REPORT ON THE INVESTIGATION

ASSIGNMENT: Design and Application of a Machine Learning System for a Practical Problem.

Name of the student: SRIRAM MURALI

Registration Number: 2003290

MSc Intelligent Systems and Robotics

Module: CE802 – Machine Learning and Data Mining

Word Count: 1310

1. Introduction:

The objective of this project is to understand the behaviour of the customers and to predict the customer who purchased travel insurance in the company whether they would make a future claim or not. Using Machine Learning analysis, we can predict the higher probability and enhance the marketing strategy in the business. Machine learning models are used to make the predict. This helps the insurance company to be one step ahead of its competitors.

2. Problem definition and Algorithm:

Machine learning is a method of data analyses that automates analytical model building. In machine learning, the systems learn from the data, identify patterns and make decisions with minimal human intervention.

2.1 Task Definition

There are two types of machine learning problems that are being dealt with. The first type of the problem is a binary classification type problem where the aim of the work is to predict whether the customer of the travel insurance company will make a future claim or not.

This dataset contains 15 independent variables (F1, F2, F3, …, F15) and one dependent variable (Class Variable).

The second type of problem is a regression type problem. The aim of the work is to predict the value of the claim which is numerical.

This dataset contains 16 independent variables (F1, F2, F3, …, F16) and one dependent variable (Target Variable).

For both the cases, training data are provided where the model learns by making use of the information available provided in the dataset and there is a separate test dataset file, where the predictions can be performed from the learned model.

2.2 Algorithm Definition

Classification is a process of categorizing the given set of data into classes. The process starts with predicting the class of a given data points that are referred to as target, label or categories.

Regression is a predictive modelling task of approximating a mapping function from the input variables to a continuous output variable. The continuous output variable will be a real-value such as integer or floating-point value.

3. Experimental Evaluation:

3.1 Methodology

Task 1: To deal with classification problem, I have used three classification algorithms Decision Tree Classifier, Support Vector Machine Classifier and Random Forest Classifier.

While experimenting with the data, it is essential to perform the exploratory data analysis and data cleaning before modelling.

Independent Variables: F1 to F15
Target Variable: Class

The experimental methods handled are given below:

- First and foremost, the data needs to be checked whether the data points are balanced or not. This was done by visualizing the data using a pie chart and matplotlib library.

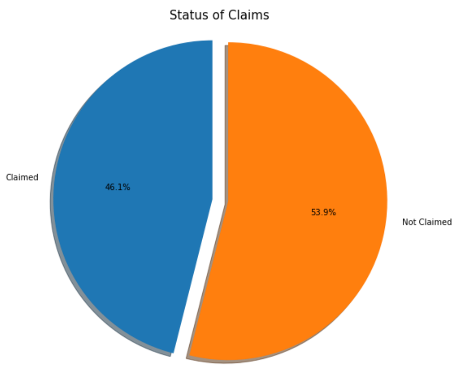
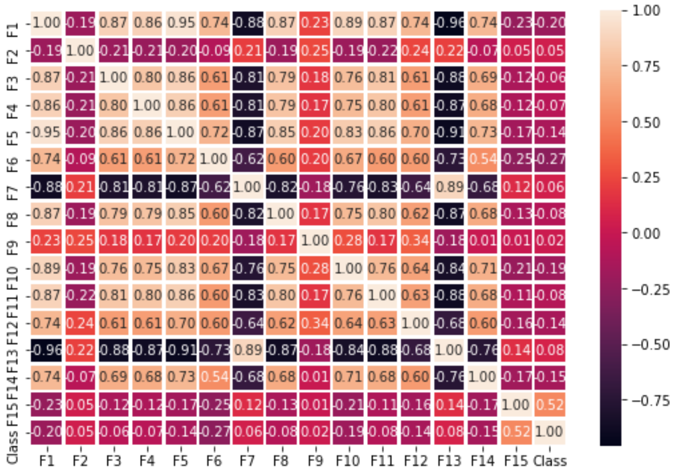


Figure 1: Pie Chart representation of the data

- The independent variable F15 had some missing values. The missing values are treated by replacing it with the median method.

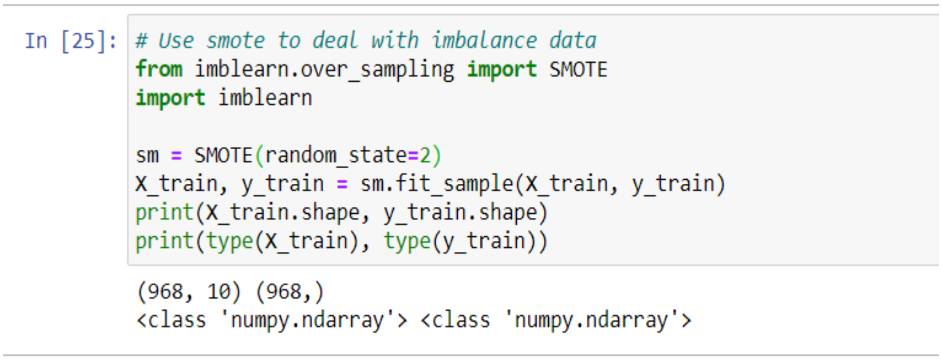
- A feature correlation heatmap is plotted using the seaborn library to find the highly correlated variables. Highly correlated variables may lead to multicollinearity problem that may affect the performance of the model.



	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	Class
F1	1.00	-0.19	0.87	0.86	0.95	0.74	-0.88	0.87	0.23	0.89	0.87	0.74	-0.96	0.74	-0.23	-0.20
F2	-0.19	1.00	-0.21	-0.21	-0.20	-0.09	0.21	-0.19	0.25	-0.19	-0.22	0.24	0.22	-0.07	0.05	0.05
F3	0.87	-0.21	1.00	0.80	0.86	0.61	-0.81	0.79	0.18	0.76	0.81	0.61	-0.88	0.69	-0.12	-0.06
F4	0.86	-0.21	0.80	1.00	0.86	0.61	-0.81	0.79	0.17	0.75	0.80	0.61	-0.87	0.68	-0.12	-0.07
F5	0.95	-0.20	0.86	0.86	1.00	0.72	-0.87	0.85	0.20	0.83	0.86	0.70	-0.91	0.73	-0.17	-0.14
F6	0.74	-0.09	0.61	0.61	0.72	1.00	-0.62	0.60	0.20	0.67	0.60	0.60	-0.73	0.54	-0.25	-0.27
F7	-0.88	0.21	-0.81	-0.81	-0.87	-0.62	1.00	-0.82	-0.18	-0.76	-0.83	-0.64	0.89	-0.68	0.12	0.06
F8	0.87	-0.19	0.79	0.79	0.85	0.60	-0.82	1.00	0.17	0.75	0.80	0.62	-0.87	0.68	-0.13	-0.08
F9	0.23	0.25	0.18	0.17	0.20	0.20	-0.18	0.17	1.00	0.28	0.17	0.34	-0.18	0.01	0.01	0.02
F10	0.89	-0.19	0.76	0.75	0.83	0.67	-0.76	0.75	0.28	1.00	0.76	0.64	-0.84	0.71	-0.21	-0.19
F11	0.87	-0.22	0.81	0.80	0.86	0.60	-0.83	0.80	0.17	0.76	1.00	0.63	-0.88	0.68	-0.11	-0.08
F12	0.74	0.24	0.61	0.61	0.70	0.60	-0.64	0.62	0.34	0.64	0.63	1.00	-0.68	0.60	-0.16	-0.14
F13	-0.96	0.22	-0.88	-0.87	-0.91	-0.73	0.89	-0.87	-0.18	-0.84	-0.88	-0.68	1.00	-0.76	0.14	0.08
F14	0.74	-0.07	0.69	0.68	0.73	0.54	-0.68	0.68	0.01	0.71	0.68	0.60	-0.76	1.00	-0.17	-0.15
F15	-0.23	0.05	-0.12	-0.12	-0.17	-0.25	0.12	-0.13	0.01	-0.21	-0.11	-0.16	0.14	-0.17	1.00	0.52
Class	-0.20	0.05	-0.06	-0.07	-0.14	-0.27	0.06	-0.08	0.02	-0.19	-0.08	-0.14	0.08	-0.15	0.52	1.00

Figure 2: Heatmap to visualize correlated variables from the dataset

- Since, we do not know the feature names and it is difficult to find out which feature should be left. So, statistical approach is followed using Variable Inflation Factor (VIF) and the features with a threshold of greater than 50 are removed.
- The outliers are dealt with using Interquartile range (IQR) method.
- In order to balance the dataset, Synthetic Minority Oversampling Technique (SMOTE) is used.



```
In [25]: # Use smote to deal with imbalance data
         from imblearn.over_sampling import SMOTE
         import imblearn

         sm = SMOTE(random_state=2)
         X_train, y_train = sm.fit_sample(X_train, y_train)
         print(X_train.shape, y_train.shape)
         print(type(X_train), type(y_train))

         (968, 10) (968,)
         <class 'numpy.ndarray'> <class 'numpy.ndarray'>
```

Figure 3: SMOTE Technique

<u>Result:</u>

The image below shows the confusion matrices of the different models. As we can see from the image, the Random forest classifier outperforms the other two models. The accuracy of random forest classifier is 0.74. Random forest classifier has achieved a precision and recall value of 0.76 and 0.75 respectively which is better when compared with other two models. Since, Accuracy is not the only parameter that describes how good a model is. Accuracy is used when the true positive and the true negative values are important. However, when false positives and false negatives are crucial, F1 score is used.

The F1 score of Random forest classifier stands at 0.75, whereas the F1 scores of SVM and Decision tree are 0.36 and 0.68 respectively. F1 score closer to 1 indicates perfect precision and recall which is reasonable using random forest classifier.

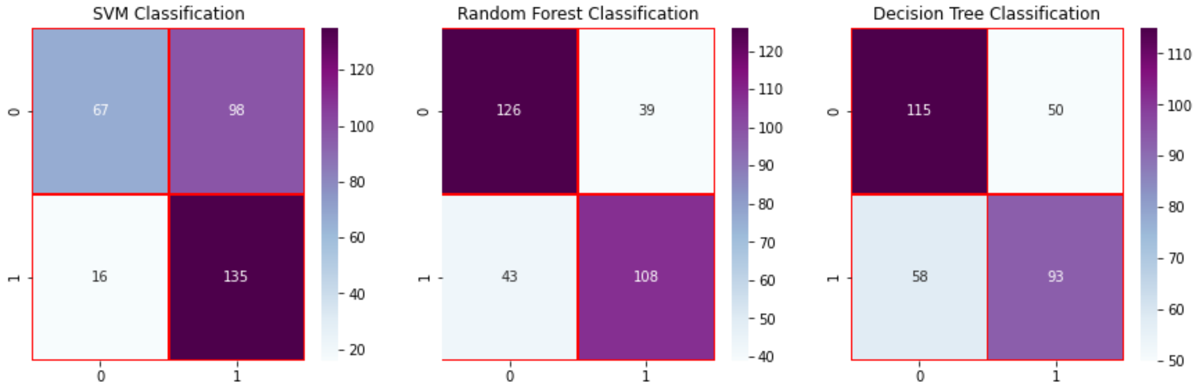


Figure 4: Confusion Matrices of the models

The ROC curve is an evaluation metric for binary classification problems. The ROC curve plots True Positive rate against False positive rate at various thresholds and essentially separates the signal from the noise. The more the area under the curve represents all the positive and the negative class points are perfectly distinguished by the model. Here Random forest classifier has an area of 0.81, which is excellent. By considering all these parameters, random forest classifier is the best model in our case scenario.

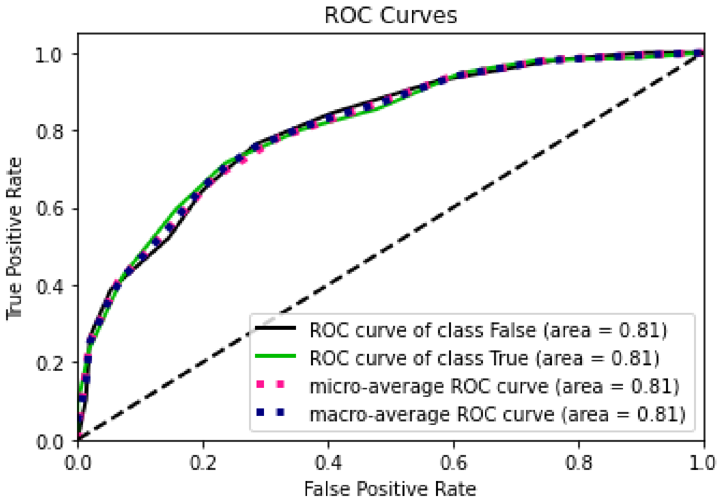


Figure 5: ROC Curve of Random Forest Classifier

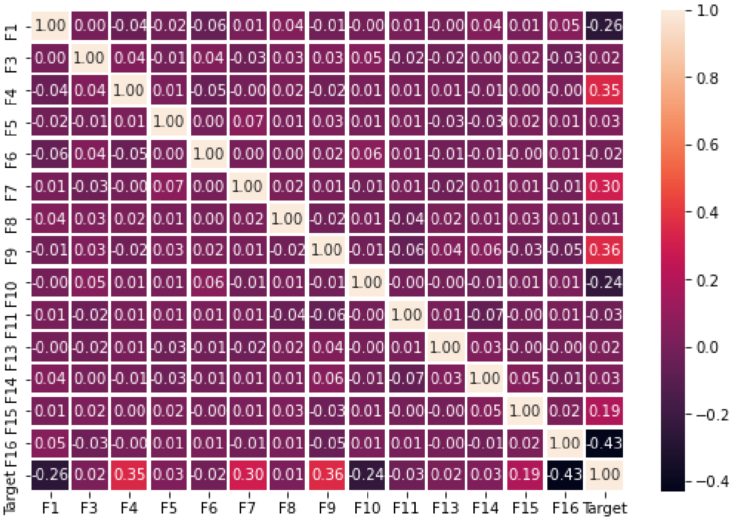
Compared with random forest classifier, SVM does not perform well when the dataset has more noise and it works well when there is a clear margin of separation between classes. Decision tree are also a good approach, but the decision trees are prone to errors in classification problem with relatively small number of training examples.

Task 2: To deal with regression problem, I have used three regression algorithms: Linear Regression, Support Vector Machine regressor and Random Forest regressor.

Data Name: CE802_P3_Data.csv
Independent variables: F1 to F16
Dependent variable: Target

There are various experimental strategies handled in this task.

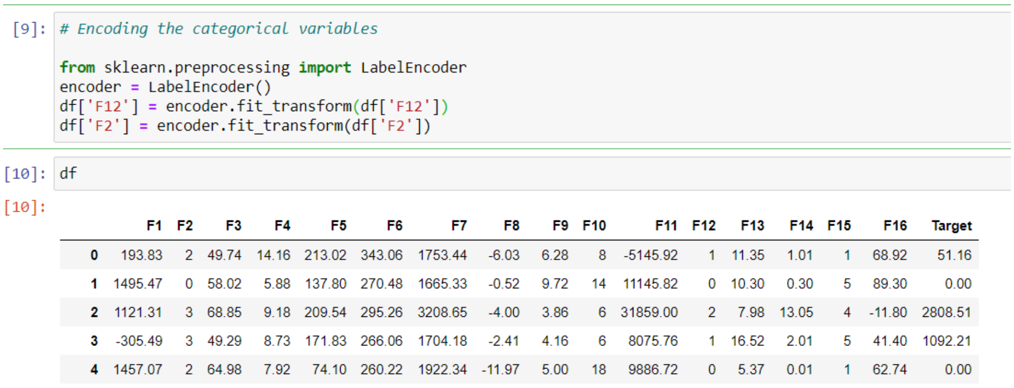
- The datapoints had some missing values and they are replaced with the median strategy.
- The features in these datasets are not correlated with each other, as a result none of the features are removed.



	F1	F3	F4	F5	F6	F7	F8	F9	F10	F11	F13	F14	F15	F16	Target
F1	1.00	0.00	-0.04	-0.02	-0.06	0.01	0.04	-0.01	-0.00	0.01	-0.00	0.04	0.01	0.05	-0.26
F3	0.00	1.00	0.04	-0.01	0.04	-0.03	0.03	0.03	0.05	-0.02	-0.02	0.00	0.02	-0.03	0.02
F4	-0.04	0.04	1.00	0.01	-0.05	-0.00	0.02	-0.02	0.01	0.01	0.01	-0.01	0.00	-0.00	0.35
F5	-0.02	-0.01	0.01	1.00	0.00	0.07	0.01	0.03	0.01	0.01	-0.03	-0.03	0.02	0.01	0.03
F6	-0.06	0.04	-0.05	0.00	1.00	0.00	0.00	0.02	0.06	0.01	-0.01	-0.01	-0.00	0.01	-0.02
F7	0.01	-0.03	-0.00	0.07	0.00	1.00	0.02	0.01	-0.01	0.01	-0.02	0.01	0.01	-0.01	0.30
F8	0.04	0.03	0.02	0.01	0.00	0.02	1.00	-0.02	0.01	-0.04	0.02	0.01	0.03	0.01	0.01
F9	-0.01	0.03	-0.02	0.03	0.02	0.01	-0.02	1.00	-0.01	-0.06	0.04	0.06	-0.03	-0.05	0.36
F10	-0.00	0.05	0.01	0.01	0.06	-0.01	0.01	-0.01	1.00	-0.00	-0.00	-0.01	0.01	0.01	-0.24
F11	0.01	-0.02	0.01	0.01	0.01	0.01	-0.04	-0.06	-0.00	1.00	0.01	-0.07	-0.00	0.01	-0.03
F13	-0.00	-0.02	0.01	-0.03	-0.01	-0.02	0.02	0.04	-0.00	0.01	1.00	0.03	-0.00	-0.00	0.02
F14	0.04	0.00	-0.01	-0.03	-0.01	0.01	0.01	0.06	-0.01	-0.07	0.03	1.00	0.05	-0.01	0.03
F15	0.01	0.02	0.00	0.02	-0.00	0.01	0.03	-0.03	0.01	-0.00	-0.00	0.05	1.00	0.02	0.19
F16	0.05	-0.03	-0.00	0.01	0.01	-0.01	0.01	-0.05	0.01	0.01	-0.00	-0.01	0.02	1.00	-0.43
Target	-0.26	0.02	0.35	0.03	-0.02	0.30	0.01	0.36	-0.24	-0.03	0.02	0.03	0.19	-0.43	1.00

Figure 6: Correlation Heatmap

- The feature F2 and F12 has categorical variables. Label Encoder technique is used to convert the categorical variables into numerical variables.



```
[9]: # Encoding the categorical variables

     from sklearn.preprocessing import LabelEncoder
     encoder = LabelEncoder()
     df['F12'] = encoder.fit_transform(df['F12'])
     df['F2'] = encoder.fit_transform(df['F2'])
```

```
[10]: df
```

[10]:

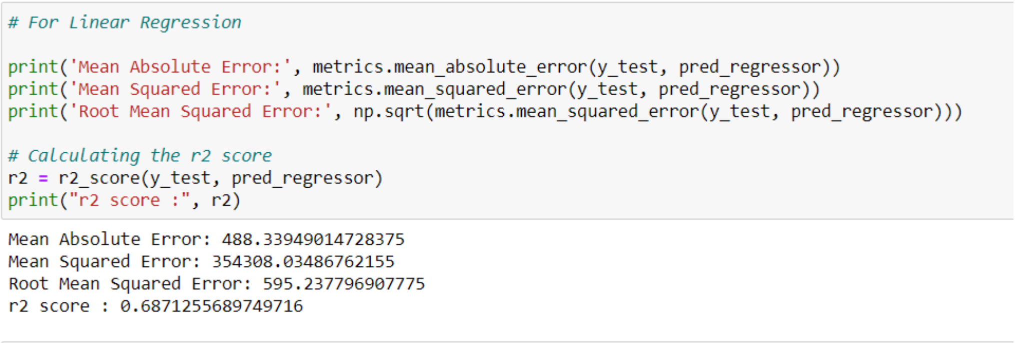
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	Target
0	193.83	2	49.74	14.16	213.02	343.06	1753.44	-6.03	6.28	8	-5145.92	1	11.35	1.01	1	68.92	51.16
1	1495.47	0	58.02	5.88	137.80	270.48	1665.33	-0.52	9.72	14	11145.82	0	10.30	0.30	5	89.30	0.00
2	1121.31	3	68.85	9.18	209.54	295.26	3208.65	-4.00	3.86	6	31859.00	2	7.98	13.05	4	-11.80	2808.51
3	-305.49	3	49.29	8.73	171.83	266.06	1704.18	-2.41	4.16	6	8075.76	1	16.52	2.01	5	41.40	1092.21
4	1457.07	2	64.98	7.92	74.10	260.22	1922.34	-11.97	5.00	18	9886.72	0	5.37	0.01	1	62.74	0.00

Figure 7: Label Encoder technique

- The outliers were found out and they were removed using Z-Score technique.

Result:

The image below shows the regression evaluation metric scores of the different models.



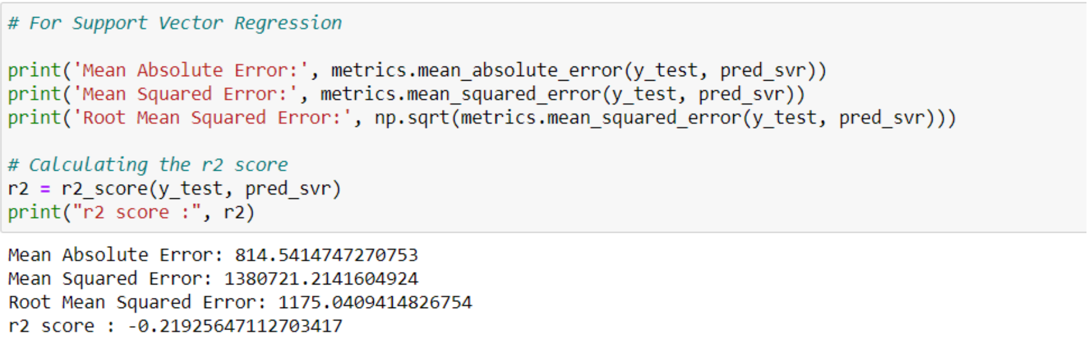
```
# For Linear Regression

print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, pred_regressor))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, pred_regressor))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, pred_regressor)))

# Calculating the r2 score
r2 = r2_score(y_test, pred_regressor)
print("r2 score :", r2)
```

```
Mean Absolute Error: 488.33949014728375
Mean Squared Error: 354308.03486762155
Root Mean Squared Error: 595.237796907775
r2 score : 0.6871255689749716
```

Figure 8: Linear Regression evaluation metrics scores



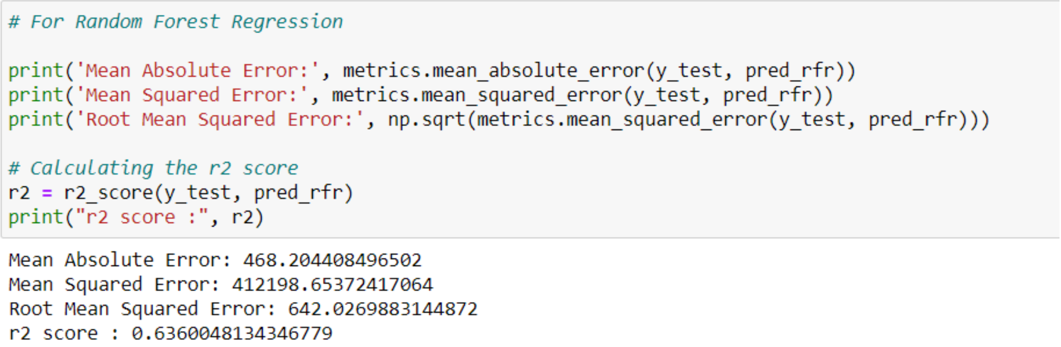
```
# For Support Vector Regression

print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, pred_svr))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, pred_svr))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, pred_svr)))

# Calculating the r2 score
r2 = r2_score(y_test, pred_svr)
print("r2 score :", r2)
```

```
Mean Absolute Error: 814.5414747270753
Mean Squared Error: 1380721.2141604924
Root Mean Squared Error: 1175.0409414826754
r2 score : -0.21925647112703417
```

Figure 9: Support Vector Machine Regression evaluation metrics scores



```
# For Random Forest Regression

print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, pred_rfr))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, pred_rfr))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, pred_rfr)))

# Calculating the r2 score
r2 = r2_score(y_test, pred_rfr)
print("r2 score :", r2)
```

```
Mean Absolute Error: 468.204408496502
Mean Squared Error: 412198.65372417064
Root Mean Squared Error: 642.0269883144872
r2 score : 0.6360048134346779
```

Figure 10: Random Forest Regression evaluation metrics scores

The Mean Squared Error (MSE) tells how close the regression line to the set of points. MSE value does not have any specific value. Lower the MSE score better the model is. We can see that the MSE score of Linear Regression is lower compared to other models. The other metric used to evaluate the model performance is RMSE score. The lower the RMSE score, better the model fits. The RMSE score of linear regression model is lower compared to the other two models used in this problem. This indicates that the performance of the linear regression is better when compared to the other two.

R2 score is the statistical measure of how close the data are fitted to the regression line. R2 is always between 0 and 100%. R2 score of closer to 100% represents a good model which is achieved by Linear regression (69%) in our case.

One of the weakness of random forest regression is that, this regressor is unable to discover trends that would enable it in extrapolating values which fall outside the training set. In this case, model such as Linear regression or SVM regression can be used. The SVM model does not perform well when the model has more noise and it also has overfitting problem. Linear regression also has overfitting problem but it can be avoided easily using cross validation and regularization techniques.

4. Conclusion:

Present day technologies are moving amazingly quick making their ways into different field of the business. The point of applying machine learning methods in the insurance company is to improve the marketing strategies, improve the business, enhance the income and lessen the costs. Our machine learning models are also doing the same in this project that would help the insurance company to predict the future.

To sum up, despite some disadvantages, Random forest classifier model is best used when there are lot of noise in the dataset. They are robust and not affected by outliers and missing data. And in regression problem, Linear regression performed best in our use case. Even though it is affected by overfitting problem it can be easily overcome by cross validation approach.

References:

1) scikit-learn: machine learning in Python — scikit-learn 0.24.0 documentation (scikit-learn.org)
2) https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/
3) https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html
4) https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5
5) https://towardsdatascience.com/heatmap-basics-with-pythons-seaborn-fb92ea280a6c