Compare Support Vector Machines to 3 layer Neural Networks on the Titanic Dataset (August,2019)

Tushar Saini

Computer Science Engineering 4th year Student , College Of Engineering Roorkee, Roorkee Uttarakhand,247667 India

sainitushar18899@gmail.com

***Abstract*— This paper describes my project In which comparing Support Vector Machines to 3 layer neural Networks on the Titanic Dataset . Anaconda open_source software is used for creating the project and Titanic dataset is used which is provided by the Kaggle.**

***Index Terms*—Support Vector Machine(SVM), Neural Network, Machine Learning, Deep learning , Data Science, Supervised Learning**

I. Introduction

This document is a final report from my Internship Project as an undergraduate student. In this Project I compared Support Vector Machines to 3 layer Neural Network on the Titanic datset ,dataset is extracted from the Kaggle.Python provide many libraries for Machine learning using this inbuilt libraries project is done.

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.we complete the analysis of what sorts of people were likely to survive. In particular, we ask you to apply the tools of machine learning to predict which passengers survived the tragedy.

II. Procedure for Project Creation

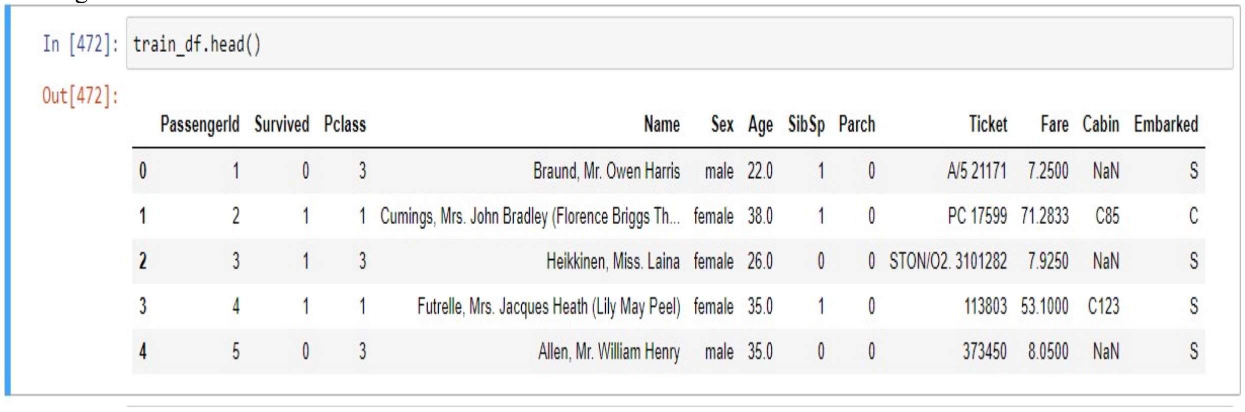
A. Gathering Data

It is the first real step of machine learning- Gathering Data. This step is very crucial as the quality and quantity of data gathered will directly determine how good the predictive model will turn out to be. The data collected is then tabulated and called as Training Data. The data has been split into two groups.

1. Training Dataset (train.csv)
2. Test Dataset (test.csv)

The training set should be used to build machine learning models. For the training set, we provide the outcome (also known as the "ground truth") for each passenger. Your model will be based on "features" like passengers' gender and class. You can also use feature Engineering to create new features. The test set should be used to see how well model performs on unseen data. For the test set, we do not provide the ground truth for each passenger. to predict these outcomes, For each passenger in the test set, use the model you trained to predict whether or not they survived the sinking of the Titanic.

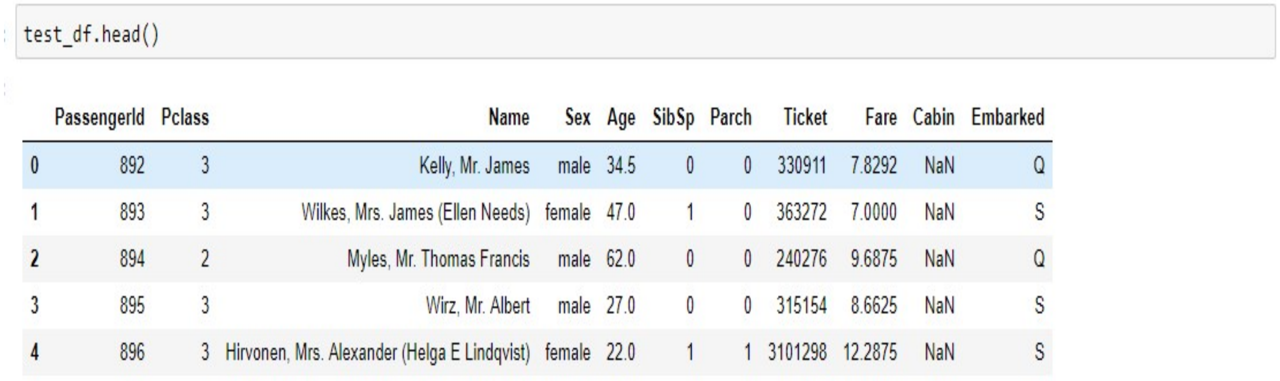
Training Dataset :



```
In [472]: train_df.head()
Out[472]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Test Datset :



```
test_df.head()
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

B. Data Preparation

After the training data is gathered, we move on to the next step of machine learning: Data preparation, where the data is loaded into a suitable place and then prepared for use in machine learning training. Here, the data is first put all together and then the order is randomized as the order of data should not affect what is learned.

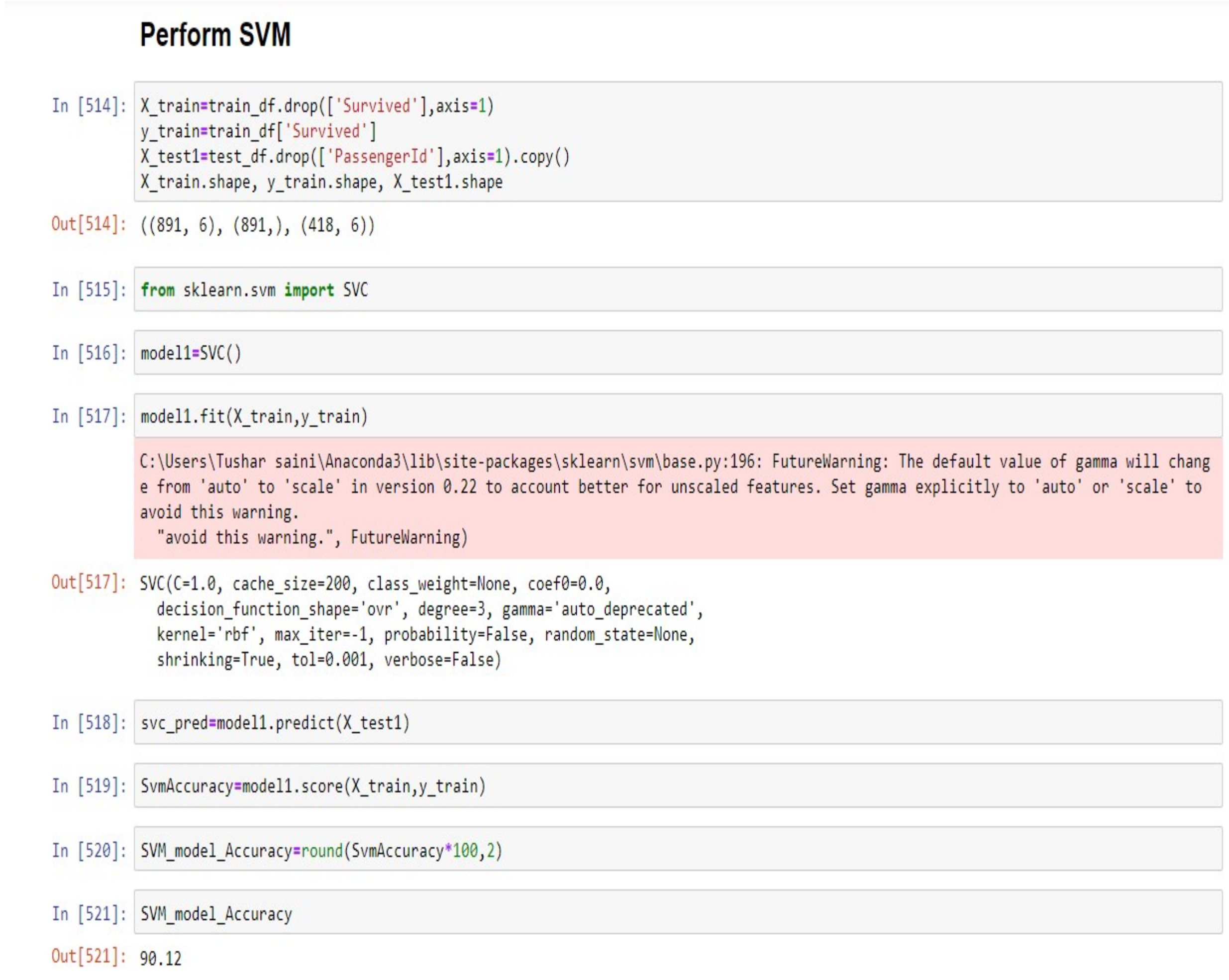
This is also a good enough time to do any visualizations of the data, as that will help we see if there are any relevant relationships between the different variables, how we can take their advantage and as well as show we if there are any data imbalances present. Also, the data now has to be split into two parts. The first part that is used in training our model, will be the majority of the dataset and the second will be used for the evaluation of the trained model's performance. The other forms of adjusting and manipulation like normalization, error correction, and more take place at this step.

C. Choosing a Model

The next step that follows in the workflow is choosing a model among the many that researchers and data scientists have created over the years. We campare SVM and 3 layer Neural Network so we used this two.

Support Vector Machine (SVM) :

In Machine learning, support-vector machines (SVMs, also support-vector networks[1]) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

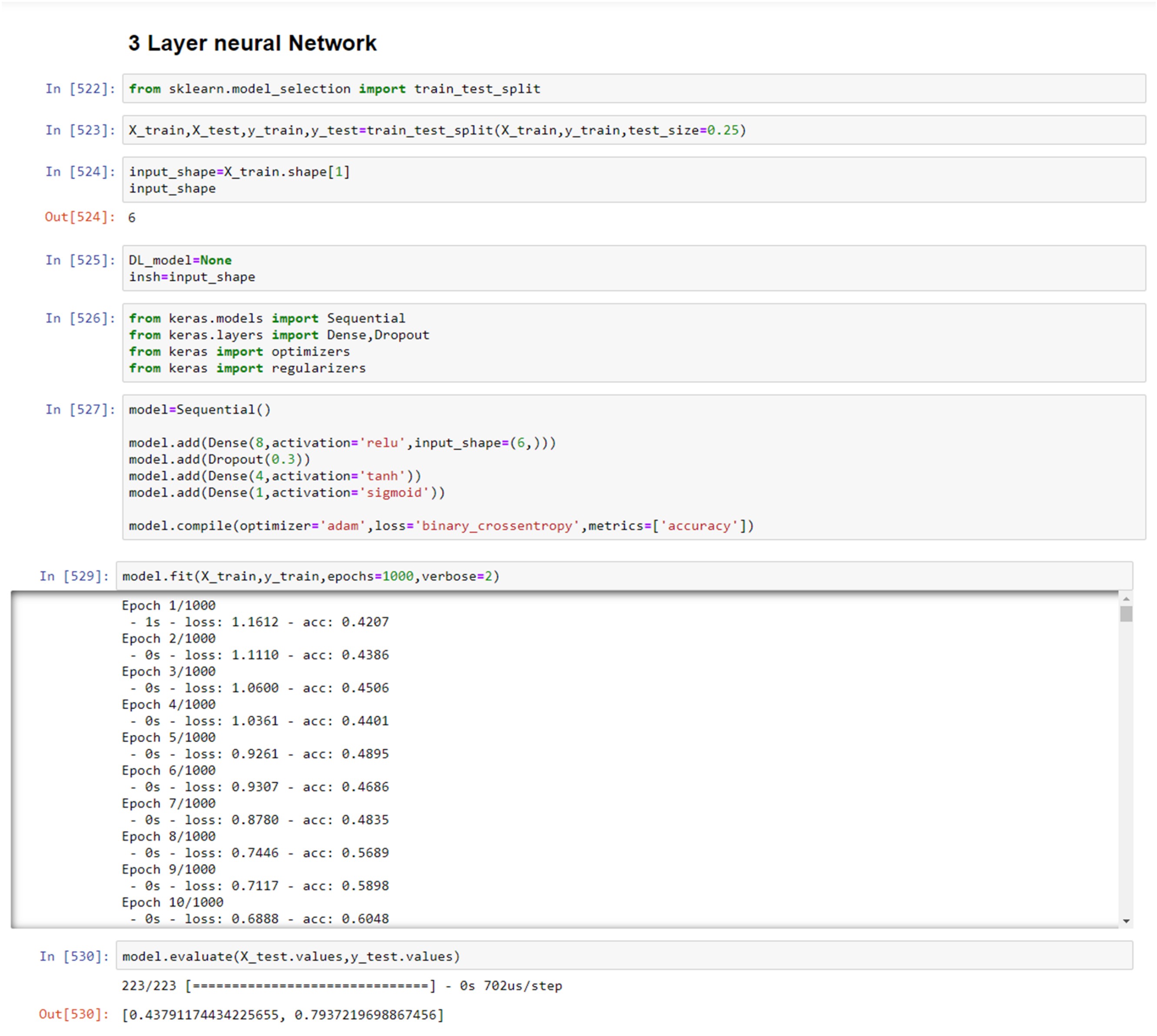


Perform SVM

```
In [514]: X_train=train_df.drop(['Survived'],axis=1)
          y_train=train_df['Survived']
          X_test1=test_df.drop(['PassengerId'],axis=1).copy()
          X_train.shape, y_train.shape, X_test1.shape

Out[514]: ((891, 6), (891,), (418, 6))

In [515]: from sklearn.svm import SVC

In [516]: model1=SVC()

In [517]: model1.fit(X_train,y_train)

          C:\Users\Tushar saini\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: FutureWarning: The default value of gamma will chang
          e from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to
          avoid this warning.
            "avoid this warning.", FutureWarning)

Out[517]: SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
            decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
            kernel='rbf', max_iter=-1, probability=False, random_state=None,
            shrinking=True, tol=0.001, verbose=False)

In [518]: svc_pred=model1.predict(X_test1)

In [519]: SvmAccuracy=model1.score(X_train,y_train)

In [520]: SVM_model_Accuracy=round(SvmAccuracy*100,2)

In [521]: SVM_model_Accuracy

Out[521]: 90.12
```

Neural Network :

A neural network is a network or circuit of neurons, or in a modern sense, an artificial neural network, composed of artificial neurons or nodes. Thus a neural network is either a biological neural network, made up of real biological neurons, or an artificial neural network, for solving artificial intelligence (AI) problems. The connections of the biological neuron are modeled as weights. A positive weight reflects an excitatory connection, while negative values mean inhibitory connections. All inputs are modified by a weight and summed. This activity is referred as a linear combination



3 Layer neural Network

```
In [522]: from sklearn.model_selection import train_test_split

In [523]: X_train,X_test,y_train,y_test=train_test_split(X_train,y_train,test_size=0.25)

In [524]: input_shape=X_train.shape[1]
          input_shape

Out[524]: 6

In [525]: DL_model=None
          insh=input_shape

In [526]: from keras.models import Sequential
          from keras.layers import Dense,Dropout
          from keras import optimizers
          from keras import regularizers

In [527]: model=Sequential()

          model.add(Dense(8,activation='relu',input_shape=(6,)))
          model.add(Dropout(0.3))
          model.add(Dense(4,activation='tanh'))
          model.add(Dense(1,activation='sigmoid'))

          model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])

In [529]: model.fit(X_train,y_train,epochs=1000,verbose=2)

          Epoch 1/1000
           - 1s - loss: 1.1612 - acc: 0.4207
          Epoch 2/1000
           - 0s - loss: 1.1110 - acc: 0.4386
          Epoch 3/1000
           - 0s - loss: 1.0600 - acc: 0.4506
          Epoch 4/1000
           - 0s - loss: 1.0361 - acc: 0.4401
          Epoch 5/1000
           - 0s - loss: 0.9261 - acc: 0.4895
          Epoch 6/1000
           - 0s - loss: 0.9307 - acc: 0.4686
          Epoch 7/1000
           - 0s - loss: 0.8780 - acc: 0.4835
          Epoch 8/1000
           - 0s - loss: 0.7446 - acc: 0.5689
          Epoch 9/1000
           - 0s - loss: 0.7117 - acc: 0.5898
          Epoch 10/1000
           - 0s - loss: 0.6888 - acc: 0.6048

In [530]: model.evaluate(X_test.values,y_test.values)

          223/223 [==============================] - 0s 702us/step

Out[530]: [0.43791174434225655, 0.7937219698867456]
```

D. Training

After the before steps are completed, you then move onto what is often considered the bulk of machine learning called training where the data is used to incrementally improve the model’s ability to predict.The training process involves initializing some random values for say A and B of our model, predict the output with those values, then compare it with the model's prediction and then adjust the values so that they match the predictions that were made previously.

This process then repeats and each cycle of updating is called one training step.

E. Evaluation

Once training is complete, you now check if it is good enough using this step. This is where that dataset you set aside earlier comes into play. Evaluation allows the testing of the model against data that has never been seen and used for training and is meant to be representative of how the model might perform when in the real world

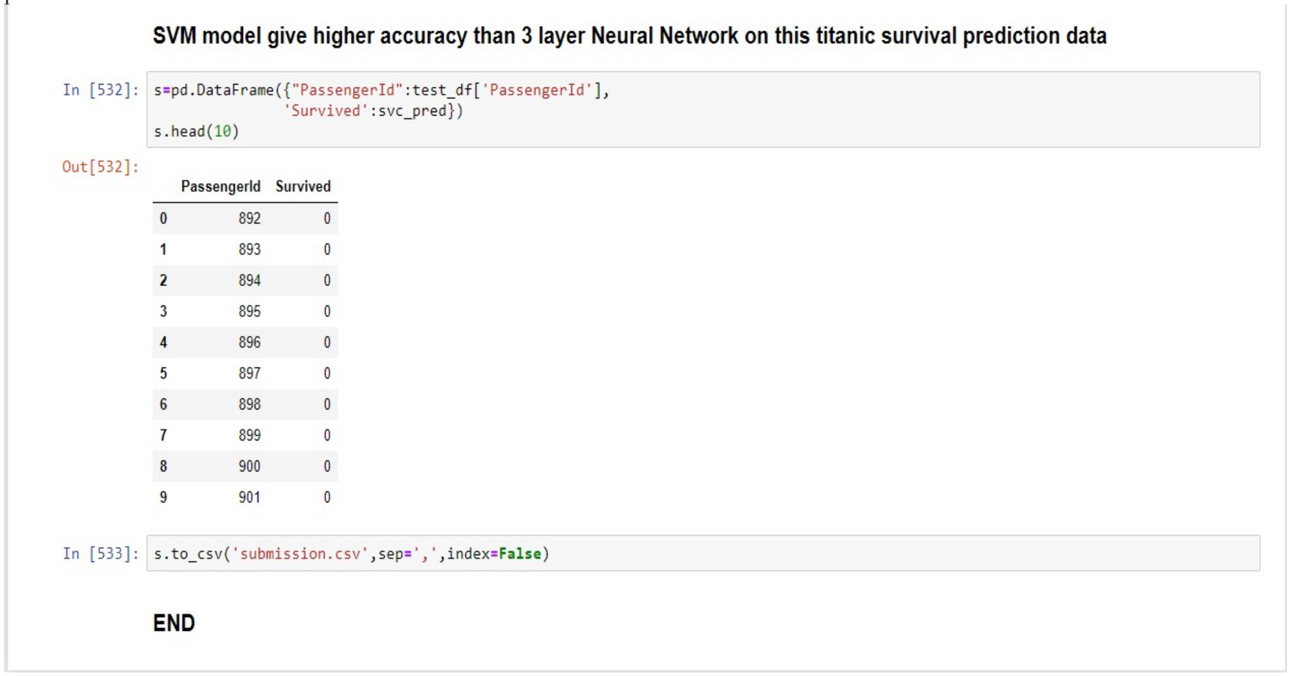
F. Parameter Tuning

Once the evaluation is over, any further improvement in your training can be possible by tuning the parameters. There were a few parameters that were implicitly assumed when the training was done. Another parameter included is the learning rate that defines how far the line is shifted during each step, based on the information from the previous training step. These values all play a role in the accuracy of the training model, and how long the training will take.

For models that are more complex, initial conditions play a significant role in the determination of the outcome of training. Differences can be seen depending on whether a model starts off training with values initialized to zeroes versus some distribution of values, which then leads to the question of which distribution is to be used. Since there are many considerations at this phase of training, it’s important that you define what makes a model good. These parameters are referred to as Hyper parameters. The adjustment or tuning of these parameters depends on the dataset, model, and the training process. Once you are done with these parameters and are satisfied you can move on to the last step.

G. Prediction

Machine learning is basically using data to answer questions. So this is the final step where you get to answer few questions. This is the point where the value of machine learning is realized. Here you can Finally use your model to predict the outcome of what you want.
Support Vector Machine(SVM) Gives the Higher accuracy score than 3 Layer Neural Network So we used SVM for prediction.



SVM model give higher accuracy than 3 layer Neural Network on this titanic survival prediction data

```
In [532]: s=pd.DataFrame({"PassengerId":test_df['PassengerId'],
                          'Survived':svc_pred})
          s.head(10)
```

```
Out[532]:
```

	PassengerId	Survived
0	892	0
1	893	0
2	894	0
3	895	0
4	896	0
5	897	0
6	898	0
7	899	0
8	900	0
9	901	0

```
In [533]: s.to_csv('submission.csv',sep=',',index=False)
```

END

Acknowledgments:

We thank Kaggle team for help in preparing the Titanic Datasets We have also greatly benefited from the insights and helpful suggestions on Kaggle , by which we can easily compare Support vector Machine and 3 layer Neural Networks and geeting the best accuracy score by Support vector Machine Learning.

References

1] Analyzing Titanic disaster using machine learning algorithms-Computing, Communication and Automation (ICCCA), 2017 International Conference on 21 December 2017, IEEE.
2] Eric Lam, Chongxuan Tang, "Titanic Machine Learning From Disaster", LamTang-Titanic Machine Learning From Disaster, 2012.
3] S. Cicoria, J. Sherlock, M. Muniswamaiah, L. Clarke, "Classification of Titanic Passenger Data and Chances of Surviving the Disaster", Proceedings of Student-Faculty Research Day CSIS, pp. 1-6, May 2014.
4] Corinna Cortes, Vlasdimir Vapnik, "Support-vector machine"
5] L Breman- "random forests", Machine Learning, 2001 Ng. CS229 Notes, Standford University, 2012.
6] SJ Russsel P Norvig-"Artificial intelligence: A modern
7] Lonnie Stevans, David L. Gleicher, "Who Survived the International Journal of Maritime History, December 2004.
8] MICHAEL AARON WHITLEY, Using statistical learning to predict survival of passengers on the RMS Titanic by Michael Aaron Whitley, 2015.
9] Kunal Vyas, Zeshi Zheng, Lin Li, Titanic- Machine Learning From Disaster- 2015.
10] EECS 349 Titanic- Machine Learning From Disaster, Xiaodong Yang, Northwestern University.