Final Project Report
Introduction to Data Analytics

Project Title:
Predicting Insurance Premiums

Prepared by:
Darshankumar Patel(N01496781)

ITE 5201 – Summer 2022
Humber College

1. Problem Statement

"Prediction of Insurance Premium"

2. Dataset Description

- We have customer information who are paying insurance premium. We've got information regarding customer such as age, sex, bmi (Body Mass Index), children, smoker, region, and expenses. Expenses is the column where shows information about how much premium customer is paying.
- Dependent Variable are age, sex, bmi (Body Mass Index), children, smoker and expenses which I have used to train my model. We have categorical values for sex and smoker columns, so I convert it to binary 1s and 0s.

```
insurance=pd.read_csv('C:\\Users\\darsh\\Downloads\\archive\\insurance.csv',header=0);
insurance
```

Out[60]:

	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
...	...	...	...	...	...	...	...
1333	50	male	31.0	3	no	northwest	10600.55
1334	18	female	31.9	0	no	northeast	2205.98
1335	18	female	36.9	0	no	southeast	1629.83
1336	21	female	25.8	0	no	southwest	2007.95
1337	61	female	29.1	0	yes	northwest	29141.36

1338 rows × 7 columns

In [19]:
```
# Changing categorical values to binary 1s and 0s
insurance['sex'] = insurance['sex'].map(lambda s :1  if s == 'female' else 0)
insurance['smoker'] = insurance['smoker'].map(lambda s :1  if s == 'yes' else 0)

insurance.head()
```

Out[19]:

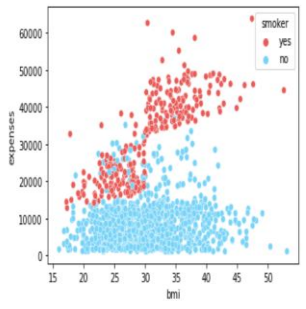
	age	sex	bmi	children	smoker	expenses
0	19	1	27.9	0	1	16884.92
1	18	0	33.8	1	0	1725.55
2	28	0	33.0	3	0	4449.46
3	33	0	22.7	0	0	21984.47
4	32	0	28.9	0	0	3866.86

3. Dataset Analysis and Observations

- Also, the region column does not affect the expenses more if we consider it during training the model. So I Decided to drop the column.
- Major factor in the dataset is smoker and bmi column if the customer is smoking then that customer is observed to be paying more amount for insurance premium (Expenses) as compared to non-smokers.
- For most adults, an ideal BMI is in the 18.5 to 24.9 range. Link to calculate bmi:- **https://www.cdc.gov/healthyweight/assessing/adult-widget/iframe.html**

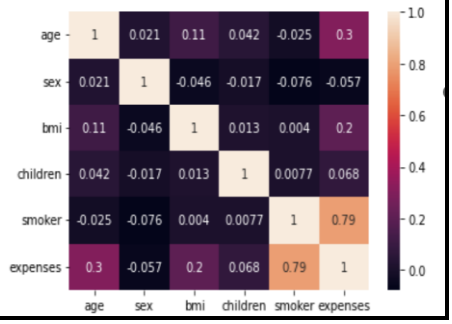
```
In [38]: import seaborn as sns  # Importing Seaborn library
         pal = ["#FA5858", "#58D3F7"] #COLOR CODE
         sns.scatterplot(x="bmi", y="expenses", data=insurance, palette=pal, hue='smoker')

Out[38]: <AxesSubplot:xlabel='bmi', ylabel='expenses'>
```




```
In [110]: import seaborn as sns
          corr_matrix = insurance.corr()
          sns.heatmap(corr_matrix, annot = True)

Out[110]: <AxesSubplot:>
```



4. Proposed Analytical/Prediction Model

- I have trained Linear Regression because it most basic model for predicting data and then I used Random Forest Regression Model because it gives most accurate prediction and good accuracy through cross validation.
- For both models, I trained dataset, and then compare both the model to know which one is best using R-Squared value.

5. Results and Discussions

1.)Linear Regression

```
In [116]: #Training and Testing..
          from sklearn.model_selection import train_test_split
          from sklearn.linear_model import LinearRegression
          from sklearn.metrics import mean_squared_error
          from math import sqrt

          X_train, X_test, y_train, y_test = train_test_split(X, y,test_size=0.20, random_state = 0)
          lr = LinearRegression().fit(X_train, y_train)

          y_train_pred = lr.predict(X_train)
          y_test_pred = lr.predict(X_test)

          print(lr.score(X_test, y_test))
          print('RMSE for Training:', np.sqrt(metrics.mean_squared_error(y_train, y_train_pred)))
          print('RMSE for Testing:', np.sqrt(metrics.mean_squared_error(y_test, y_test_pred)))

          0.7978966946106113
          RMSE for Training: 6150.385952617393
          RMSE for Testing: 5671.039719424828
```

2.)Random Forest Regression

```
In [109]: from sklearn.metrics import r2_score, mean_squared_error
          from numpy import sqrt
          # Prediction with training dataset:
          y_pred_RFR_train = random_forest_reg.predict(X_train)

          # Prediction with testing dataset:
          y_pred_RFR_test = random_forest_reg.predict(X_test)

          # Find training accuracy for this model:
          accuracy_RFR_train = r2_score(y_train, y_pred_RFR_train)
          print("Training Accuracy for Random Forest Regression Model: ", accuracy_RFR_train)

          # Find testing accuracy for this model:
          accuracy_RFR_test = r2_score(y_test, y_pred_RFR_test)
          print("Testing Accuracy for Random Forest Regression Model: ", accuracy_RFR_test)

          # Find RMSE for training data:
          RMSE_RFR_train = sqrt(mean_squared_error(y_train, y_pred_RFR_train))
          print("RMSE for Training Data: ", RMSE_RFR_train)

          # Find RMSE for testing data:
          RMSE_RFR_test = sqrt(mean_squared_error(y_test, y_pred_RFR_test))
          print("RMSE for Testing Data: ", RMSE_RFR_test)

          # Prediction with 10-Fold Cross Validation:
          y_pred_cv_RFR = cross_val_predict(random_forest_reg, X, y, cv=10)

          # Find accuracy after 10-Fold Cross Validation
          accuracy_cv_RFR = r2_score(y, y_pred_cv_RFR)
          print("Accuracy for 10-Fold Cross Predicted Random Forest Regression Model: ", accuracy_cv_RFR)

          Training Accuracy for Random Forest Regression Model:  0.8810737094617402
          Testing Accuracy for Random Forest Regression Model:  0.8939483596286619
          RMSE for Training Data:  4129.165103400073
          RMSE for Testing Data:  4108.0401170627365
          Accuracy for 10-Fold Cross Predicted Random Forest Regression Model:  0.8573765551768575
```

Conclusion: -

- If the RMSE for the test set is much higher than that of the training set, it is likely that you've badly over fit the data. A low RMSE value indicates that the simulated and observed data are close to each other showing a better accuracy. Thus lower the RMSE better is model performance.
- Moreover, we can see the accuracy of random forest regression is around 86% while the accuracy of linear regression is 80%.
- Hence, I can say that Random Forest Regression is best for this dataset for evaluation.