CAR INSURANCE PLAN

Machine Learning I Project

Lianjie Shan

Xi Zhang

1. Introduction

1.1 How dose machine learning work on car insurance?

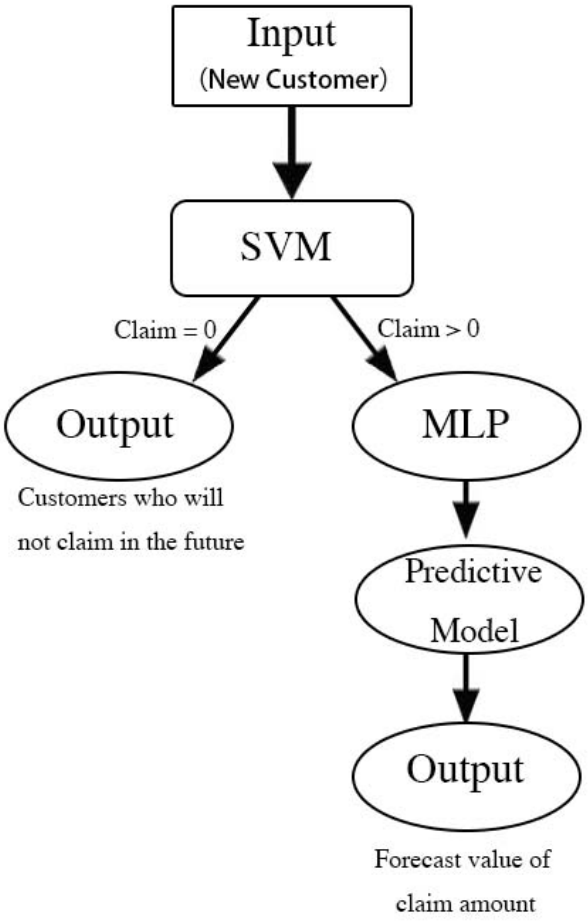
As the automobile gradually becomes the necessity of every family, the car insurance also becomes more and more prosperous for auto insurance companies, to maximize revenue, they must sell corresponding insurance plans to different customers. However, because the types of customers are so diverse and the correlation between the characteristics is not obvious, the use of simple statistics cannot enable insurance companies to make accurate judgments about customers. With the advent of machine learning, more models are available to learn data in depth. Thus, more accurate predictions can be achieved.

1.2 Definition of the Problem

Our initial plan is to use the existing data of the auto insurance company to train the models, so that when new customers come in, we can use these models to predict whether they will claim or not according to their characteristics. With only two types of target data, SVM became our first choice.

After determining whether the customer will claim in the future, we divide the customers who will claim and use the new model to predict the amount.Due to the high tolerance of multilayer pecenptron to data and its better handling of specific values, we decided to use MLP network to predict specific values.

After training the MLP model with existing data, if the evaluation is reliable, we can use the characteristics of new customers to predict their future claims. But before we can implement this process, we need to start with data processing.



2. Data Description

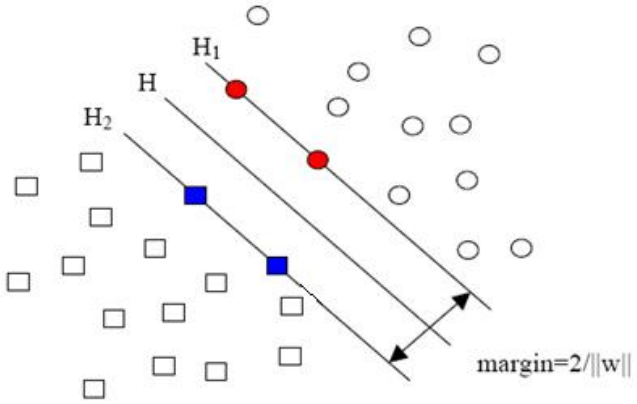
The data we chose was released by Kaggle, an open-source data site. The distributor xiaomengsun published it in 2018. It is made up of a record of 10,302 observations and 27 variables. This data can be downloaded from the following websites for study and research:

https://www.kaggle.com/xiaomengsun/car-insurance-claim-data

3. Machin learning Algorithm

3.1. Support Vector Method (SVM)

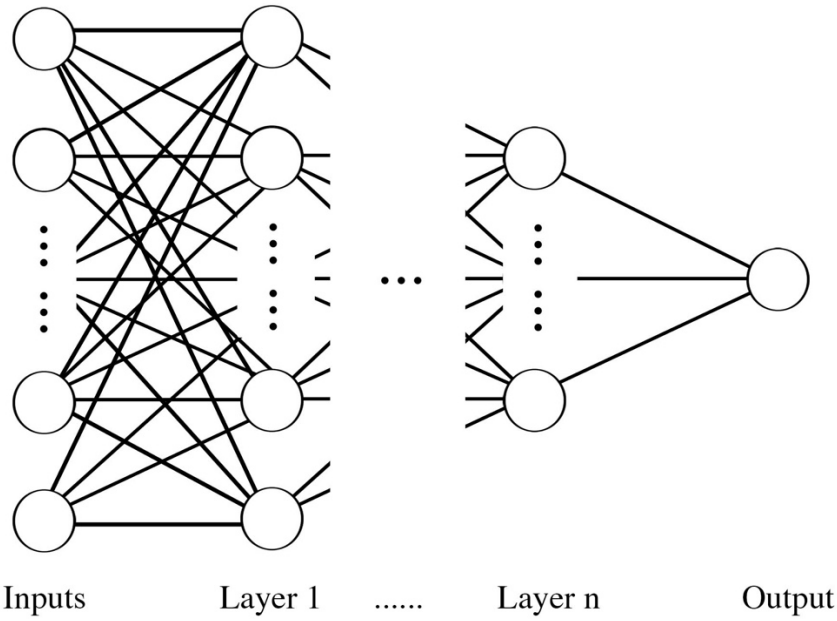
Support Vector Method (SVM) as a popular machine learning tool is most used for classification and regression. Generally speaking, SVM tries to find a plane that has the maximum margin and the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence. The following plot shows the basic idea of SVM



Moreover, SVM classification is considered a nonparametric method because of the kernel functions. To be specific, kernel function can split the points which cannot be regarded as linear problems.

3.2. Multilayer Perceptron Network

For this research, we used the multilayer perceptron network (MLP) to compute the prediction. The basic logic of MLP is as fowling:



In the part of model selection, we hope to train a neural network to achieve our goal because of the complexity of data and the relatively vague correlation between variables. MLP has a high degree of parallel processing, a high degree of nonlinear global function, good fault tolerance, associative memory function, very strong adaptive, self-learning function, so we finally decided to use MLP multilayer perceptron.

Milo Harper in his project shows a two-layer perceptron model to solve the XOR problem. He uses Sigmoid as the activation function and illustrates back propagation, which is perfectly fitting our research. It guides us in the following research.

4. Experimental Setup and Results

4.1.Data Preparation

	ID	KIDSDRIV	BIRTH	AGE	HOMEKIDS	YOJ	INCOME	PARENT1	HOME_VAL	MSTATUS	...	CAR_TYPE	RED_CAR	OLDCLAIM	C
0	63581743	0	16MAR39	60.0	0	11.0	$67,349	No	$0	z_No	...	Minivan	yes	$4,461	
1	132761049	0	21JAN56	43.0	0	11.0	$91,449	No	$257,252	z_No	...	Minivan	yes	$0	
2	921317019	0	18NOV51	48.0	0	11.0	$52,881	No	$0	z_No	...	Van	yes	$0	
3	727598473	0	05MAR64	35.0	1	10.0	$16,039	No	$124,191	Yes	...	z_SUV	no	$38,690	
4	450221861	0	05JUN48	51.0	0	14.0	NaN	No	$306,251	Yes	...	Minivan	yes	$0	

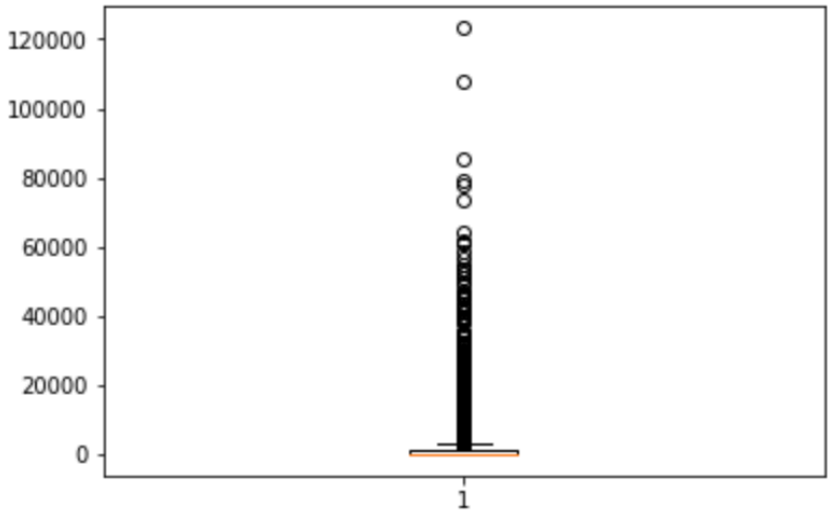
To ensure the reliability of the final result and the efficiency of the data algorithm, we first cut out the repeated or useless variables. This reduces our column count from 27 to 22.

Later, to avoid python syntax errors, I unified all value types as floats, removed the currency symbol and digitalized the data of "Yes", "No", gender and other categories.

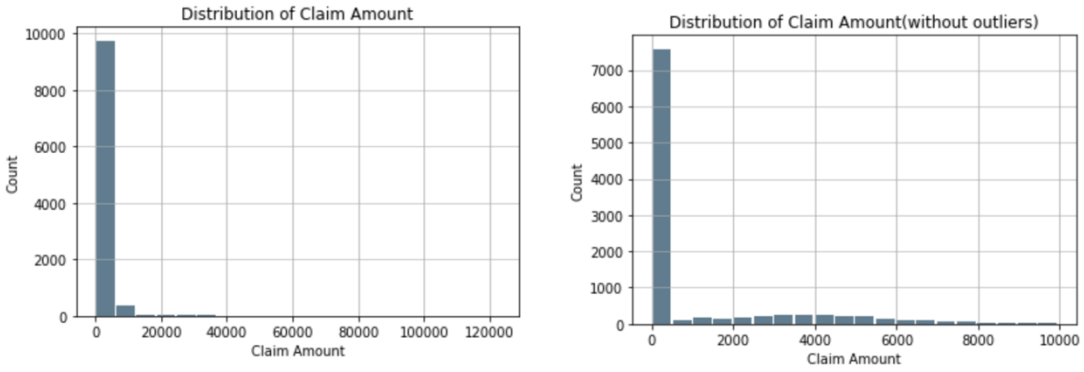
Claim Amount, our target column, as the most important data, we have an overview of it. From its average, median, quartile and box chart, it can be seen that the majority of claim amount are zero.

```
Mean                    : 1511.2664531158998
Median                  : 0.0
Minimum                 : 0.0
Maximum                 : 123247.0
25th percentile of arr : 0.0
50th percentile of arr : 0.0
75th percentile of arr : 1144.75
```

Further, because there are too many outliers, the whole box diagram is compressed very flat, so next we will try our best to remove these outliers.



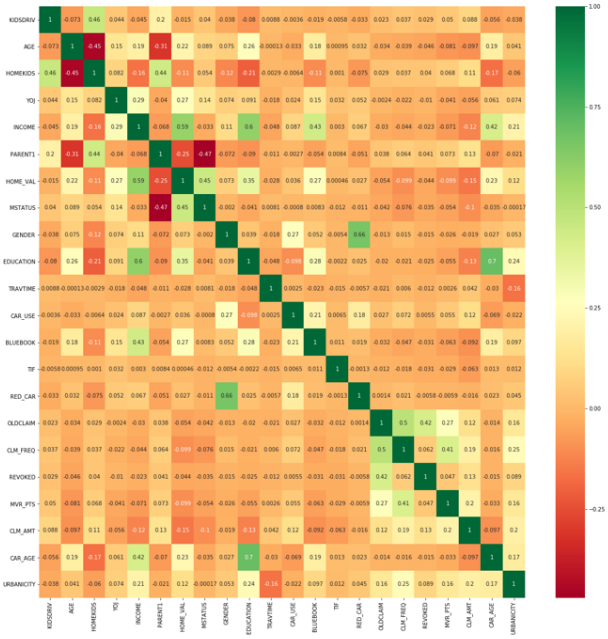
So, we used the histogram to visualize the claim amount.



Now, a few observations are clearly in the 10,000 to 12,0000 range, so we've narrowed it down to zero to 10,000. In addition, the data of observations has been reduced to 8,014 after the removal of outliers.

MSTATUS	GENDER	EDUCATION	...	BLUEBOOK	TIF	RED_CAR	OLDCLAIM	CLM_FREQ	REVOKED	MVR_PTS	CLM_AMT	CAR_AGE	URBANICITY
0.0	1.0	3.0	...	14230.0	11.0	1.0	4461.0	2.0	0.0	3.0	0.0	18.0	1.0
0.0	1.0	0.0	...	14940.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
0.0	1.0	1.0	...	21970.0	1.0	1.0	0.0	0.0	0.0	2.0	0.0	10.0	1.0
1.0	0.0	0.0	...	4010.0	4.0	0.0	38690.0	2.0	0.0	3.0	0.0	10.0	1.0
0.0	0.0	1.0	...	17430.0	1.0	0.0	0.0	0.0	0.0	0.0	2946.0	7.0	1.0

With the basic data processing done, we can use the correlation plot to see the overview of the remaining relationships between variables.



As can be seen in the figure, the claim amount is weakly related to most of the variables. This is because the correlation plot is based on linear regression, with a relatively shallow correlation. In order to further find useful variables, we will use decision tree to do feature selection next.

4.2.Feature Selection

In this section, our goal is to determine the way to pick up the important feature.

4.2.1 Data preparation

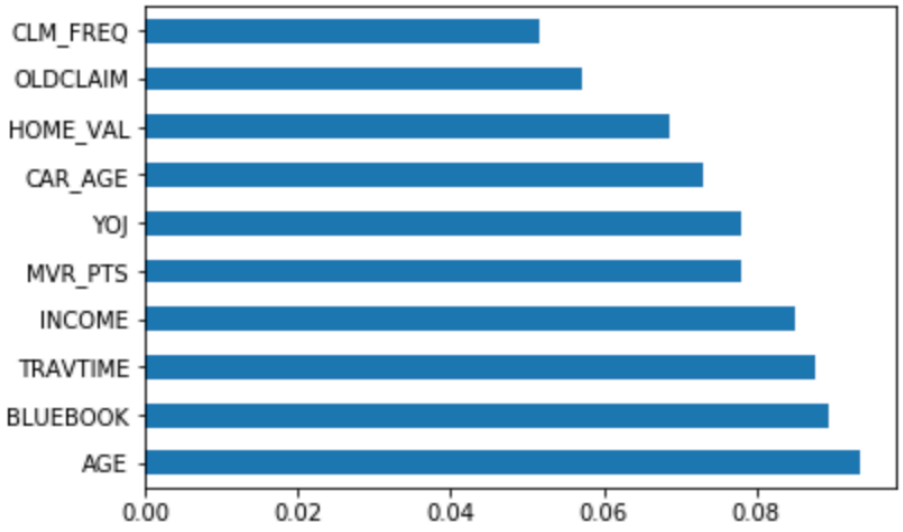
Step 1, Select all the columns (except 'CLM_AMT').

Step 2, Set 'CLM_AMT' as the target column.

4.2.2 Decision Tree

We used a model based on the principle of decision tree to calculate the importance of features and filtered out the top ten, and used bar chart to visualize them.

Feature selection based on decision tree is more suitable for our problem than linear regression. At the same time, we took into account the ease of information collection and independence, and selected the top five characteristics – 'AGE', 'BLUEBOOK', 'TRAVTIME', 'INCOME', 'MVR_PTS' for further research.



4.3. Classification (SVM) and Results

4.3.1 Data preparation

Step1 Replaceing all claim values greater than 1 with 1

In this part, since we tried to make a classification for customers based on the value of claim. We used 1 to replace all the value of claim which were bigger than 1. The new dataset is as following:

6	17430.0	46.0	125301.0	0.0	34.0	1.0
7	8780.0	33.0	18755.0	0.0	54.0	0.0
8	18930.0	21.0	50815.0	2.0	40.0	1.0
9	5900.0	30.0	43486.0	0.0	44.0	0.0
10	16970.0	44.0	107961.0	10.0	37.0	1.0
11	11200.0	34.0	62978.0	0.0	34.0	1.0

Step2 spliting data into training dataset and test dataset

We respectively divided data into training data (70%) and test data (30%).

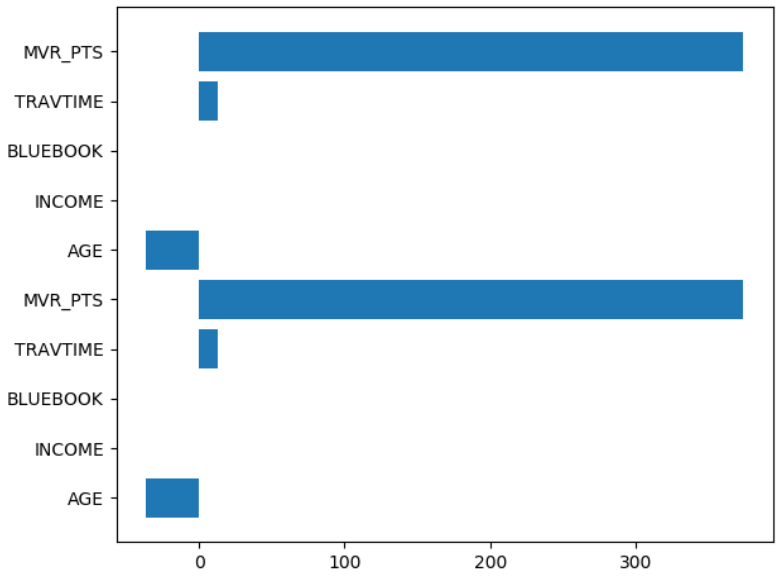
4.3.2 Model Implement (SVM) and Results

In our study, the first goal is whether the claim will be generated in the future after the customer purchases the car insurance. In addition, as a method of supervise learning, we set the target values to be 0 and 1. Where 0 represents the value of claim is 0, 1 represents the value of claim is greater than 1.

For the kernel function, we select linear kernel as our kernel function. The principle of linear kernel is that the output is given by the inner product plus an optional constant c. the formula of the linear kernel is as following:

$$k(x, y) = x^T y + c$$

Since we have selected the kernel function, we can input our training dataset into Python to get the final classification results, which is the value of the claim according to the 5 features.



From the plot, we could get the weight vector:

[[-4.62760907e-02 1.28986284e+01 -1.12976268e-01 3.73541821e+02

-3.62537945e+01]]

4.3.3 Model Evaluation

After we build the SVM model, we used Python to calculate the accuracy of the model and the result is as following:

```
Classification Report:
              precision    recall  f1-score   support

           0       0.76      0.93      0.84      2135
           1       0.44      0.15      0.22       727

   micro avg       0.74      0.74      0.74      2862
   macro avg       0.60      0.54      0.53      2862
weighted avg       0.68      0.74      0.68      2862


Accuracy :  73.51502445842068
```

From the plot above, we can see that the correct percentage of classification 0 is 76%, the precision of classification 1 is 44%, and the final total accuracy is about 73.515%, which indicating the SVM-based classifier model is acceptable.

4.4. Regression (MLP) and Results

After separating potential claimants with SVM, we can predict accurately whether insurance customers will pay claims in the future, but we're not satisfied. We hope to get more concrete results. Therefore, our goal is to find a model that is able to predict the specific amount of customer claims.

4.4.1. Data preparation

In this part, we use the new dataset 'CLM1value.csv' that we split from original data. It contains 2,413 observations with non-null values and 5 customer characteristics - BLUEBOOK, TRATIME, INCOME, MVR_PTS and AGE as variables, CLM_AMT with specific integers as our target.

	BLUEBOOK	TRAVTIME	INCOME	MVR_PTS	AGE	CLM_AMT
6	17430.0	46.0	125301.0	0.0	34.0	2946.0
8	18930.0	21.0	50815.0	2.0	40.0	6477.0
10	16970.0	44.0	107961.0	10.0	37.0	4021.0
11	11200.0	34.0	62978.0	0.0	34.0	2501.0
13	18300.0	15.0	77100.0	0.0	53.0	6077.0

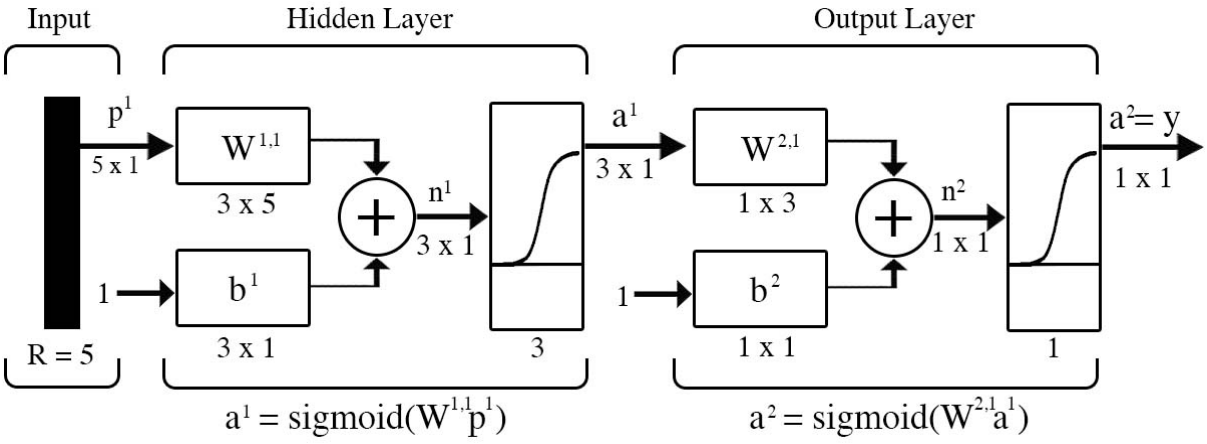
In practice, we stacked by encountered overflow problem. Obviously, a wide range of data makes MLP troubles to operate, so we standardized the data.

	BLUEBOOK	TRAVTIME	INCOME	MVR_PTS	AGE	CLM_AMT
6	0.074086	0.467391	0.000000	0.230769	0.483333	0.291478
8	0.118703	0.260870	0.137686	0.000000	0.200000	0.688553
10	0.161179	0.119565	0.118515	0.000000	0.733333	0.526172
11	0.140435	0.228261	0.097696	0.000000	0.266667	0.449420
13	0.176655	0.163043	0.053070	1.000000	0.400000	0.612204

Similar to SVM, remove missing values. We respectively divided data into training data (70%) and test data (30%) after standardization and removed the missing values. Because there were not too many observations in the original data and some missing values appear in the standardization process, only 403 values of training data can be obtained after these operations, while only 174 values of test data can be obtained.

4.4.2 Network Architecture

For this research, we used the multilayer perceptron (MLP) method to train the dataset and to build the predictive model. Our MLP model included 2 layers which can also call the hidden layer and output layer. To be more specific, the log sigmoid function was used in both of two layers. At the same time, we selected 5 features as our input data. The plot of network architecture is as follows.



$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^{x}}{e^{x} + 1}$$

As we can see from the plot above, when we have training dataset of inputs (feature), we can obtain the outputs after two layers of training.

4.4.3 Process of Building Model and Results

As we have done in the data preparation section, we split data into the training dataset and test dataset. And training dataset was used to produce a predictive model after MLP training. For the initialization of MLP training, we set the number of neurons is equal to 5 and the times of epochs are equal to 1000. After initialization, we put our inputs (training dataset) in the MLP model to obtain the weight matrix to get the predictive model. And the weight vector we obtained from the MLP is as following (neurons = 5, epochs = 1000)

```
Stage 1) Random starting synaptic weights:
    Layer 1 (3 neurons, each with 5 inputs):
[[-0.16595599  0.44064899 -0.99977125 -0.39533485 -0.70648822]
 [-0.81532281 -0.62747958 -0.30887855 -0.20646505  0.07763347]
 [-0.16161097  0.370439   -0.5910955   0.75623487 -0.94522481]
 [ 0.34093502 -0.1653904   0.11737966 -0.71922612 -0.60379702]
 [ 0.60148914  0.93652315 -0.37315164  0.38464523  0.7527783 ]]
    Layer 2 (1 neuron, with 4 inputs):
[[ 0.78921333]
 [-0.82991158]
 [-0.92189043]
 [-0.66033916]
 [ 0.75628501]]
Stage 2) New synaptic weights after training:
    Layer 1 (3 neurons, each with 5 inputs):
[[-3.14944429 -0.10220113 -1.53256736 -1.85949674 -3.23175782]
 [-5.46013835 -2.02573787 -0.84631291 -2.92134375 -4.29557013]
 [-2.81680502  0.55810412  1.90918765 -0.39694533 -3.16541547]
 [-2.57080477 -4.80057779 -1.66770337 -4.49542986 -3.80295988]
 [-5.62322547 -7.56311993 -1.47108119 -6.35276698 -5.70353273]]
    Layer 2 (1 neuron, with 4 inputs):
[[-2.02821766]
 [ 4.03192123]
 [-2.65357125]
 [ 1.79833417]
 [-0.83030681]]
```

According to the weight vector, we can get the predive model with the assistance of the formula:

$$n = f(W.P+b)$$

In this equation, n is neural network output; f is transformed function; W is weight vector; P is neural network input; b is biased in the network; "." means dot product. Therefore, based on the formula above, we can obtain the predictive function:

$$N = \text{sigmoid}(W.P)$$

However, according to the output value of our model, we cannot intuitively see the specific value of the claim that the customers may make in the future because of the standardized data.

To solve this problem, we apply the inverse function in Python to convert the data into its original form. In this way, we can see the specific value of the claim. The final results of predictions are as followings.

Inverse Prediction Values from Standerded to Real

[0.3673214]		[3671.99163825]
[0.42892033]		[4282.74509204]
[0.38475559]		[3844.85163628]
[0.37078258]		[3706.30930536]
[0.32756746]		[3277.8313754]
[0.38619124]		[3859.08617053]
[0.37190939]		[3717.48156009]
[0.44247311]		[4417.12087744]
[0.39065171]		[3903.31168905]
[0.37662705]	→	[3764.25721683]
[0.37076745]		[3706.15926393]
[0.37946452]		[3792.39073447]
[0.3826891]		[3824.36247572]
[0.36139664]		[3613.24770296]
[0.39366315]		[3933.17009324]
[0.37221891]		[3720.55045839]
[0.32236985]		[3226.29706708]
[0.31275087]		[3130.92492173]
[0.43665823]		[4359.46630453]
.		.
.		.
.		.

4.4.4 Evaluation and Improvement of MLP Model

Introduction of the Mean Square Error

The mean square error(MSE), as a popular method of calculating errors, is often used as a standard to evaluate a model. The following is the MSE equation.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2$$

In this function, N is equal to the number of the inputs; e is the errors; t is the target; a is the network output; i is equal to ith input of the dataset.

To define the network performance, MLP is supervised learning which indicates the targets and errors are included in this model. Thus we can use MSE to evaluate the network quality.

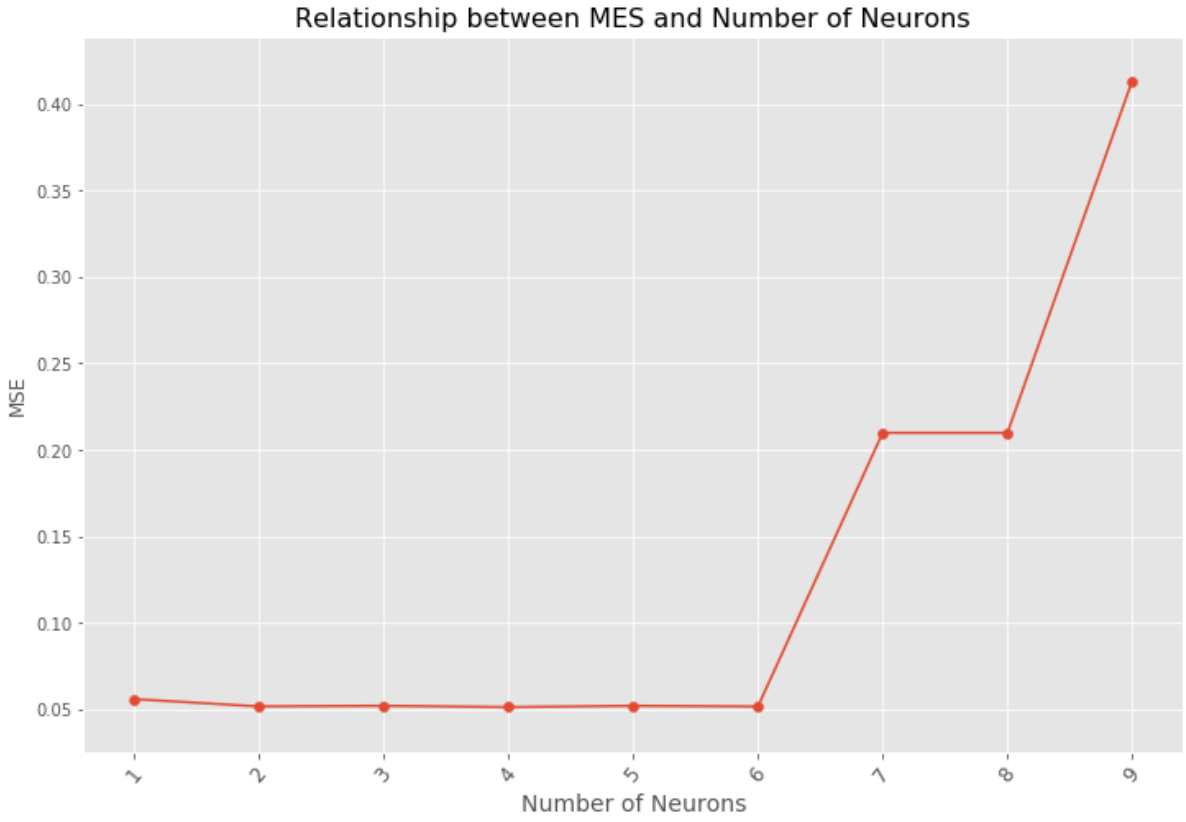
Evaluating the Model with MSE

In order to evaluate the predictive model, we use the test dataset which we have split from the original dataset to calculate the value of MSE. And the value of MSE is equal to 0.3 indicating that our predictive model is acceptable. Thus, our model can be used in the real-life that to assist the insurance company to predict the value of the claim the customers may produce in the future.

Improving the Accurate of the Model

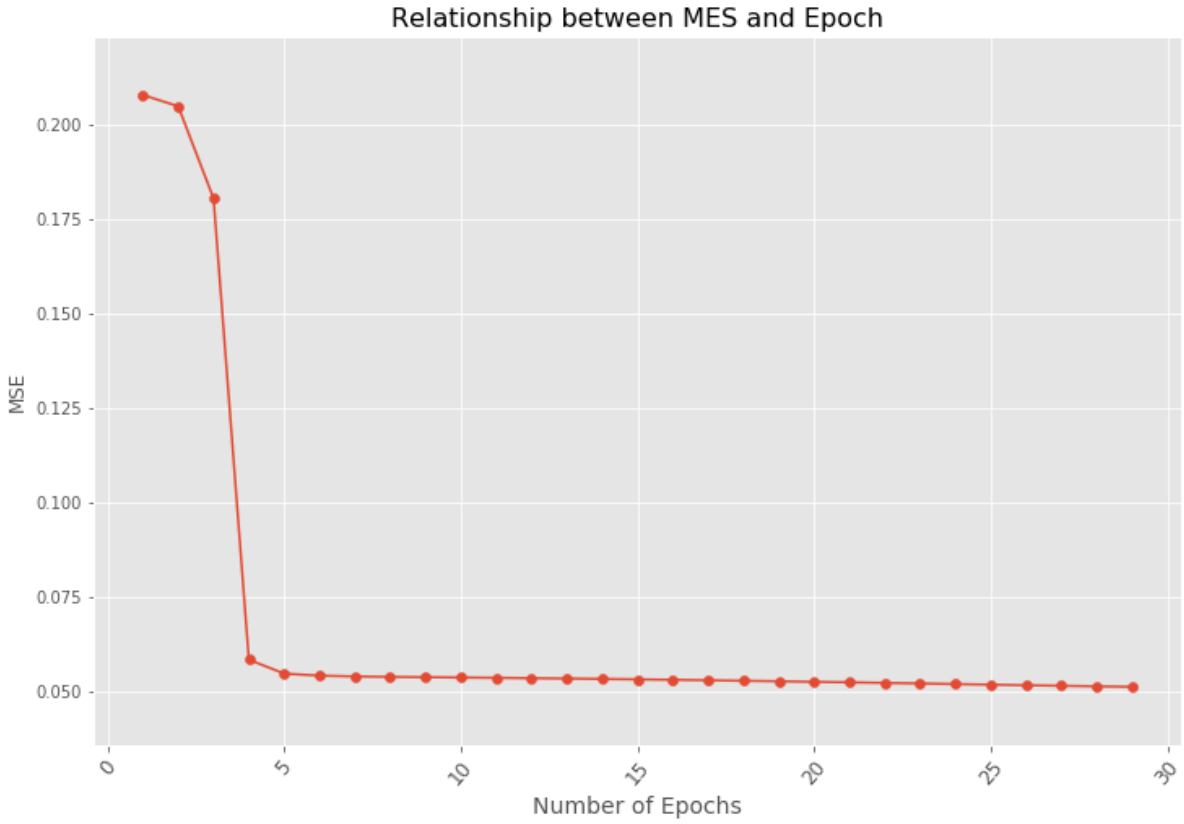
After building the predictive model, we want to adjust the number of neurons and epochs to improve the accuracy of the model. And we can adjust the MSE value to the minimum through the control variable method to enhance the quality of the model.

For the neurons section, we fix the value of epochs is equal to 100. And the relationship between the MSE and neurons is as following.



From this plot, when the value of neurons is greater than 6, an obvious increase in the value of MSE indicates the problem of overfitting. Thus, we selected the number of neurons to minimize the value of MSE. And the ideal number of neurons is 3.

For the epochs section, we fix the value of neurons is equal to 3. And the relationship between the MSE and epochs is as following.



From this plot, as the epoch was greater than 4, the MSE value became minimal and stable. Thus, the optimal value of epoch is equal to 4.

When we established the best values of epoch and neurons, we can re-calculate the MSE values of the model. After calculation, we get that the value of MSE is about 0.03 which is a significant decline compared with before. Thus, the accuracy of the model has been improved in this way.

5. Summary and Improvement

5.1. Summary

For this study, we used SVM method to make a classification which would able to divide customers by whether they will claim after buying car

insurance. At the same time, we applied MLP method to build predictive model in order to predict specific claim amount. Consequently, both MLP and SVM provided satisfying results indicating the feasibility of this research method.

5.2. Improvement of the Model

Due to the huge and complex data volume of users in real life, the accuracy of our prediction model will be reduced to some extent. In this case, we would use K-means algorithm to classify our original data and use the MLP method for each subclass to make regression.

5.2.1 Introduction of K-means Algorithm

Clustering is a kind of unsupervised learning, that is, grouping similar data into the same cluster. The more similar the objects in the cluster are, the better the clustering effect is. In this case, the K-means algorithm is a method of dividing data into K classes without any supervisory signals. And the general steps of the algorithm are as follows:

Step1: Choose K center points at random

Step2: Assign each data point to its nearest central point;

Step3: The average distance from the point in each class to the center of the class is recalculated

Step4: Assign each data to its nearest central point;

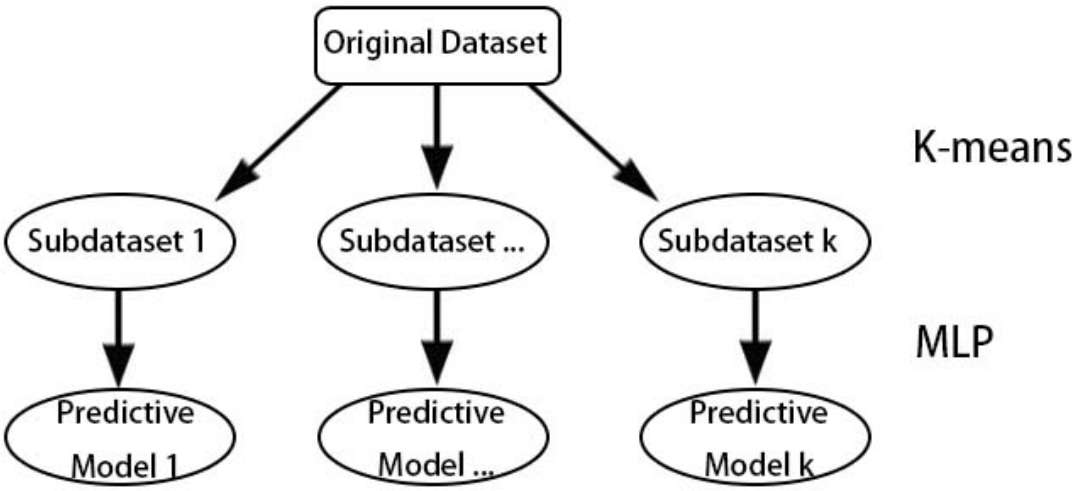
Step5: Repeat steps 3 and 4 until all observations are no longer assigned or the maximum number of iterations is reached.

5.2.2 Application of K-means Algorithm

In our research, we want to use the k-means algorithm to find an optimal classification group number, that is to say, the classification group number that can make the value of MSE become the smallest. In this case, we are breaking down the original data into k classes, and within each of those classes we will re-using MLP to build a predictive model. That brings our total number of predictive models to K.

When we use the established K models to make predictions, we first find the cluster to which the input (customer) belongs, and then use the model to predict the value of claim.

With the assistance of this method, we can minimize the value of MSE and improve the accuracy of the model.



In our research, we did not use K-means algorithm to improve the accuracy of our model because the amount of data was not large, and the final model evaluation was acceptable. However, we believe our approach is feasible. In the future research, when we face with a larger and more complex dataset, we will use K-means algorithm to verify the feasibility of this method.

6. References

Sato, Kaz. "*Using Machine Learning for Insurance Pricing Optimization* | Google Cloud Blog." *Google*, Google Cloud Platform, 19 Mar. 2017, cloud.google.com/blog/big-data/2017/03/using-machine-learning-for-insurance-pricing-optimization.

Malhotra, Ravi, and Swati Sharma. *MACHINE LEARNING IN INSURANCE - Accenture.com*. Accenture, 2018, www.accenture.com/t20180822T093440Z__w__/us-en/_acnmedia/PDF-84/Accenture-Machine-Leaning-Insurance.pdf.

Raschka, Sebastian, and Vahid Mirjalili. *Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and TensorFlow*. Pack Publishing, 2018.

7. Appendix

Columns Explanation

#	Columns	Description	Data type
1	ID	Customers' ID Number	int
2	KIDSDRIV	The number of kids the customers need to drive for.	int
3	BIRTH	Date of birth.	Factor
4	AGE	Age	int
5	HOMEKIDS	The number of kids the customers have.	int
6	YOJ	Working age.	int
7	INCOME	Income.	int
8	PARENT1	Whether the customers' parents pay for their insurance.	Factor
9	HOME_VAL	Property value.	Factor
10	MSTATUS	Marriage status.	Factor
11	GENDER	Gender.	Factor
12	EDUCATION	Education level.	Factor
13	OCCUPATION	Occupation.	Factor
14	TRAVTIME	Travel times.	int
15	CAR_USE	Whether customers' cars are used for private or commercial.	Factor
16	BLUEBOOK	Blue Book is a guidebook that compiles and quotes prices for new and used automobiles and other vehicles of all makes, models and types.	Factor
17	TIF	The insured vehicle.	int
18	CAR_TYPE	Car Type.	Factor
19	RED_CAR	Red cars are more expensive to insure.	Factor
20	OLDCALIM	Cumulative claim amount.	Factor
21	CLM_FREQ	Claim frequency.	int
22	REVOKED	Revoked time.	Factor

23	MVR_PTS	Motor vehicle record points. The Violation/Accident Guidelines and Points Columns on the right are used to assign points to each accident or violation over a three-year period.	int
24	CLM_AMT	Claim amount.	Factor
25	CAR_AGE	Car age.	int
26	CLAIM_FLAG	Claim flag.	int
27	URBANICITY	Car insurance varies according to the area of activity.	Factor