**Report on the Investigation**

Word count:

# **Comparative study on the Part 2**

## Brief description of the methods’ implementation

Since, the next step includes finding the best model with optimal parameters using GridSearchCV, there were made up sets for each classifier with various parameters to test that are reflected in the tables below.

### ***Decision trees***

Table 1 – The set of various parameters for the Decision tree method.

|  |  |  |
| --- | --- | --- |
| The name of a parameter | The range of a parameter | The purpose of a parameter |
| *criterion* | “gini”,”entropy” | Tree branching quality estimation |
| *max\_depth* | 1-29 ,”None” | Specification of a decision tree’s maximum depth |
| *min\_samples\_split* | 2-7 | Specification of a minimum number of samples in the tree to split |
| *min\_samples\_leaf* | 1-15 | Specification the minimum number of a tree leaves |

### ***K-nearest neighbours***

Table 2 – The set of various parameters for the K-nearest neighbours’ method.

|  |  |  |
| --- | --- | --- |
| The name of a parameter | The range of a parameter | The purpose of a parameter |
| *n\_neighbors* | 1-150 | K- number of the most similar samples to the considered one |

### ***Support vector machine***

Table 3 – The set of various parameters for the Support vector machine method with a radial basis function kernel (SVM with rbf).

|  |  |  |
| --- | --- | --- |
| The name of a parameter | The range of a parameter | The purpose of a parameter |
| *gamma* | 0.0001-0.050 | A specified coefficient for the kernel |

Table 4 – The set of various parameters for the SVM with a linear function kernel.

|  |  |  |
| --- | --- | --- |
| The name of a parameter | The range of a parameter | The purpose of a parameter |
| *C* | 0.1-10.0 | The regularization parameter of the SVM |

Besides these sets, *GridSearchCV* for each model is initialized with several other main parameters such as:

* *cv = 10* – a number of cross-validation folds
* *scoring = “accuracy”* – scoring a model measuring a classification accuracy
* *n\_jobs = -1* – specification of a maximum number of available CPUs to speed up finding an optimal model

## Obtained results

In the table below the best parameters for each classifier, that were revealed by GridSearchCV, can be seen.

Table 6 – The best parameters for each model

|  |  |
| --- | --- |
| Parameter name | Parameter value |
| K-nearest neighbours | |
| *n\_neighbors* | 43 |
| SVM with rbf | |
| *gamma* | 0.0097 |
| SVM with a linear function kernel | |
| *C* | 9.9 |
| Decision trees classifier | |
| *criterion* | 'gini' |
| *max\_depth* | 9 |
| *min\_samples\_split* | 2 |
| *min\_samples\_leaf* | 9 |

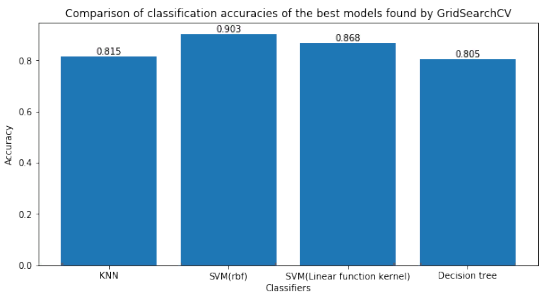


Figure 1 – A bar chart of mean classification accuracy comparison among four methods after performed cross-validation in the GridSearchCV.

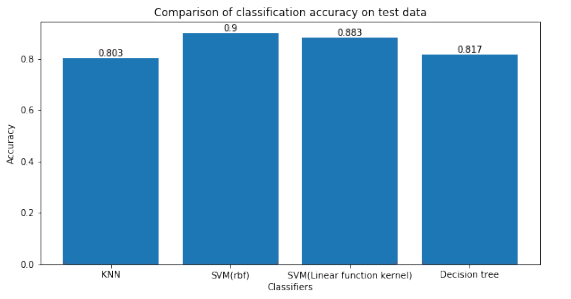


Figure 2 – A bar chart of mean classification accuracy comparison among three methods on the prepared test share of the training set.

According to figures 1,2 above, it can be noticed that the largest recession of classification accuracy is inherent in k-nearest neighbours, while the other classifiers’ accuracies have increased their ones but by a small extent. Besides that, the SVM with rbf model has reached the highest accuracy, while the k-nearest neighbours model– the lowest one.

Apart from the accuracies results on figures 3,4 below it is reflected more information about other metrics such as precision, recall, f1-score retrieved in the result of the prediction.

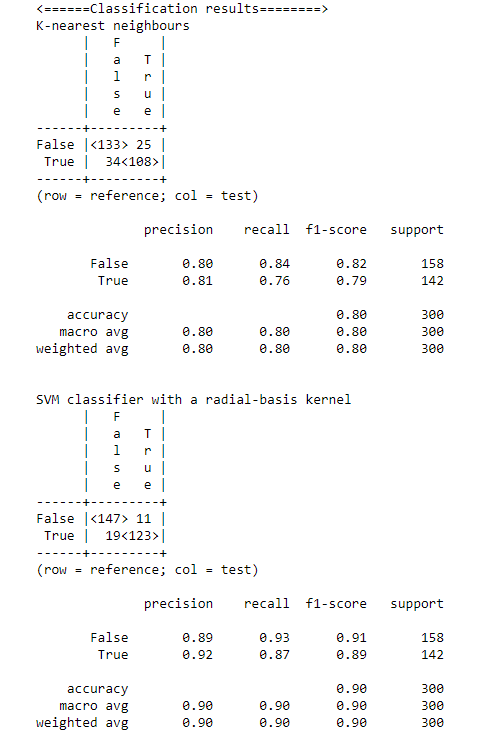


Figure 3 – A detailed report on prediction performance for K-NN and SVM with rbf.

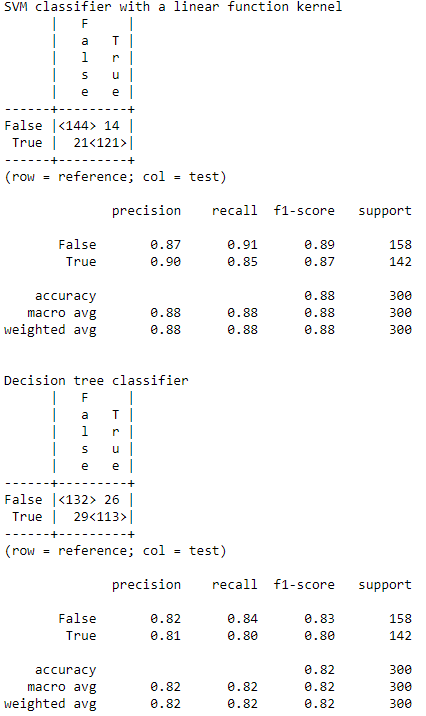


Figure 4 – A detailed report on prediction performance for K-NN and SVM with rbf.

As it can be clearly seen from figure 3,4 above that a share of correct predictions is the most significant in the confusion matrix of the SVM classifier with rbf.

Eventually, in the result of prediction of target values from the test dataset by trained classification models the following result table was made up. This table is depicted below.

Table 7 – Comparative table of results obtained from applied classification models.

| Predicted # | Predicted values | | | |
| --- | --- | --- | --- | --- |
| K-nearest neighbours | Support vector machine with a linear function kernel | Support vector machine with rbf | Decision trees |
| 0 | False | True | False | False |
| 1 | True | False | True | True |
| 2 | **True** | **True** | **True** | **True** |
| 3 | **False** | **False** | **False** | **False** |
| 4 | **False** | **False** | **False** | **False** |
| 5 | **False** | **False** | **False** | **False** |
| 6 | **False** | **False** | **False** | **False** |
| 7 | **True** | **True** | **True** | **True** |
| 8 | True | True | True | False |
| 9 | **True** | **True** | **True** | **True** |
| 10 | **True** | **True** | **True** | **True** |
| 11 | **True** | **True** | **True** | **True** |
| 12 | **False** | **False** | **False** | **False** |

While considering results from the sample above we can notice one interesting detail regarding the content of the table. In essence, the table contains two clusters of identical results from each of the methods. Thus, this feature indicates the consistency between the forecast results of the same data and that those classifiers have very good performance.

## Summary

According to the obtained results from the conducted investigation on classification of the customers, we should state that we have managed to determine whether a certain customer shall obtain a discounted premium and get insured for the next time or not by applying the ML techniques.

The list of implemented techniques includes a decision tree classifier, a support vector machine with a radial-basis function kernel, a support vector machine with a linear function kernel and a k-nearest neighbours’ classifier. Thus, by applying unique features of each method to solve the given task, all of these classifiers have provided rather fine performance both on train and test dataset.

By the way, we can determine strengths and weaknesses of applied methods by considering the obtained results.

For example, the main strength of the decision tree is its powerful ability to find hidden rules from the given data and represent them as a tree. It allows anyone to better understand what are the logical connections between different variables. On the other hand, in our case, the obtained tree was too large, so it turned out hard to perceive anything. Besides that, this method is not powerful enough to achieve a high accuracy on rather large volumes of real data. In addition, it has too many parameters to adjust, so it could be a hard task to find the optimal ones to reach a decent level of precision.

In contrast, both SVM implementations have provided decent performance on such large datasets, but the best one had a radial-basis kernel which allowed it to achieve 90% of accuracy on the test data. Therefore, the main strength of them is their excellent performance and applicability to vast amount of information. However, the weakness of SVM is its sensibility to a scale of data. This fact indicates that a researcher has to investigate how to standardise his raw data in order to obtain a precise model.

The last model – k-nearest neighbours has got the similar accuracy to the decision tree one. In essence, it also has the same flaw as decision tree method – sensibility to a volume of data, so it does not perform enough well on large datasets.

On the other hand, it does not have many parameters to adjust. So, due to the basic idea embedded in its algorithm, it has proven its applicability to the given datasets and demonstrated a decent performance indeed.

All in all, in the result if the conducted investigation on the ML methods application to classify customers, it should be claimed that the given task can be solved using such classifiers as: decision trees, support vector machine with a radial-basis function kernel, a support vector machine with a linear function kernel and a k-nearest neighbours. Thus, in essence, a travel insurance company can apply the aforementioned techniques to get to know whether they should suggest a discounted premium in further or not.

# **Comparative study on the Part 3**

## Brief description of implemented methods

Since, the next step includes finding the best model with optimal parameters using GridSearchCV, there were made up sets for each method with various parameters to test that are reflected in the tables below.

### ***Support vector machine***

Table 5 – The set of various parameters for the Support vector machine method with a radial basis function kernel.

|  |  |  |
| --- | --- | --- |
| The name of a parameter | The range of a parameter | The purpose of a parameter |
| *C* | 0.1-5.0 | The regularization parameter of the SVM |

### ***Multi-layer perceptron***

Table 5 – The set of various parameters for the Multi-layer perceptron method.

|  |  |  |
| --- | --- | --- |
| The name of a parameter | The range of a parameter | The purpose of a parameter |
| *hidden\_layer\_sizes* | (50),  (50,50),  (50,50,50),  (50,50,50,50),  (50,50,50,50,50),  (50,50,50,50,50,50),  (50,50,50,50,50,50,50) | Content description of all layers in the neural network model |

### ***Linear regression***

Since this method does not have any specific hyperparameters to adjust, it does not need the GridSearchCV step either. Instead, it only uses cross\_val\_score method for performing cross-validation and estimate a mean squared error of prediction on the test data.

Besides those sets, Cross\_val\_score and *GridSearchCV* for each model are initialized with several other main parameters such as:

* *cv = 5* – a number of cross-validation folds
* *scoring = “neg\_mean\_squared\_error”* – scoring a model measuring a classification accuracy
* *n\_jobs = -1* – specification of a maximum number of available CPUs to speed up finding an optimal model

## Obtained results

In the table below the best parameters for each model, that were revealed by GridSearchCV, can be seen.

Table 6 – The best parameters for each model

|  |  |
| --- | --- |
| Parameter name | Parameter value |
| Multi-layer perceptron | |
| *hidden\_layer\_sizes* | (50, 50) |
| Support vector machine | |
| *C* | 2.6 |

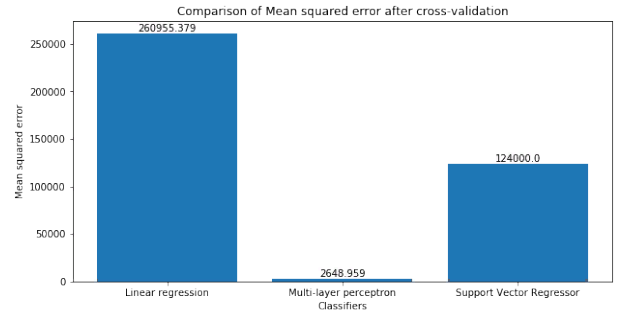


Figure 1 – A bar chart of the mean squared error comparison among three methods after cross-validation in the GridSearchCV.

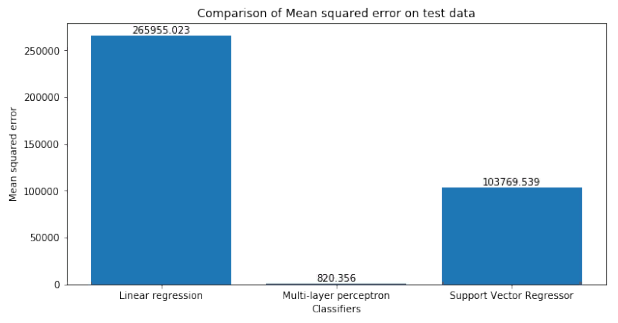


Figure 2 – A bar chart of the mean squared error comparison among three methods on the prepared test share of the training set.

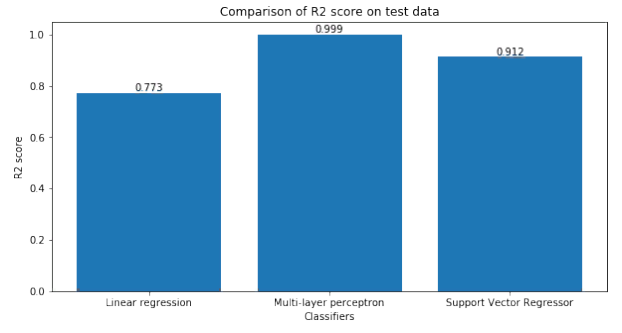


Figure 3 – A bar chart of the R2 score comparison among three methods on the prepared test share of the training set.

Considering the results reflected on the figures 1,2,3 above, it should be stated that even after testing the multi-layer perceptron model on the unseen data its mean squared error has decreased. Moreover, the R2 score on the test data is almost one. Thus, both of these facts confirms the assumption that the model was trained perfectly and therefore has a relatively tiny prediction error.

The difference between accuracies of each method is demonstrated better in the table 7 below.

Table 7 – The table of comparison between actual values and predicted ones.

| Record | Actual | Predicted values | | |
| --- | --- | --- | --- | --- |
| Multi-layer perceptron | Support vector machine | Linear regression |
| **0** | 475.90 | 488.710508 | 588.486319 | 726.400002 |
| **1** | 171.50 | 165.830640 | 183.146305 | 385.820427 |
| **2** | 728.66 | 785.677288 | 913.438289 | 956.316462 |
| **3** | 0.00 | -11.795736 | -43.762342 | -344.788691 |
| **4** | 580.16 | 591.074779 | 776.467087 | 906.901620 |
| **5** | 0.00 | 1.369397 | 253.393827 | 516.467786 |
| **6** | 0.00 | 1.572601 | -33.665007 | -425.026423 |
| **7** | 2031.42 | 1969.674818 | 1764.234917 | 1761.464708 |
| **8** | 0.00 | 0.533966 | -107.127773 | -162.947552 |
| **9** | 0.00 | -5.058856 | 533.353645 | 704.834123 |
| **10** | 2908.23 | 2864.793513 | 3967.994901 | 2647.761125 |

While comparing the actual values and predicted ones, it can be noticed that the Multi-layer perceptron model has almost identical results, what indicates its reliability and indeed excellent performance.

In contrast to it, the Linear regression model has shown the worst prediction results that are rather distant from the real values.

By the way, it is worth mentioning a fact that SVM also finely managed with the task, because in most cases its forecasted values are rather similar to the actual ones, despite having sometimes a large bias.

Besides the table above, the significance of squared errors for each prediction result is reflected on the figures 4,5,6 below.

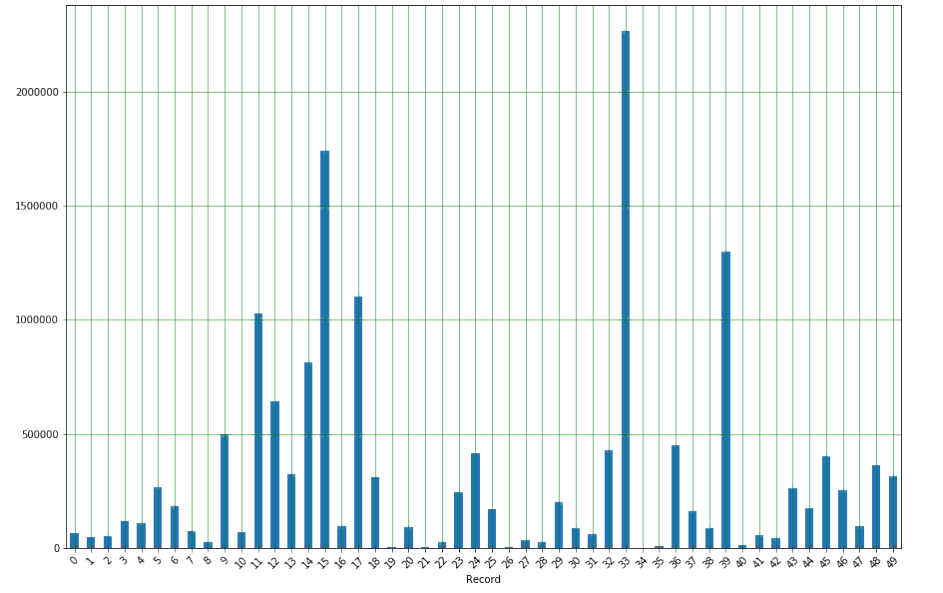


Figure 4 – A bar chart of the first 50 squared errors of the linear regression model for each prediction result.

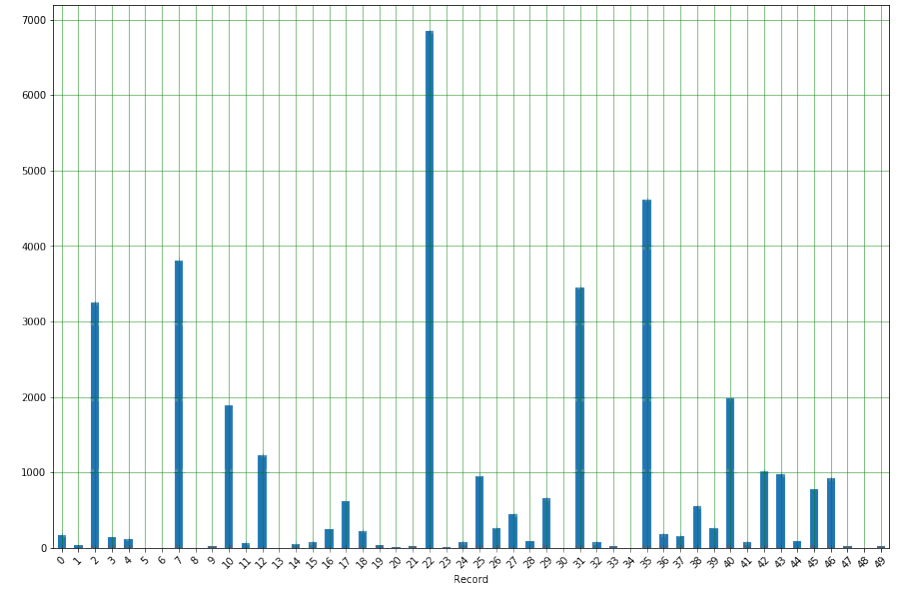


Figure 5 – A bar chart of the first 50 squared errors of the multi-layer perceptron model for each prediction result.

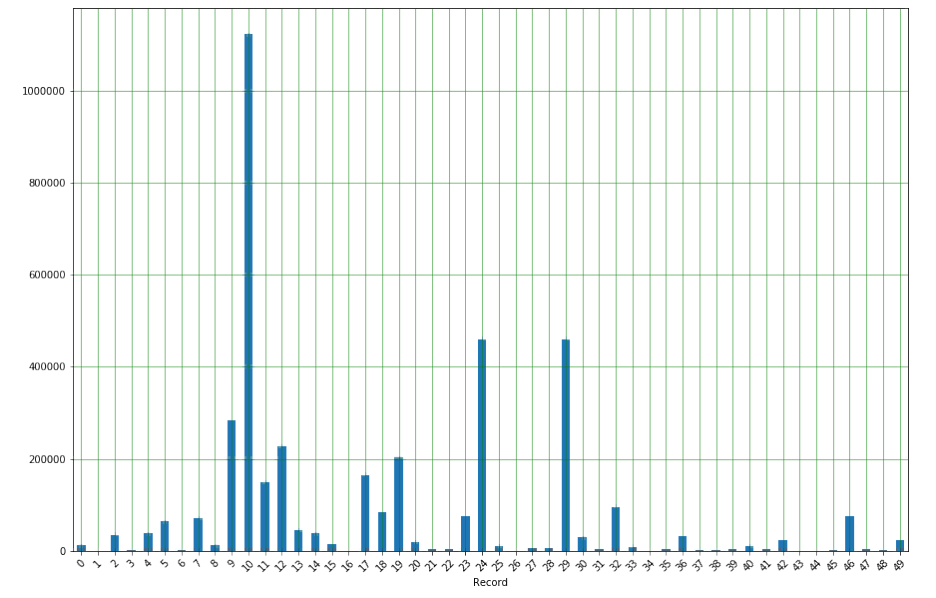


Figure 6 – A bar chart of the first 50 squared errors of the support vector machine model for each prediction result.

Thus, while taking a look at the figures 4,5,6 above, it can be realized how significant is the difference between ranges of each methods’ squared errors. So, the narrowest range of errors certainly has multi-layer perceptron and the widest one has the linear regression model.

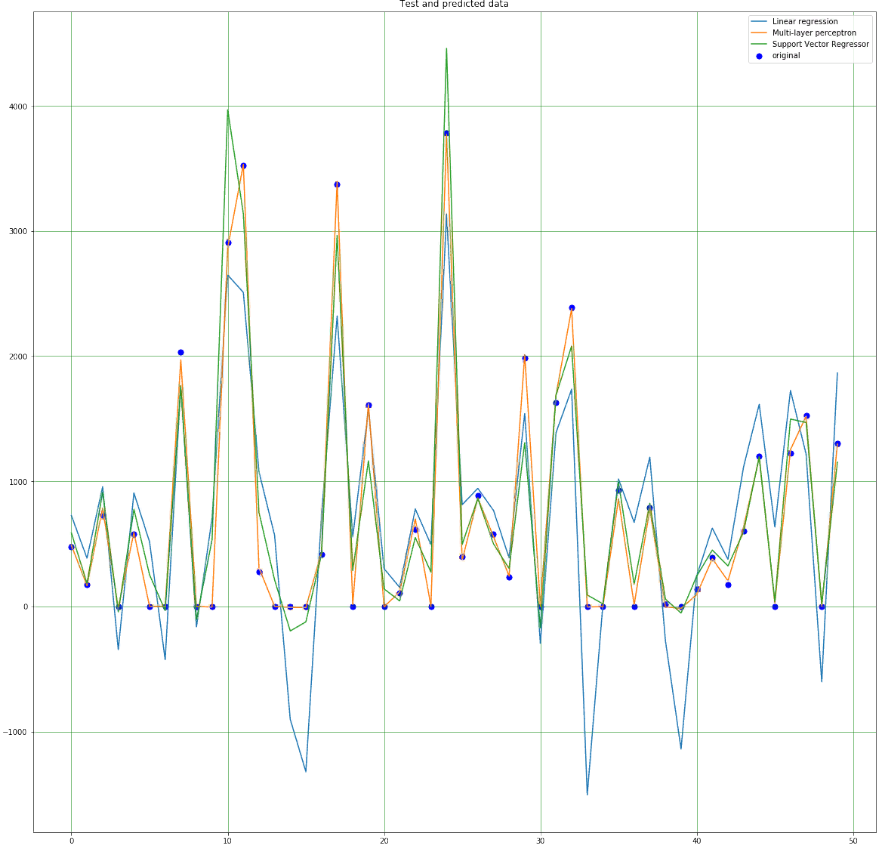


Figure 7 – The comparative plot of actual values from the test data and predicted ones by each method.

On the figure 7, it is clearly demonstrated how significant in some spots is the difference between prediction results of each method. Thus, barely noticeable deviations are inherent in multi-layer perceptron’s model, while the linear regression model has the most considerable ones. By the way, the support vector machine model in turn has some obvious deviations around the actual values, but they occur rather occasionally.

The prediction results of the final test dataset are demonstrated in the table 8 below.

Table 8 – The table of comparison between actual values and predicted ones.

| Predicted # | Multi-layer perceptron | Support Vector Regressor | Linear regression |
| --- | --- | --- | --- |
| 0 | 406.311946 | 537.405356 | 739.920670 |
| 1 | -0.019752 | -99.969252 | 275.760310 |
| 2 | 3694.361982 | 3248.694602 | 2653.437827 |
| 3 | 2482.271118 | 2138.339931 | 1765.704389 |
| 4 | -2.509869 | -569.030387 | 151.045057 |
| 5 | 2092.810065 | 2252.262093 | 2100.274914 |
| 6 | 214.274578 | 669.279691 | 744.634067 |
| 7 | 476.626691 | 906.738057 | 842.083932 |
| 8 | 1.857013 | -186.832498 | -107.316205 |
| 9 | 2196.381988 | 1998.867712 | 1658.471504 |
| 10 | 757.385198 | 720.911585 | 890.819016 |

From the predicted values of the best method (Multi-layer perceptron) the assumption can be made about what the real values are. For example, results -0.019752, -2.509869 and 1.857013 for the customers 1, 4, 8, accordingly, indicate that these customers shall not obtain a discounted premium for the next time. In turn, both other methods also provide relatively small values for these customers what implies that those people are unlikely to get insured.

## Summary