Compare Support Vector Machines to a 3 Layer Neural Networks on Titanic Dataset

Mr.Omkar Jadhav
Machine Learning Intern
AI Technology & Systems
https://www.ai-techsystems.com
Mumbai,India
jadhavomkar85@gmail.com

Mr. Omkar Masurekar
Machine Learning Intern
AI Technology & Systems
https://www.ai- techsystems.com
Mumbai,India
omkarmasurekar18@gmail.com

Abstract – The RMS Titanic was a British passenger liner that sank in the North Atlantic Ocean in the early morning hours of 15 April 1912, after it collided with an iceberg during its maiden voyage from Southampton to New York City. There were an estimated 2,224 passengers and crew aboard the ship, and more than 1,500 died, making it one of the deadliest commercial peacetime maritime disasters in modern history. The RMS Titanic was the largest ship afloat at the time it entered service and was the second of three Olympic-class ocean liners operated by the White Star Line. The Titanic was built by the Harland and Wolff shipyard in Belfast. Thomas Andrews, her architect, died in the disaster. Here, we will use two different Machine Learning models on the famous Titanic dataset and compare them.

I. INTRODUCTION

The sinking of the RMS Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. This sensational tragedy shocked the international community and led to better safety regulations for ships. One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew. Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class. This data is available on Kaggle.com.[1][6] ML is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead.

It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task.

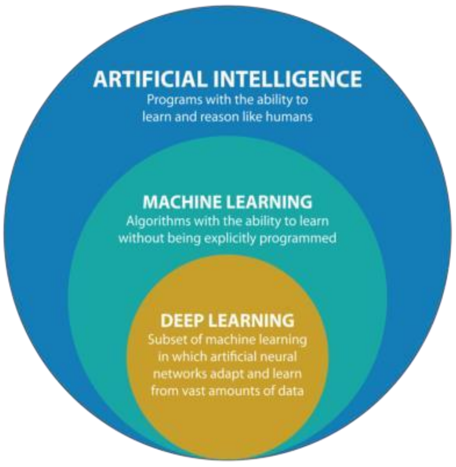

ARTIFICIAL INTELLIGENCE
Programs with the ability to learn and reason like humans
MACHINE LEARNING
Algorithms with the ability to learn without being explicitly programmed
DEEP LEARNING
Subset of machine learning in which artificial neural networks adapt and learn from vast amounts of data

Fig 1: Hierarchical structure of AI [2]

Here our key objective is to apply two different ML algorithms named SVM (Support Vector Machines) and 3-layer Artificial Neural Network (ANN) and compare their accuracy in percentage on the dataset along with exploring various aspects and characteristics of the dataset..

II. DATASET

The dataset we use for our paper was provided by the Kaggle.com website. It is a free open source platform to all. The data has been split into two groups: training set (train.csv) and test set (test.csv). The training set is used to train our models i.e. to make our model learn by finding various hidden patterns in the dataset. The test set should be used to see how well the model performs on unseen data based on the patterns found in the training dataset. The data consists of 891 rows in the train set which is a passenger sample with their associated labels. The data is in the form of a CSV (Comma Separated Value) file. For the test data, we were given a sample of 418 passengers in the same CSV format. The Description of Dataset fields in titanic is listed below:

Description of Dataset fields in titanic:
PassengerId: Id of every passenger
Survived: 1 if the person has survived and 0 if not
Pclass: The class the passenger belonged(1,2 or 3)
Name: Name of the person
Sex: Gender of the person.
Age: No. of year person has lived.
SibSp: Number of Siblings/Spouses Aboard
Parch: Number of Parents/Children Aboard
Ticket: Ticket no.

Fare: Cost of travel.
Cabin: Cabin no. allocated to passenger.
Embarked: Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)

III. DATA EXPLORATION

Before we begin with the creation of actual models, exploration of data is equally important. So, we need to analyze the data to take necessary steps ahead. We have done this by using the help of matplotlib and seaborn to do some graphical analysis. The results are shown below:

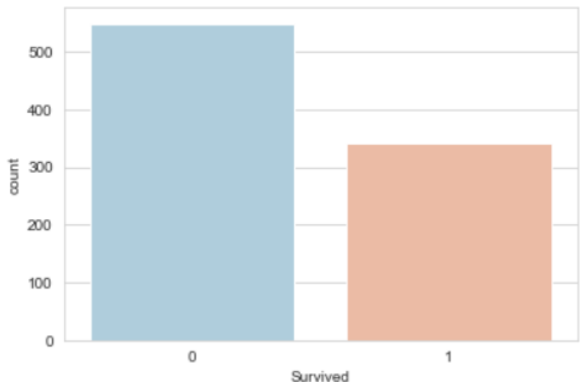


Chart 1: Total count of survival

As we can see from above chart, the death count is more than the survival count.

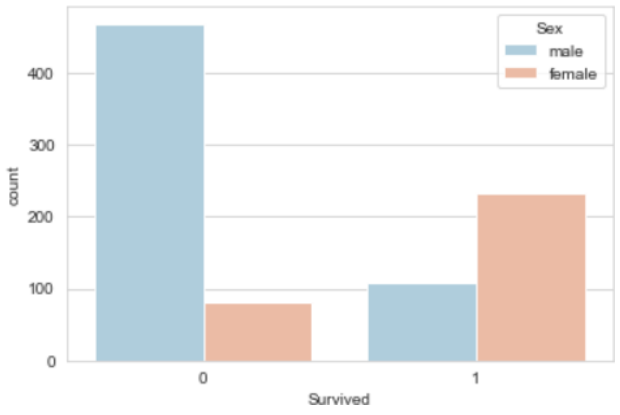


Chart 2: Sex wise classification of survival

Moving ahead, we further classify the previous result into male and female classes.

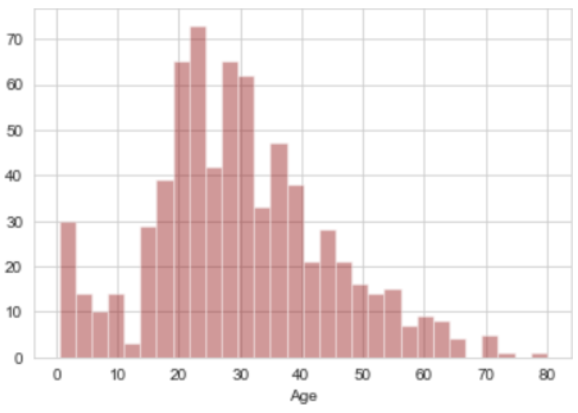


Chart 3: Analyzing the age distribution of passengers.

From the above graph, we come to know that the maximum people aboard are from the age group of 20 to 40

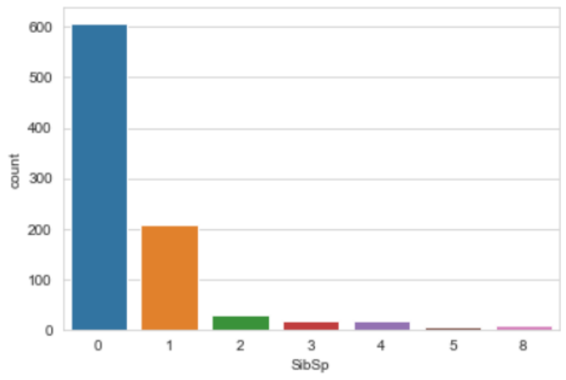


Chart 4: No. of people having Spouse or siblings

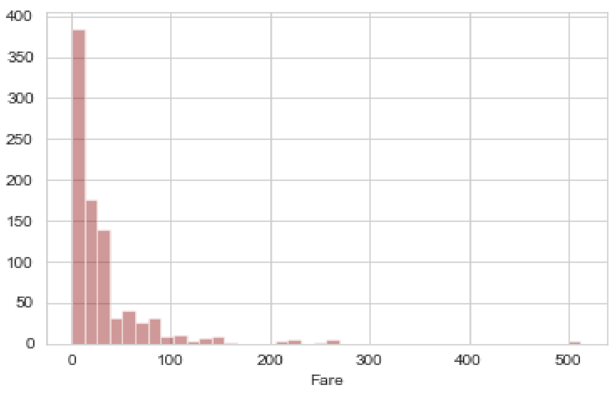


Chart 5: Fare distibution.

The above graph gives us inforamation about the distribution of Fares present at that time & chart 4 shows us the number of people having spouses or siblings along with them.

IV. DATA PREPROCESSING

After exploring the given data by using statistical techniques, we move forward to the next step i.e data preprocessing. In this step, the main aim is to deal with the missing data and scaling of values.

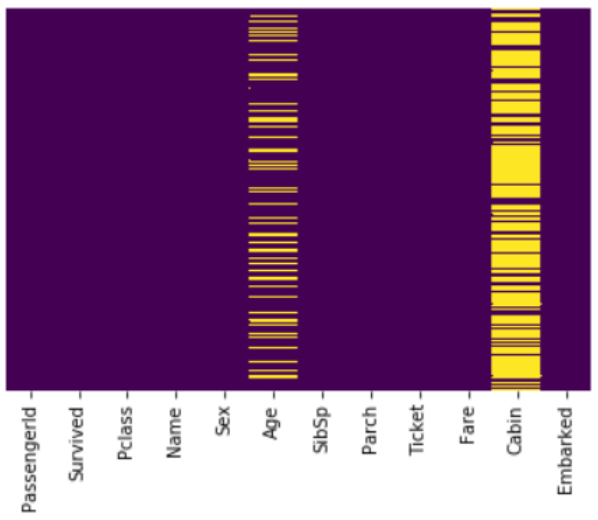


Fig 2:Missing data present before preprocessing

Using heatmap, we can see the null values present in each attributes, denoted by yellow strips. The 'Cabin' attribute contains too many missing values and hence is not of much importance to us. So we drop that column, whereas the 'Age' attribute have few missing values which can be dealt with.

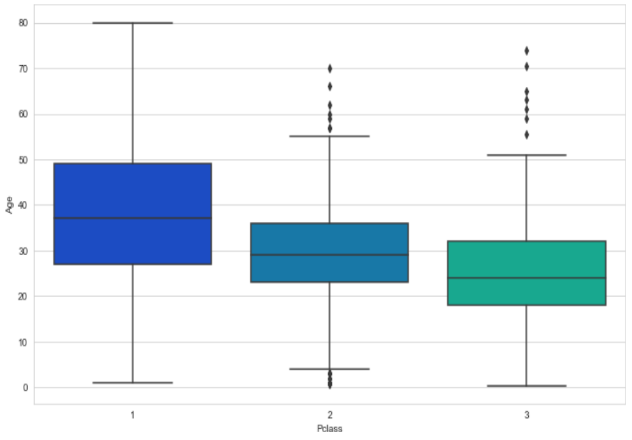
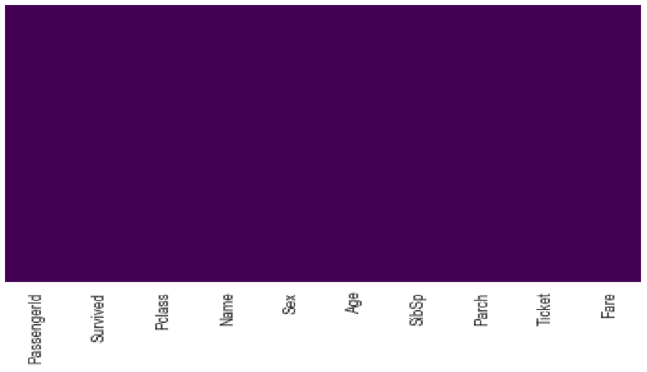


Fig 3: Mean values of ages for every type of 'Pclass'

We can see that the wealthiest passengers in the higher classes tends to be older, which makes sense. We will use these average values to impute based on 'Pclass' for age. After putting corresponding mean values for the 'Age' column, we get a complete dataset with no null values present in it.



After dealing with the null values, we need to convert character values into some numerical values as character values cannot be used for training our model.

The attributes containing character values in our dataset are 'Embarked' and 'Sex'.These are converted into corresponding numerical values using get_dummies() of pandas. 'Embarked' column is converted into three separate columns 'P', 'Q' and 'S' each having 0 or 1 values and the 'Sex' column is converted into 'Male' and 'Female' columns each having 0 or 1 values.

After conversion we get rid of some attributes which are of no use while training the data. These columns are 'PassengerId', 'Name', 'Ticket' and the old columns 'Embarked' and 'Sex'.

Some data needs to be scaled, this is done using StandardScalar( ) of sklearn. This step needs to be done as it converts every value of the dataset in a common range. This improves the performance of the machine learning model.

After this step, a clean dataset is obtained on which Machine Learning models can be easily implemented.

Following are the Machine Learning models that we have implemented:

V. Algorithms Implemented

A. *Artificial Neural Network (ANN):*

An artificial neuron network (ANN) is a computational model based on the structure and functions of biological neural networks. Information that flows through the network affects the structure of the ANN because a neural network changes - or learns, in a sense - based on that input and output.

ANNs are considered nonlinear statistical data modeling tools where the complex relationships between inputs and outputs are modeled or patterns are found[3].

A neural network consists of:

1. Input layers: Layers that take inputs based on existing data. Here we have 7 input variables.
2. Hidden layers: Layers that use back propagation to optimize the weights of the input variables in order to improve the predictive power of the model. Here we are using 1 hidden layer with 4 hidden units (neuron).
3. Output layers: Output of predictions based on the data from the input and hidden layers.

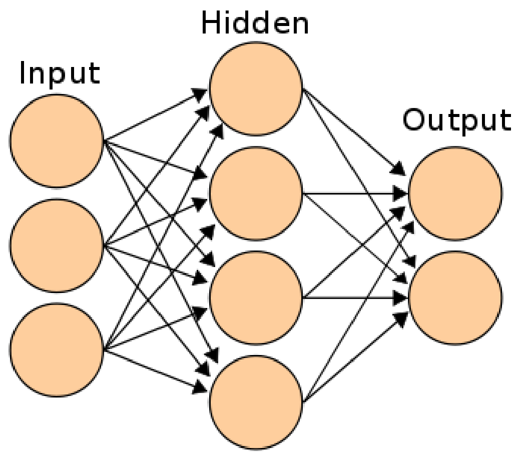


Fig 5: Simple ANN

If we zoom in one of the hidden or output node we will encounter the following figure:

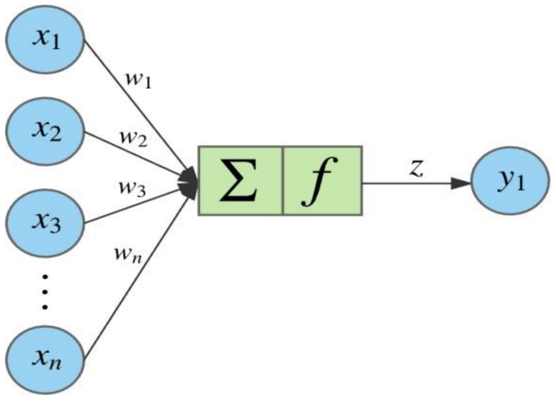


Fig 6: Internal structure of node

A given node takes the weighted sum of its inputs, and passes it through a non-linear activation function. This is the output of the node, which then becomes the input of another node in the next layer. The signal flows from left to right, and the final output is calculated by performing this procedure for all the nodes. Training this deep neural network means learning the weights associated with all the edges.

The equation for a given node looks as follows. The weighted sum of its inputs passed through a non-linear activation function. It can be represented as a vector dot product, where n is the number of inputs for the node.

$$z = f(x \cdot w) = f\left(\sum_{i=1}^{n} x_i w_i\right)$$

$$x \in d_{1\times n},\ w \in d_{n\times 1},\ z \in d_{1\times 1}$$

B. *Support Vector Machine (SVM)*

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.

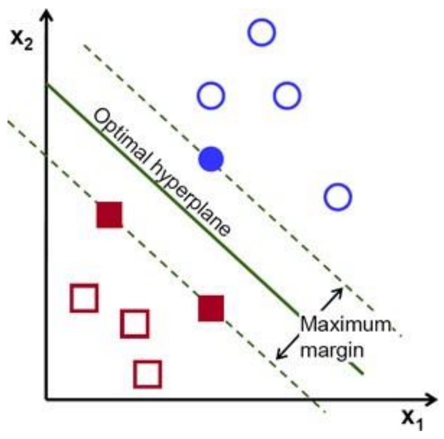


Fig 7 : Hyperplanes in SVM

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.

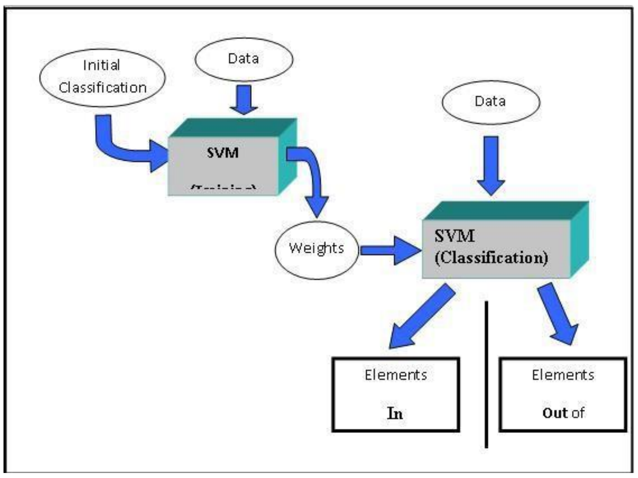


Fig 8: SVM Process Overview[3]

The above figure is used to define the basic SVM mode. The Train and Classify option allows one to run both phases of the algorithm. Starting with a presumptive classification and expression data the result is a final classification of each element. The Train only option produces a list of weights which can be stored as an 'SVM' file along with training parameters so that they can be applied to data to classify at a later time. The Classify only option prompts the user for an SVM file of weights and parameters and results in final classification. The user also has an option to produce hierarchical trees on the two resulting sets of elements.

VII. Performance Of Algorithms

The parameters used for testing the completeness of model are accuracy and f1 score. Accuracy is the ratio of number of instances correctly predicted to that of total number of cases whereas f1 score depends on the precision and recall and can be calculated based on confusion matrix.

A. Confusion Matrix:

A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made.

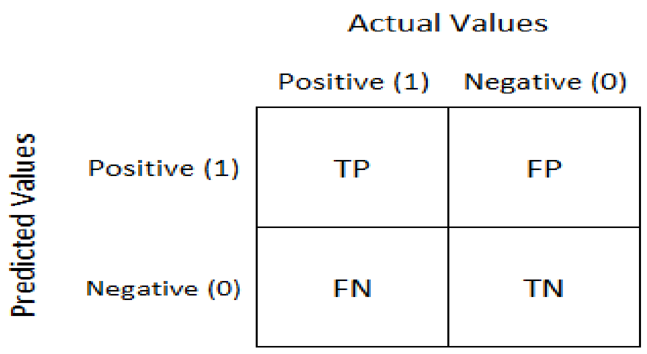


Fig. 9: Confusion matrix

TP = True Positive (Actual Positive which have been predicted as positive)
TN = True Negative (Actual Negative which have been predicted as Negative)
FP = False Positive (Actual Negative which have been predicted as positive)
FN = False Negative (Actual Positive which have been predicted as negative)
Recall is the ratio of Correctly predicted positive values to that of total number of positive values present in data. Mathematically:

Recall = TP / (TP + FN)

Precision is the ratio of total number of correctly predicted positive values to that of total number of predicted positive values

F1 score is the weighted average of Precision and Recall Mathematically:

F1 Score =(2*Precision*Recall) / (Precision + Recall)

Classification Rate or Accuracy is given by the relation:
Accuracy = (□□. □□ □□□□□□□ □□□□□□□□□□□) / □□□□□ □□ □□ □□□□□□□□□□□ □□□□

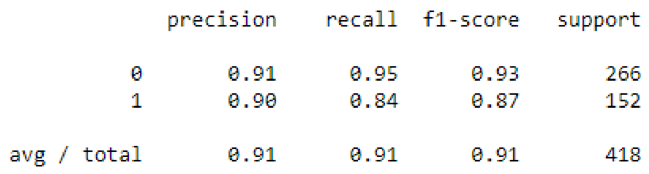
VIII. RESULTS

A. Observed Confusion Matrix on both algorithms:

ANN TEST			
	0	1	Total
0	252	14	266
1	25	127	152
Total	277	143	418

SVM TEST			
	0	1	Total
0	255	11	266
1	10	142	152
Total	265	153	418

B. Classification Report

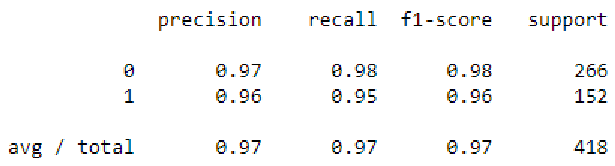


```
             precision    recall  f1-score   support

          0       0.91      0.95      0.93       266
          1       0.90      0.84      0.87       152

avg / total       0.91      0.91      0.91       418
```

Fig. 10: Precision & F1 Score using ANN[5]



```
             precision    recall  f1-score   support

          0       0.97      0.98      0.98       266
          1       0.96      0.95      0.96       152

avg / total       0.97      0.97      0.97       418
```

Fig. 11: Precision & F1 Score using SVM

C. Result:

	Accuracies in Percentage(%)	
	ANN	SVM
TRAIN DATA	73	95
TEST DATA	91	97

IX. Conclusion

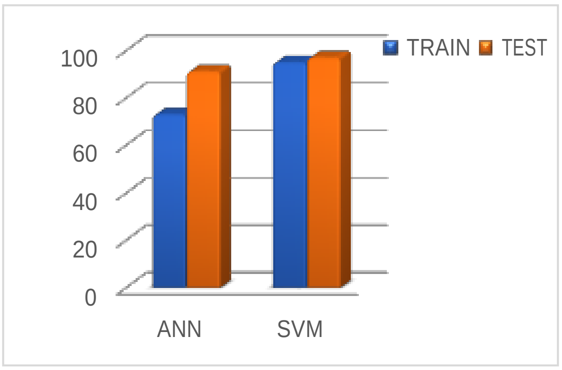


Fig. 12: Comparison of accuracy of ANN and SVM

The above graph shows that SVM performed well than 3-Layerd ANN on both, train as well as test data.

References

1. https://www.kaggle.com/c/titanic/overview
2. https://www.qubole.com/blog/deep-learning-the-latest-trend-in-ai-and-ml/
3. https://towardsdatascience.com/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6
4. http://home.cc.umanitoba.ca/~psgendb/birchhomedir/doc/MeV/manual/svm.html
5. https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9
6. https://www.kaggle.com/ravaliraj/titanic-datavisualization-and-ml