Compare Support Vector Machines to a 3 layer Neural Networks on the Titanic dataset.

Akash Bisht

Bangalore,India

akashbisht81@gmail.com

Machine Learning Intern, AITs

ai-techsystems.com

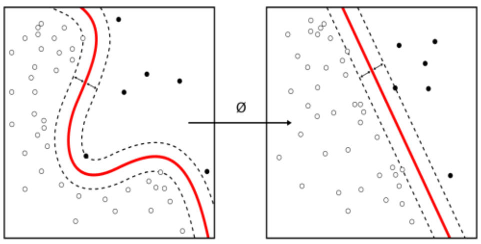
***Abstract*—This project reports the performance of two different algorithms namely Support Vector Machine and a Three- Layer Neural networks on dataset. It observes the accuracy between two of these. Later on, we see some changes made, and what is the final conclusion. The dataset is about the people who onboarded titanic and their likeness of survival. Although, we can predict and store the final predictions, but our focus will be only on comparative study between two of these.**

We see the combat between these two models relatively one - over another in some conditions.

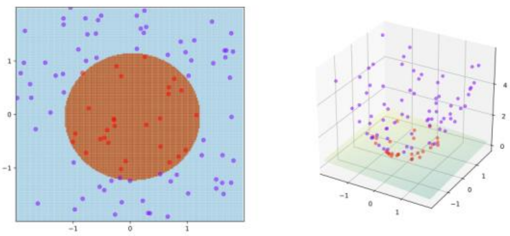
I. Introduction

We present the following case of comparative study of two different algorithms to evaluate the performance based on accuracy metric.

The first model to be used is Support Vector Machine. Support Vector Machine are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects belong either to class A or class B. SVM supports both regression and classification tasks and can handle multiple continuous and categorical variables. Its uses the concept of **"maximizing the margin between two different classes".**



To learn complex non-linear functions, SVM makes use of kernel methods, which enable them to operate in a high-dimensional, implicit feature space without ever computing the coordinates of the data in that space, but rather by simply computing the inner products between the all pairs of data in feature space. This operation is computationally cheaper than the explicit computation of the coordinates. This is also called the **"kernel trick"** as shown in the figure below.

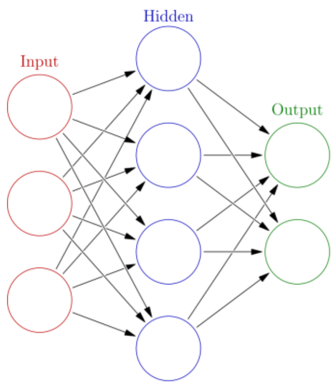


The second model to be used is of neural networks, but use three-layer neural network in this case. Neural network models can be viewed s simple mathematical models defining a function

$$f : X \to Y$$

Mathematically, a neuron's network function f(x) is defined as a composition of other functions g(x) that can further be decomposed into other functions. This can be conveniently represented as a network structure, with arrows depicting the dependencies between functions. A widely used type of composition is the nonlinear weighted sum, where (commonly referred to as the activation function) is some predefined function, such as the hyperbolic tangent, sigmoid function, SoftMax function, or rectifier function. The important characteristic of the activation function is that it provides a smooth transition as input values change, i.e. a small change in input produces a small change in output.

$$f(x) = K\left(\sum_i w_i g_i(x)\right)$$



II. DATASET AND PRE-PROCESSING

We take our first step through the raw data, which is **“not ready to use”** in order to fit in our models. It can be downloaded from https://www.kaggle.com/c/titanic/data.

This Dataset comprises of three different files, but we only consider two files for our comparative study on this dataset.

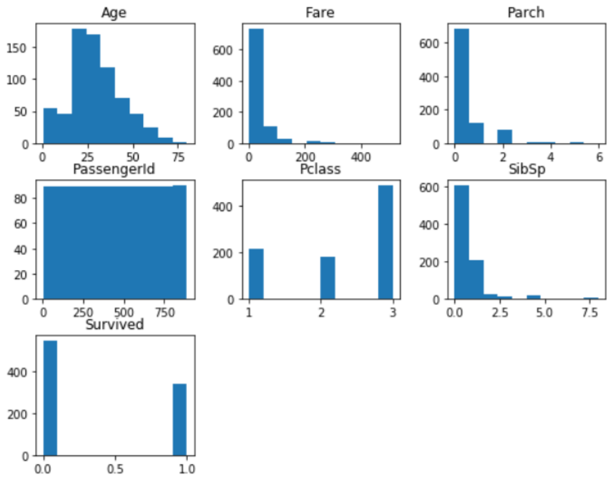
We take our dataset into consideration and split it across on average of between 77 – 23 for train and test set. We then perform operations on this dataset.

Data pre-processing is the first step to clean and pre-process in order to be fit into our model. To perform such cleaning process, we do the following: -

First we check the missing or null values in the Data Frame, using **“isnull.sum()”** method. By Doing that, we get the following results: -

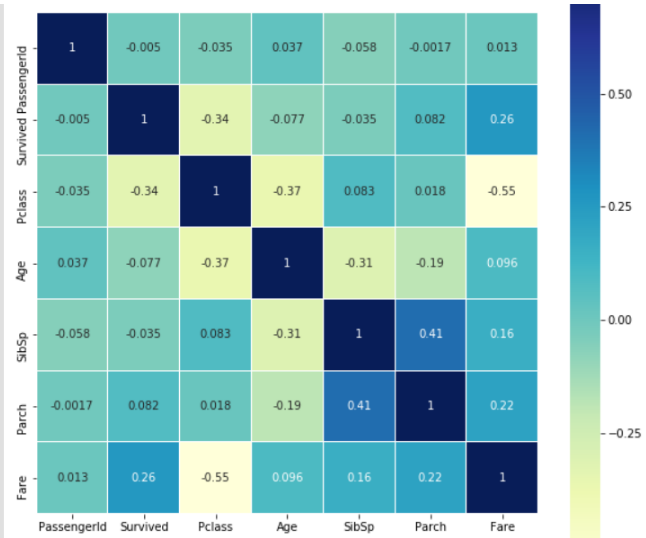
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

We see that “Age”,”Embarked”, and “Cabin” has missing values.Also we plot histogram to check the “Skewness of the dataset” and to fix it after that.



We can see that the distributions of Age and Fare Variable is not proper in format. We can also use log-transformation for that.

We can also visualize the correlation heat map to better understand the relationships. —



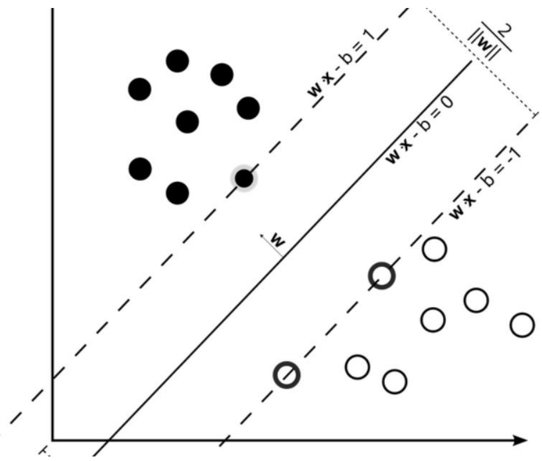
Description

We pre-process the data after analyzing the visualization, and convert it to all numerical-valued feature, and the result we got it is: -

	PassengerId	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Lname	NamePrefix
0	892	3	1	7	0	0	0	7	401	19
1	893	3	0	0	1	0	0	7	843	20
2	894	2	1	3	0	0	1	7	552	19
3	895	3	1	7	0	0	1	7	851	19
4	896	3	0	4	1	1	1	7	342	20

Support Vector Machine vs Neural-Network

The formulation of SVM has been posed and derived absolutely on the basis of algebraic/statistical techniques by Vladimir Vapnik. What it tries to do is construct a linear separating hyper-plane in N-dimensional space to separate all the classes with as large-margin as possible.



Neural Networks are known to be quite sophisticated models in the ML community. Without directly finding out a separator margin between classes, it tries to map the inputs to another higher/lower dimensional space where the inputs can be separated by a linear classifier if they are not linear in nature.

There are many applications where they're better, many applications where they are worse. It also depends on who you ask. It is hard to say this type of data or that type of application. An example where ANN, in particular CNN, work better than SVMs would be on MNIST dataset.

On Classification datasets – first results in short: svms learn fast and predict slow – neural networks learn slow but predict fast and have very lightweight models. **Concerning accuracy, both methods seem to be on par.**

"In comparison with traditional multilayer perceptron neural networks that suffer from the existence of multiple local minima solutions, convexity is an important and interesting property of nonlinear SVM classifiers. [more]"
Suykens et al. (2002).

III. RESULTS

In **Support Vector Machine,** we use the sklearn's SVC library, and find the best estimator values from the set of parameters and then perform Grid Search on it. We use the following parameter values: -

```
parameters = {'kernel':('linear', 'rbf'), 'C':(1,0.25,0.5,0.75),'gamma': (1,2,3,'auto')
```

By the end of training, we get our Final-accuracy values as total of 79.80% and F1-score value of 77.96%.

For, **Neural-Network,** we make use of keras library that uses TensorFlow Back-end.
Our Architecture for our **Simple Keras Model** is as follows:

PART I NN ARCHITECTURE: -

LAYER ONE— DENSE, RELU (128)
LAYER TWO— DENSE, RELU (32)
LAYER THREE— DENSE, RELU (16)
LAYER LAST— DENSE, SIGMOID (1)

We use **ADAM** optimizer, with the beta values of 0.9 and 0.999, our **LOSS as binary cross-entropy,** and we train over **100 epochs.** We get the following accuracy metrics.

```
acc: 76.263%
```

This table summarizes as follows for our comparative study between these two models.

.**TABLE I:-**

Metric	Support Vector M/c	Neural-Network
Accuracy Score	0.7980	0.7621

This Clearly shows that Support Vector Machines Performs relatively best.

But, To Out-perform the SVM, The NN-Model Architecture would undergo some changes.

PART II- NN ARCHITECTURE: -

DENSE, RELU (128)
(ADDED) BATCH NORM and DROPOUT (0.5)
DENSE, RELU (32)
(ADDED) BATCH NORM and DROPOUT (0.5)
DENSE (16)**, (ADDED) DROPOUT (0.3)**
DENSE, SIGMOID (1)

Now, we trained this model with batch-size of 32 and over 150 epochs, and surprisingly we get over 82.047% of accuracy.

This table summarizes as follows for our comparative study between these two models.: -

TABLE II: -

Increased the number of epochs training and added extra dropout and Batch norm layers to compete with SVMs, i.e., in order to increase the performance.

Metric	Support Vector M/c	Neural-Network
Accuracy Score	0.7980	0.8204

Therefore, we can conclude that SVM Performs Best relatively with 3-NN Architecture with a High-Score of Architecture.
But with the Increase no of additional Layers, Neural Networks can outperform SVM in this case.

IV. REFERENCES

[1] Evgeny Byvatov, Uli Fechner, Jens Sadowski and Gisbert Scheider "Comparison of Support Vector Machine and Artificial Neural Network Systems for Drug/ NonDrug Classification.

[2] Mohsen Behzad, Keyvan Asghari, Morteza Eazi, Maziar Palhang, "Generalization performance of support vector machines and neural networks in runoff modeling.

[3] Zan Huang, Hsinchun Chen, Chia-Jung Hsu, Wun-Hwa Chen, Soushan Wu, "Credit rating analysis with support vector machines and neural networks: a market comparative study.

[4] https://stats.stackexchange.com/questions/30042/neural-networks-vs-support-vector-machines-are-the-second-definitely-superior

[5] https://web.archive.org/web/20120304030602/http://indiji.com/svm-vs-nn.html

[6] http://www.statsoft.com/textbook/support-vector-machines

[7] https://stackoverflow.com/questions/8326485/ann-and-svm-classification

[8] KSDurgesh,BLekha http://jatit.org/volumes/twelfth_volume_1_2010.php