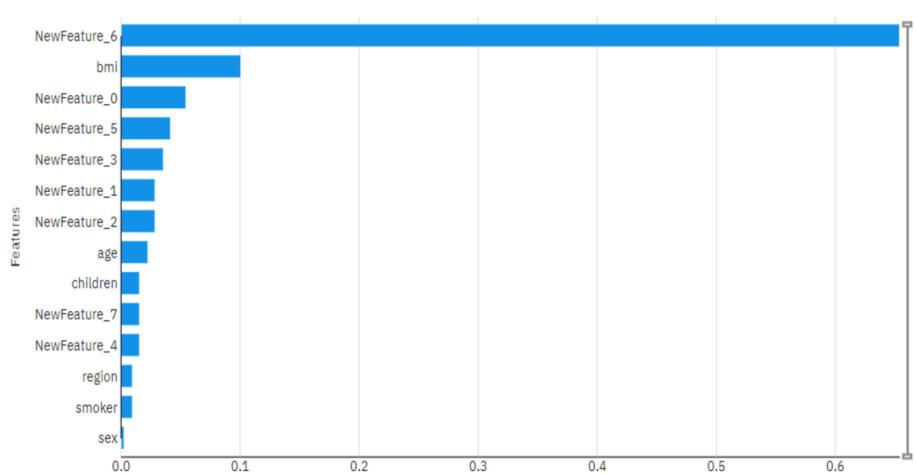
Feature importance

Evaluation

Model evaluation

Feature importance

Features by importance



Rank
1

Pipeline 7

Holdout RMSE *(Optimized)*
4575.693

Algorithm
Linear Regression

Enhancements
HPO-1 FE

Model viewer

Model information

Feature transformations

Feature importance

Evaluation

Model evaluation

Model evaluation

Model evaluation measure

Measures	Holdout score	Cross validation score
Root mean squared error	4575.693	4903.651
R squared	0.871	0.833
Explained variance	0.871	0.834
Mean squared error	20936964.000	24100102.667
Mean absolute error	2957.034	2977.977
Median absolute error	1711.525	1691.882

- ## Prediction using Machine Learning Models:

SmartInternz

Project: Health Insurance-Premium-Prediction

Name : Rijoy Paul

Institution: Christ University, Bangalore

Problem Statement:

One major issue in Health insurances with the common people or new users is to estimate the cost of premiums to decide which would be the best for them. This project aims at building a web App that automatically estimates premium cost by taking the input values from user.

In [6]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [7]:
```
data =pd.read_csv('Insurance.csv') #Reads the file as csv
data.head() #filters the first top datas in dataset
```

Out[7]:

	age	sex	bmi	children	smoker	region	expenses
0	19	female	27.9	0	yes	southwest	16884.92
1	18	male	33.8	1	no	southeast	1725.55
2	28	male	33.0	3	no	southeast	4449.46
3	33	male	22.7	0	no	northwest	21984.47
4	32	male	28.9	0	no	northwest	3866.86

In [4]:
```
data.shape #size of the dataset
```

Out[4]: (1338, 7)

In [5]:
```
data.describe()
```

Out[5]:

	age	bmi	children	expenses
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.665471	1.094918	13270.422414
std	14.049960	6.098382	1.205493	12110.011240
min	18.000000	16.000000	0.000000	1121.870000
25%	27.000000	26.300000	0.000000	4740.287500
50%	39.000000	30.400000	1.000000	9382.030000
75%	51.000000	34.700000	2.000000	16639.915000
max	64.000000	53.100000	5.000000	63770.430000

In [6]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1338 non-null   int64
 1   sex       1338 non-null   object
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64
 4   smoker    1338 non-null   object
 5   region    1338 non-null   object
 6   expenses  1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
```

Null Value identifcation

In [135]:
```
data.isnull().sum() #displays the null value counts for each parameters
```

Out[135]:
```
Year        0
Age         0
Agesq       0
Nbh         0
Cbd         0
Intst       0
Lintst      0
Price       0
Rooms       0
Area        0
Land        0
Baths       0
Dist        0
Ldist       0
Wind        0
Lprice      0
Y81         0
Larea       0
Lland       0
Y81Ldist    0
Lintstsq    0
Nearinc     0
Y81Nrinc    0
Rprice      0
Lrprice     0
dtype: int64
```

So the above null value count displays that there is no null values present in the dataset for each parameters.

In [ ]:

```
In [8]: data_columns=data.columns #columns in dataset
        data_columns
```

```
Out[8]: Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'expenses'], dtype='object')
```

```
In [9]: from collections import Counter as c
        print(c(data.sex))
        print(c(data.smoker))
        print(c(data.region))
```

```
Counter({'male': 676, 'female': 662})
Counter({'no': 1064, 'yes': 274})
Counter({'southeast': 364, 'southwest': 325, 'northwest': 325, 'northeast': 324})
```

Label Encoding

```
In [10]: from sklearn.preprocessing import LabelEncoder
         le= LabelEncoder()
         data.sex=le.fit_transform(data.sex)
         data.smoker=le.fit_transform(data.smoker)
         data.region=le.fit_transform(data.region)
         print(c(data.sex))
         print(c(data.smoker))
         print(c(data.region))
```

```
Counter({1: 676, 0: 662})
Counter({0: 1064, 1: 274})
Counter({2: 364, 3: 325, 1: 325, 0: 324})
```

Correlation

```
In [13]: data.corr()
```

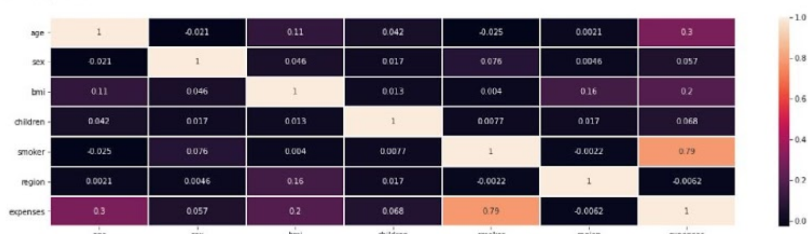
Out[13]:

	age	sex	bmi	children	smoker	region	expenses
age	1.000000	-0.020856	0.109341	0.042469	-0.025019	0.002127	0.299008
sex	-0.020856	1.000000	0.046380	0.017163	0.076185	0.004588	0.057292
bmi	0.109341	0.046380	1.000000	0.012645	0.003968	0.157439	0.198576
children	0.042469	0.017163	0.012645	1.000000	0.007673	0.016569	0.067998
smoker	-0.025019	0.076185	0.003968	0.007673	1.000000	-0.002181	0.787251
region	0.002127	0.004588	0.157439	0.016569	-0.002181	1.000000	-0.006208
expenses	0.299008	0.057292	0.198576	0.067998	0.787251	-0.006208	1.000000

Correlation visulaization using Heatmap

```
In [14]: plt.figure(figsize = (20,5))
         sns.heatmap(data.corr(),annot=True,linewidths=1)
```

Out[14]: <AxesSubplot:>



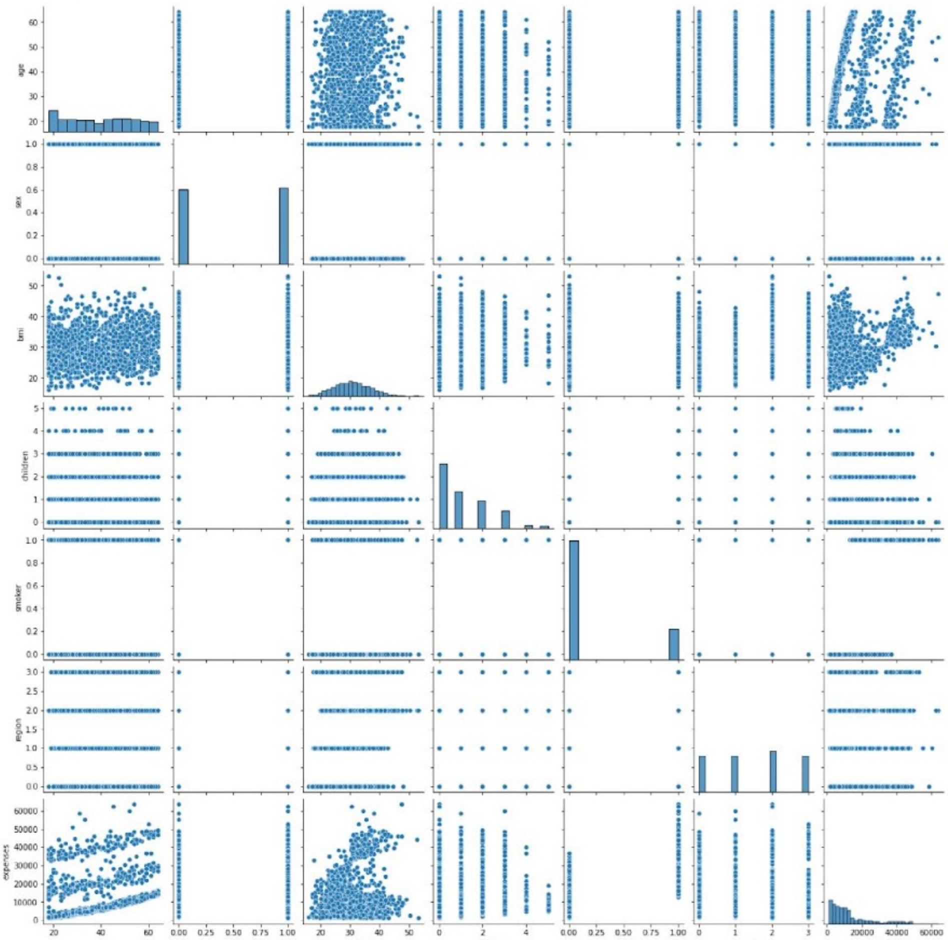
From the above Correlation Matrix and Heatmap is evident that variables say: age, gender, bmi, children, smoker, region have a positive correlation and age and smoker have a high correlation. Although age and smoker have a high correlation other attributes also contribute as their diffrence is comparitively low. Hence these are the parameters to be used as input variables for building the model

Visualizations

```
In [6]: data_columns=data.columns #columns in dataset
        data_columns
```

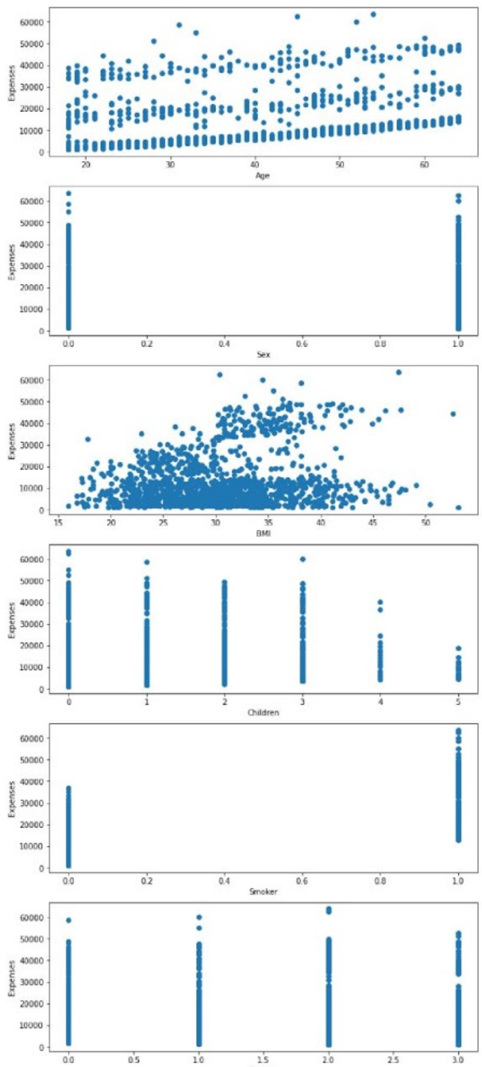
```
Out[6]: Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'expenses'], dtype='object')
```

```
In [16]: sns.pairplot(data,diag_kind="hist")
Out[16]: <seaborn.axisgrid.PairGrid at 0x17e71454f40>
```



In [17]:
```
fig, ax = plt.subplots(6, figsize=(10, 25))
ax[0].scatter(x = data['age'], y = data['expenses'])
ax[0].set_xlabel("Age")
ax[0].set_ylabel("Expenses")

ax[1].scatter(x = data['sex'], y = data['expenses'])
ax[1].set_xlabel("Sex")
ax[1].set_ylabel("Expenses")

ax[2].scatter(x = data['bmi'], y = data['expenses'])
ax[2].set_xlabel("BMI")
ax[2].set_ylabel("Expenses")

ax[3].scatter(x = data['children'], y = data['expenses'])
ax[3].set_xlabel("Children")
ax[3].set_ylabel("Expenses")

ax[4].scatter(x = data['smoker'], y = data['expenses'])
ax[4].set_xlabel("Smoker")
ax[4].set_ylabel("Expenses")

ax[5].scatter(x = data['region'], y = data['expenses'])
ax[5].set_xlabel("Region")
ax[5].set_ylabel("Expenses")
```

Out[17]:
```
Text(0, 0.5, 'Expenses')
```

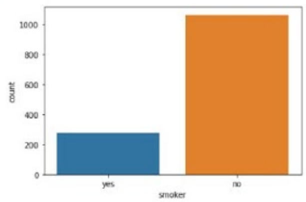


In [ ]:

In [5]:
```python
sns.countplot(data['smoker'])
```

```
  be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[5]: `<AxesSubplot:xlabel='smoker', ylabel='count'>`



This indicates that 'smoker' column in the dataset is significant as the smoking preference in the data also contribute to the changes in insurance premium charges. Also this shows that the insurance companies are more keen on non smokers.

In [ ]:

Extracting required Independent & Dependent variables

In [11]:
```python
predmod_columns=['age', 'sex', 'bmi', 'children', 'smoker', 'region']
```

Independent & Dependent variables

age, gender, bmi, children, smoker, region are taken as the Input variables(Independent) and expense as Target variable(Dependent) for the model.

In [19]:
```python
predmod_columns
```

Out[19]: `['age', 'sex', 'bmi', 'children', 'smoker', 'region']`

In [12]:
```python
x=data.iloc[:,0:6]
x.head()
```

Out[12]:

	age	sex	bmi	children	smoker	region
0	19	0	27.9	0	1	3
1	18	1	33.8	1	0	2
2	28	1	33.0	3	0	2
3	33	1	22.7	0	0	1
4	32	1	28.9	0	0	1

In [13]:
```python
y=data.iloc[:,6:]
y.head()
```

Out[13]:

	expenses
0	16884.92
1	1725.55
2	4449.46
3	21984.47
4	3866.86

Split the dataset to Train and Test data

In [14]:
```python
from sklearn.model_selection import train_test_split
tts=train_test_split
x_train,x_test,y_train,y_test=tts(x,y,test_size=0.2,random_state=0)

print(x_train.shape) #training input
print(y_train.shape) #training output
print(x_test.shape)#testing input
print(y_test.shape)#testing output
```

```
(1070, 6)
(1070, 1)
(268, 6)
(268, 1)
```

In [ ]:

1. Multiple Linear Regression Model (Model Building)

In [42]:
```python
#model Building

from sklearn.linear_model import LinearRegression
mlr=LinearRegression()
mlr.fit(x_train,y_train)
```

Out[42]: `LinearRegression()`

Input Variable: age, gender, bmi, children, smoker, region

Output Variable: expense

In [26]:
```
y_test[:5]
```
Out[26]:

	expenses
578	9724.53
610	8547.69
569	45702.02
1034	12950.07
198	9644.25

In [29]:
```
mlr.predict(x_test[:5])
```
Out[29]:
```
array([[11016.49787742],
       [ 9796.8325871 ],
       [38004.81817394],
       [16128.17665663],
       [ 6945.5990141 ]])
```

Model Accuracy

In [43]:
```
from sklearn.metrics import r2_score

r2_score(y_test,mlr.predict(x_test))
```
Out[43]:
```
0.7999053396503136
```

In [ ]:

In [ ]:

2. Random Forest Regression Model (Model Building)

In [61]:
```
#model Building

from sklearn.ensemble import RandomForestRegressor
rfc=RandomForestRegressor()
rfc.fit(x_train,y_train)
```
```
<ipython-input-61-771bd87aaf4c>:5: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  rfc.fit(x_train,y_train)
```
Out[61]:
```
RandomForestRegressor()
```

Input Variable: age, gender, bmi, children, smoker, region

Output Variable: expense

In [26]:
```
y_test[:5]
```
Out[26]:

	expenses
578	9724.53
610	8547.69
569	45702.02
1034	12950.07
198	9644.25

In [65]:
```
y_pred_rfc=rfc.predict(x_test)
y_pred_rfc[:5]
```
Out[65]:
```
array([11929.4207,  9731.299 , 44080.3832, 13250.9454,  9963.9071])
```

Model Accuracy

In [66]:
```
from sklearn.metrics import r2_score
r2_score(y_pred_rfc,y_test)
```
Out[66]:
```
0.8677605594986721
```

In [ ]:

3. Decision Tree Regression Model (Model Building)

In [15]:
```
from sklearn.tree import DecisionTreeRegressor
dtr=DecisionTreeRegressor()
dtr.fit(x_train,y_train)
```
Out[15]:
```
DecisionTreeRegressor()
```

Input Variable: age, gender, bmi, children, smoker, region

Output Variable: expense

In [16]:
```
y_test[:5]
```
Out[16]:

	expenses
578	9724.53
610	8547.69
569	45702.02
1034	12950.07
198	9644.25

In [17]:
```
y_pred_dtr=dtr.predict(x_test)
y_pred_dtr[:5]
```
Out[17]:
```
array([ 9487.64, 21232.18, 42983.46, 13143.86,  9566.99])
```

Model Accuracy

In [18]:
```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred_dtr)
```

Out[18]: 0.6436260257112161

Hyperparameter Tuning

In [19]:
```
params={
    'criterion':['mse','mae'],
    'splitter':['best','random'],
    'max_depth':[1,2,3],
    'min_samples_split':[1,2,3]
}
```

In [20]:
```
from sklearn.model_selection import GridSearchCV
gridcv=GridSearchCV(dtr,params,cv=5,n_jobs=-1)
```

In [21]:
```
gridcv.fit(x_train,y_train)
```

Out[21]:
```
GridSearchCV(cv=5, estimator=DecisionTreeRegressor(), n_jobs=-1,
             param_grid={'criterion': ['mse', 'mae'], 'max_depth': [1, 2, 3],
                         'min_samples_split': [1, 2, 3],
                         'splitter': ['best', 'random']})
```

In [52]:
```
gridcv.best_params_
```

Out[52]:
```
{'criterion': 'mse',
 'max_depth': 3,
 'min_samples_split': 2,
 'splitter': 'best'}
```

In [24]:
```
from sklearn.tree import DecisionTreeRegressor
dtr_cv=DecisionTreeRegressor(criterion= 'mse',
 max_depth= 3,
 min_samples_split= 3,
 splitter='best')
dtr_cv.fit(x_train,y_train)
```

Out[24]: DecisionTreeRegressor(max_depth=3, min_samples_split=3)

In [25]:
```
y_test[0:5]
```

Out[25]:

	expenses
578	9724.53
610	8547.69
569	45702.02
1034	12950.07
198	9644.25

In [26]:
```
y_pred_cv=dtr_cv.predict(x_test)
y_pred_cv[0:5]
```

Out[26]:
```
array([13786.34940639, 10411.87707006, 45656.34255319, 13786.34940639,
       10411.87707006])
```

Model Accuracy

In [27]:
```
r2_score(y_test,y_pred_cv)
```

Out[27]: 0.8820170441826178

In [34]:
```
r2_score(y_train,dtr_cv.predict(x_train))
```

Out[34]: 0.8466402728661795

Since r2_score of testing data is 88% and r2_score of train data is 85%, it as a good model

In [73]:
```
import pickle
pickle.dump(dtr_cv,open('Insurance.pkl','wb'))
```

In [ ]:

Solutions/ Conclusions:

It has been found that Decision Tree Regressor model, which is built upon a decision tree, is the best performing model with an accuracy of 88%. Various factors were used, and their effect on predicted amounts was examined. It was observed that a persons age and smoking status affects the prediction but it is to be noted that other parameters were also significant as the difference of values as per the correlation matrix was comparatively less. So we have built the model using all the parameters for our prediction of expense. Also to note that age, smoking preference has high impact on the increase of expense

In [ ]:

b. UI output Screenshot: