CSE303: Statistics for Data Science [Spring 2021]

Project Report

Submitted by:

Student ID	**Student Name**	**Contribution Percentage**
2018-2-60-007	Al Azim Islam Khan Iram	**37**
2018-2-60-016	Nishi	**26**
2018-2-60-028	Rakibul Huda	**37**
2018-2-60-030	Md. Shamsuzzaman Bhuiyan	**Not found**

1. Introduction

In this project we just tried to build and compare those models from given datasets. We had been given two datasets where one has independent and dependent columns while other has only independent columns.

At first we read the given train and test datasets. Then we checked if there is any unnecessary column which has missing or null value, has string value in the place of integer or has integer value in the place of string. There was not any missing or null value but we had to cancel out the 'id' column as it has no contribution on 'target' or dependent column.

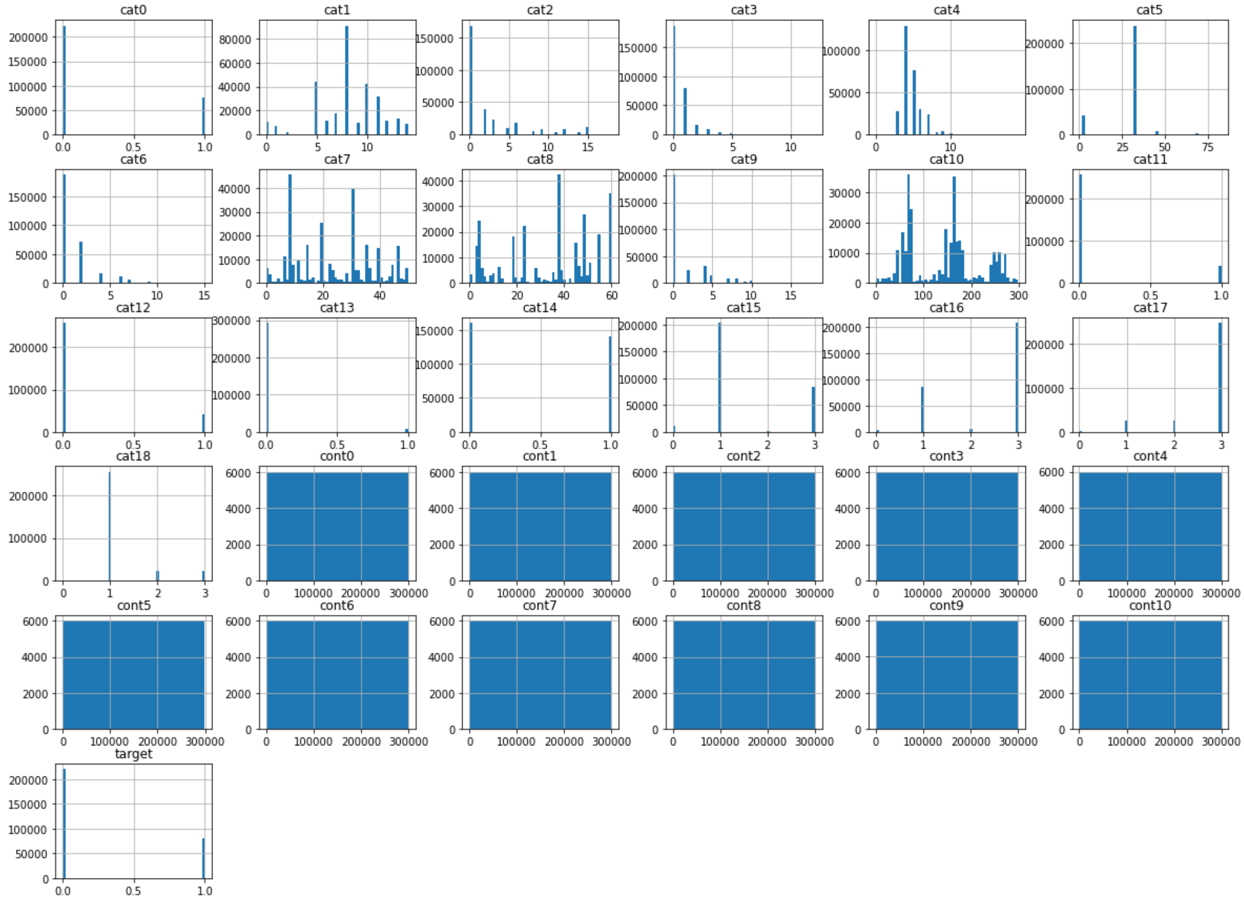
Later we changed all the columns which have datatype as float64 to int64. In this case the same data has been replaced with the same value of datatype int64 and thus for different data we had different values.

Secondly we divided the data for X and Y in terms of features and target respectively but we did not have to do this for test dataset as there are only the features available. In the dataset among the features, there are big difference for which had to scale down the difference. For principal component analysis we set the parameter to 0.95 by which it holds 95% of the variance and required number of components for that.

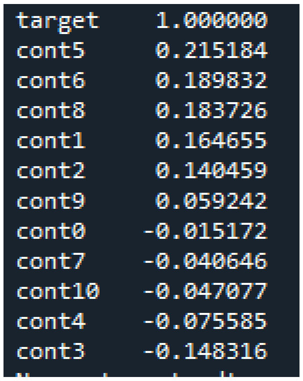
Finally we used two models which are Logistic Regression, Support Vector Classifier and Logistic Regression (tuned). The accuracy scores of these two models are 0.8367, 0.8366 and 0.8368 for LR, SVC and LR (tuned) respectively.

2. Exploratory Data Analysis

In this dataset there are 32 columns and around 300000 rows. If we dug deep into this dataset we find that there is column named Id and many categorical variables. Id is not important here. Therefore we will remove the Id column. Let us visualize the histogram of above dataset.



From this histogram we can see the data distribution of all the column, how their data has been distributed. Now let's see the correlation with target variable to all other columns.


```
target     1.000000
cont5      0.215184
cont6      0.189832
cont8      0.183726
cont1      0.164655
cont2      0.140459
cont9      0.059242
cont0     -0.015172
cont7     -0.040646
cont10    -0.047077
cont4     -0.075585
cont3     -0.148316
```

From here we can see that cont5 has the highest positive correlation with the target column. This correlation is not very high but it is the heist positive of all correlations. This indicates that cont5 has the huge effect on the target variable. On the other hand cont3 has the lowest correlation among all that indicates cont3 has the lowest effect of the target variable.

3. Machine Learning Models

Logistic Regression: In this algorithms we try to build a model where our model will be able to predict the result as accurate as it can. **The result can have two or more options** like 0 and 1 or positive, neutral and negative etc. There will be several features which help us to find the result in terms of our model. It can be used in binary, multinomial and ordinal. The binary model divides the two classes with a S-shaped curve. The result is being predicted from an equation like p(y) = 1 / (1+$e^{-(y)}$) where we find the probability and based on this probability we decide the result or value of y. If the probability gets equal or higher than 0.5 then the result will fall in one class and vice versa. Basically from the hypothesis we are getting two values which are the probability and the value of y. To find the cost function of Logistic Regression we cannot use the cost function of Linear Regression as it has multiple local minima and in partial derivation we can get stuck into one local minima and will not be able to move another local minima. In cost function we get only two types of values as either 0 or infinity. If our actual value is equal to model value then cost function is 0 or it will be infinity.

Support Vector Machine: In this algorithm we can work with two or more classes or categories. Let's discuss about only two classes. We tried to divide the two classes in a way where there will be a hyperplane between them. If there are more than two classes, there will be multiple hyperplane. We place this hyperplane by using orthogonal projection which helps to draw the hyperplane as middle as possible from the closest support vectors of two classes. The closest data point of the hyperplane are supposed to be support vector. Our goal is to maximize the width of that area (the parallel area of the hyperplane) so that we can predict the result with more confidence. To predict the result we must find the predicted value in either side of the area but not in between.

4. Data Preprocessing

We did the following things for data pre-processing:

- There was not any null values in datasets
- Encoded all the columns which had datatype float64 to turn them into int64
- Used PCA to reduce the dimension of the data by setting the variance

5. Different Models

Hyper Parameter Tuning of Logistic Regression: Here we tried to tune hyper parameter of the base logistic regression. It consists two type of regularization l1 and l2. We have tuned to l2 regularization as we used the “newton-cg” solver. We kept the maximum number of iteration to 100. We have tuned this max_iter value several time but we didn’t get any decent upgrade except downgrading. As it is binary problem that’s why we have chosen “ovr”. Here we have tuned the value of c as the smaller values of c gives the stronger regularization. We tuned c to 0.001 which slightly increases the performance. As we have processed the data in such a way so that the default model can easily give the perfect result. For that reason we could come up with a slight increasing in the accuracy.

C value	0.001
multiclass	Ovr
solver	newton-cg
penalty	l2

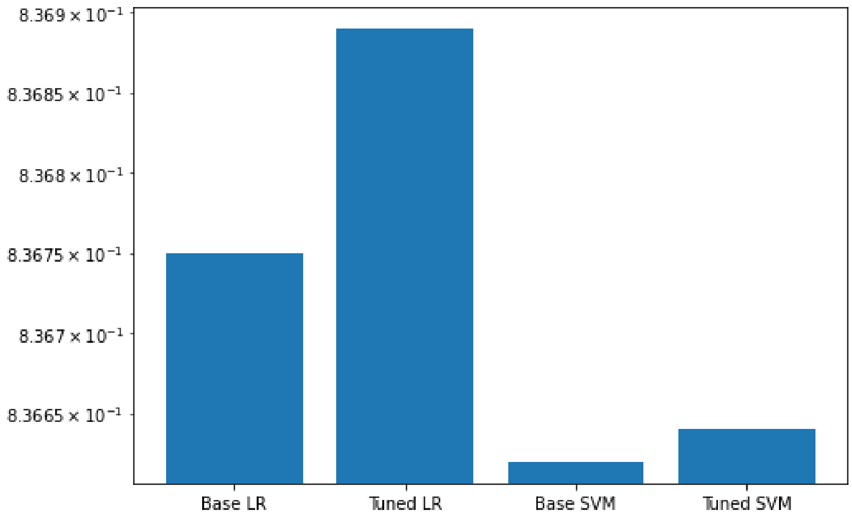
Hyper Parameter Tuning of Support Vector machine: We have used the linear svc with a c value of 1 as a base model. Then we tried to tune the model to get a more accurate result. We have used the grid search method for finding the perfect combination of hyper parameter tuning but we got the same accuracy as the base model gave. We have tuned the max_iter value, c value, penalty, loss and dual to get a little change of the accuracy. Here loss indicates the loss function, penalty is the regularization method, max_iter is the number of iteration and c is the strength of regularization. We have used c=0.6, penalty='l2', loss='squared_hinge', dual=False, max_iter=10000. We didn’t get much higher accuracy than the base model is might be that, we have we have processed the data perfectly before applying the base model and the tuned model.

C value	0.6
penalty	l2
loss	squared_hinge
max_iter	10000
dual	False

6. Performance Evaluation

Base Logistic Regression output: 0.83675
Tuned Logistic Regression output: 0.83689
Base SVM output: 0.83662

Tuned Base SVM output: 0.83664



From this bar plot we can visualize our model. Our Difference is so little that we took y_scale range to “log” scale. From the Bar plot we can conclude that we will get better rest using our tuned Logistic Regression Model.

7. Discussion

Among all of our model the accuracy scores are shown below as

Tuned Logistic Regression > Logistic Regression > Support Vector Machine

Here Tuned LR works better as we tried many combination of the parameters. In our model basically it is LR which performed slightly better than SVM. We expected to perform more efficiently for SVM as it is better to reduce the risk of error and also less probability for overfitting model.

8. Conclusion

We are not satisfied with our work because we think we could not utilize the Hyper Parameter Tuning properly as there is not very much difference we could make. The datasets are also inefficient to real life use as we cannot understand what every features actually mean or what these datasets are all about.