8/18/2020

[Project-Titanic]

[Deep Learning-Long Short-Term Memory (LSTM)]

Zhao, Ying
ECE 5831

Table of Contents

Abstract: The sinking of the Titanic is one of the most tragedy in the history. Titanic, named “unsinkable”, sank after colliding with the huge ice mountain. From data collected, there are only 722 from 2224 passengers and crew survived, because of lack of the lifeboats. Many people use machine learning model to predict which kind of passengers could survive during this disaster. To estimate the survival passenger on the Titanic, Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) are combined in my project. Comparing with the traditional machine learning models, such as linear regression, logistic regression model, and Naïve Bayes, the results illustrated that RNN-LSTM model has the highest forecasting accuracy. The main outstanding of this project is to build a deep learning network model using one of the RNN, vanilla LSTM. Using this model, it could fit some other dataset for the further searching. And based on the basic vanilla LSTM, it still can carry out more different models.

Keywords: Titanic; Survival passenger; deep learning; RNN-LSTM model

1.Introduction

RMS *Titanic* was a British passenger liner operated by the White Star Line that sank in the North Atlantic Ocean in the early morning hours of 15 April 1912, after striking an iceberg during her maiden voyage from Southampton to New York City. Of the estimated 2,224 passengers and crew aboard, more than 1,500 died, making the sinking one of modern history's deadliest peacetime commercial marine disasters. RMS *Titanic* was the largest ship afloat at the time she entered service and was the second of three *Olympic*-class ocean liners operated by the White Star Line. She was built by the Harland and Wolff shipyard in Belfast. Thomas Andrews, chief naval architect of the shipyard at the time, died in the disaster.[1]

While there was some element of luck involved in surviving, it seems some groups of people were more likely to survive than others. So, in this project, I would like to find these groups people, and using this model I build, I predict if this passenger survive or not.

2.Data

2.1 Get the Data

The dataset in this project I use is Titanic from Kaggle.com (https://www.kaggle.com/c/titanic). The data in the Kaggle has three csv data explores, I mainly use two explorers, test.csv and train.csv. These two data explorers contain passengers' information, for example, age, sibling or not, sex, cabin number and so on.

2.2 Data Structure

The data in the Kaggle has been split two groups, training set (891 unique values) and test set (418 unique values). From the class, we all know, the training set will be used to build the model, then using test set to check whether the model is well or not.

In Figure.1, I illustrate the first five rows of training set data, and each value has 12 columns, which contains Passengerld, Survival, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, and Embarked. For illustrating the dataset full-scale, I calculate the average, standard deviation, and other indexes in Figure.2.

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

Figure.1

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	418.000000	332.000000	418.000000	418.000000	417.000000
mean	1100.500000	0.363636	2.265550	30.272590	0.447368	0.392344	35.627188
std	120.810458	0.481622	0.841838	14.181209	0.896760	0.981429	55.907576
min	892.000000	0.000000	1.000000	0.170000	0.000000	0.000000	0.000000
25%	996.250000	0.000000	1.000000	21.000000	0.000000	0.000000	7.895800
50%	1100.500000	0.000000	3.000000	27.000000	0.000000	0.000000	14.454200
75%	1204.750000	1.000000	3.000000	39.000000	1.000000	0.000000	31.500000
max	1309.000000	1.000000	3.000000	76.000000	8.000000	9.000000	512.329200

Figure.2

2.3 Data Dictionary [2]

a. Survived is survival of this passenger, 0 is not survival, and 1 is survival.

b. Pclass is ticket class, 1 is first class ticket (upper class), 2 is the second-class ticket (middle class), and 3 is the third-class ticket (lower class).

c. Sex is sex.

d. Age is age in years. In this dataset, age is fractional if less than 1.

e. Sibsp is number of siblings or spouses aboard the Titanic. Sibling is brother, sister, stepbrother, and stepsister. Spouse is husband and wife.

f. Parch is the number of parents or children.

g. Ticket is the ticket number. (This maybe has no meaning for this project, so I delete)

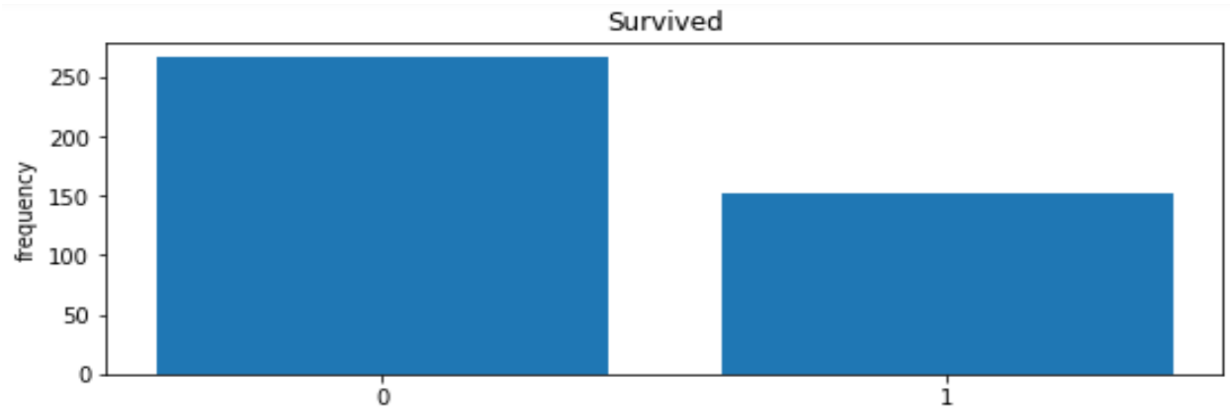
h. Fare is passenger fare. Parent is mother and father. Child is daughter, son, stepdaughter, and stepson. (Some children without parents abroad)

i. Cabin is Cabin number. (This maybe has no meaning for this project, so I delete)

j. Embarked is port of embarkation, C is Cherbourg, Q is Queenstown, and S is Southampton.

3.Explore the data

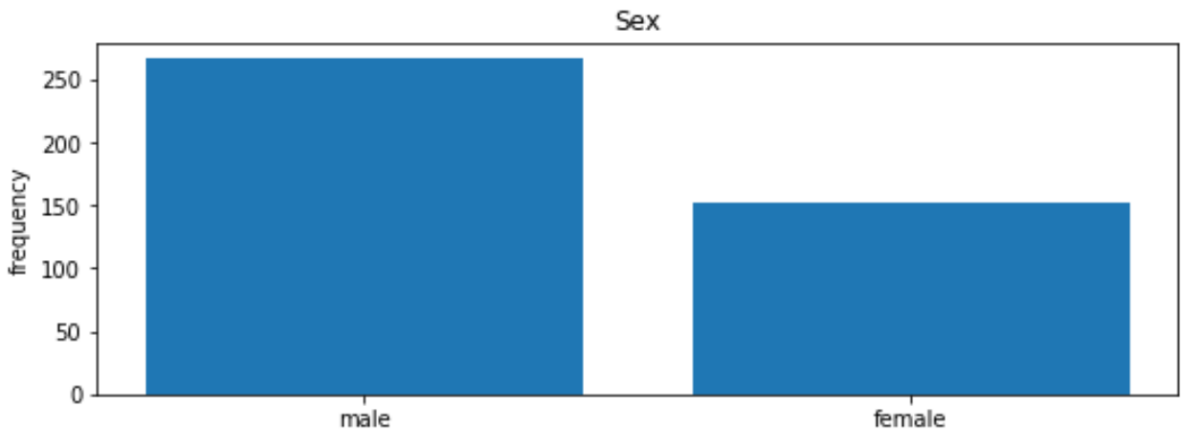
3.1 Data models (In these models, I only use test.csv)




```
Survived:
 0    266
1    152
Name: Survived, dtype: int64
```

Figure.3

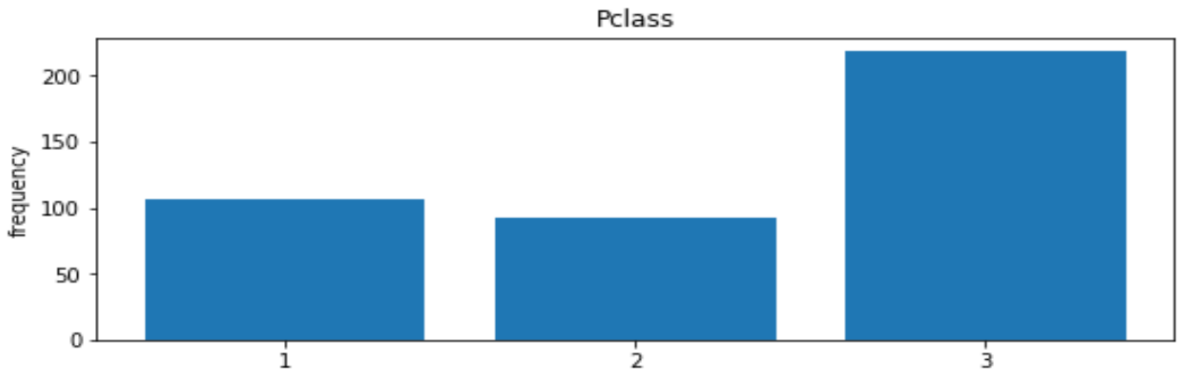
In figure 3, I summarize the survival in test set, from the bar chart, only 152 survived from 418 passengers, unfortunately only 36.36%.




```
Sex:
 male      266
female    152
Name: Sex, dtype: int64
```

Figure.4

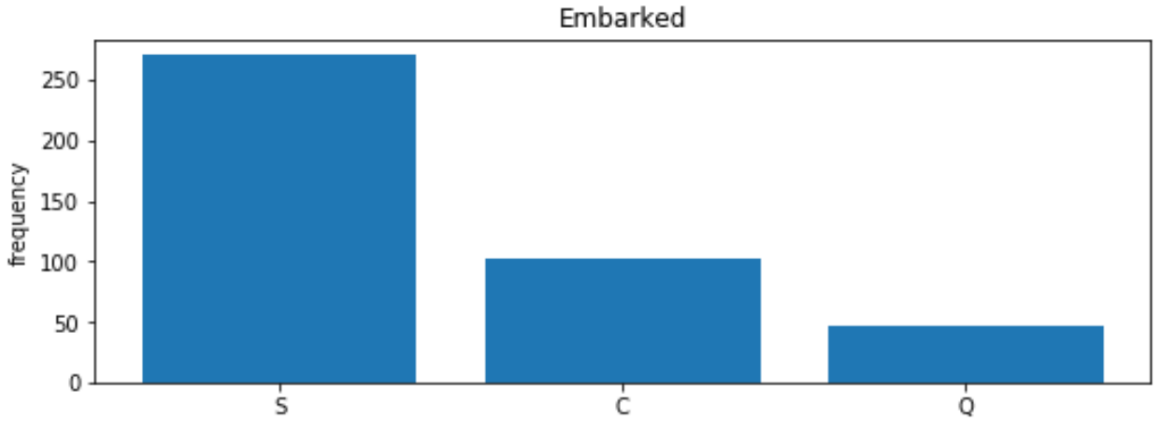
In figure 4, it shows the male and female numbers in the test set, and in this set, male takes the more percentage, nearly 63.6%. Similar in train set, male take the bigger percentage on the Titanic.




```
Pclass:
 3    218
1    107
2     93
Name: Pclass, dtype: int64
```

Figure.5

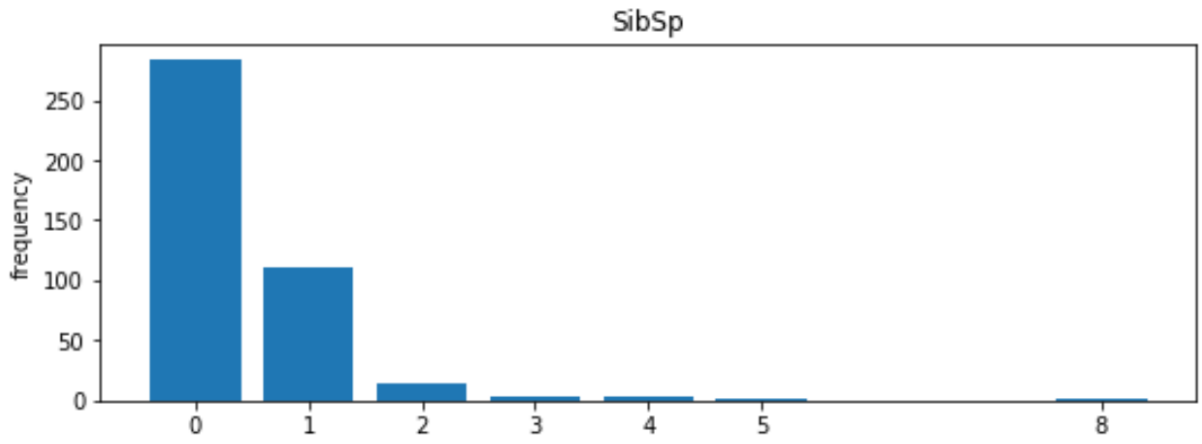
In figure 5, it shows the distribution of the P-class, which could see the socio-economic status from the test set. From 418 passengers, class 3 has the most passengers, 218 (nearly 52.153%), following the class 1 and class 2, 107 passengers and 93 passengers.




```
Embarked:
 S    270
C    102
Q     46
Name: Embarked, dtype: int64
```

Figure.6

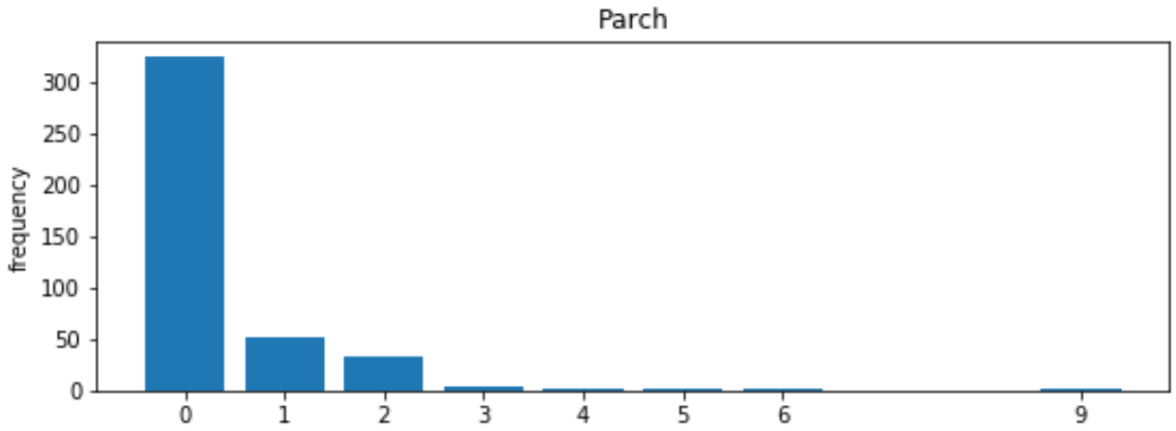
In figure 6, it illustrates the different embarked places for passengers. Southampton(S) has the most people to embark, 270(64.59%). And Cherbourg(C) has 168, and Queenstown(Q) has 77. From its definition, it perhaps has no meaning for the prediction. While, in all my prediction models, I transfer Embarked into dummy variables.




```
SibSp:
 0    283
1    110
2     14
4      4
3      4
8      2
5      1
Name: SibSp, dtype: int64
```

Figure. 7

In figure 7, we could see the sibling and spouse's number of people aboard. Passenger with no relatives on the boat has 283 out of 418. Following that, 1 is 110, 2 is 14, while, seldom passengers have more than two sibling and spouse on the boat




```
Parch:
 0    324
1      52
2      33
3       3
9       2
4       2
6       1
5       1
Name: Parch, dtype: int64
```

Figure.8

In figure 8, it illustrates the parent and children's number of people aboard. Similar like the sibling and spouse, no children and parent on the board is the most, 324 from 418.

3.2 Cross Matrix Model

Survived	0	1	All
Pclass			
1	57	50	107
2	63	30	93
3	146	72	218
All	266	152	418

Figure. 9

	Pclass	1	2	3	All
Sex	Survived				
female	1	50	30	72	152
male	0	57	63	146	266
All		107	93	218	418

Figure.10

For all two cross matrixes above, it just shows the one or two indexes. In Figure 9, we could see that the class 1 (highest socioeconomic status) has the largest percentage of survival rate. Then, after cross analysis from sex and p-class, male from class three has the largest number people survival. So, for the prediction, we could not only concentrate on the single index, and different index has different weights.

Survived	0	1	All
Agroup			
0	186	77	263
1	3	9	12
2	2	8	10
3	8	4	12
4	9	19	28
5	20	9	29
6	11	13	24
7	183	111	294
8	186	106	292
9	131	81	212
10	9	14	23
11	49	34	83
12	18	9	27
All	815	494	1309

Figure.11

And all the information from the dataset, age has the largest range. I divide the passengers into 12 groups according to their ages. We could see that class 7 (age from 16 to 25) has the most survival, but when it comes to the percentage, the younger and older have the higher percentage to survive, we could see that group 1 and group 2 has the top two percentage survival, which age is smaller 1 and from 1 to 2. It shows that people will always save the children and female when disaster comes.

4.Prepare the Data

4.1 Training Set ant Test Set

The model is initially fit on a training dataset, which is a set of examples used to fit the parameters (e.g. weights of connections between neurons in artificial neural networks) of the model.[3]

Finally, the test dataset is a dataset used to provide an unbiased evaluation of a final model fit on the training dataset. If the data in the test dataset has never been used in training (for example in cross-validation), the test dataset is also called a holdout dataset.[4]

In my project, I combine the train set and test set firstly, because there are some missing data and normalization together later. Then I use train_test_split(python code) to split the dataset into 2/3 and 1/3 for new train set and test set.

4.2Missing data

In statistics, missing data, or missing values, occur when no data value is stored for the variable in an observation. Missing data are a common occurrence and can have a significant effect on the conclusions that can be drawn from the data.[5]

Because of the missing indexes in the dataset, I import Imputer (from sclera) to replace the missing data, such as some passengers' ages are Nan. In my data frame, I choose 'mean' to replace the missing items.

4.3Feature Scaling

Feature scaling is a method used to normalize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step.[6]

There are many ways for scaling, for example standardize or normalize. In my project, I choose the standardize for scaling, because input layers have 10 variables,

after scaling, all of them in the same range. The figure below shows the first train data after preparing.

4.4 Dummy variables

In statistics and econometrics, particularly in regression analysis, a dummy variable[a] is one that takes only the value 0 or 1 to indicate the absence or presence of some categorical effect that may be expected to shift the outcome.[4][5] They can be thought of as numeric stand-ins for qualitative facts in a regression model, sorting data into mutually exclusive categories (such as smoker and non-smoker).[7]

In my project, I drop some columns, which is no contribution to the prediction, such as Name, ticket, cabin, and passenger number. Then, I transfer some variables into dummy variables, such as sex and embarked.

5. Different Models Analysis and Results

5.1 Simple Linear regression

In statistics, simple linear regression is a linear regression model with a single explanatory variable. That is, it concerns two-dimensional sample points with one independent variable and one dependent variable (conventionally, the x and y coordinates in a Cartesian coordinate system) and finds a linear function (a non-vertical straight line) that, as accurately as possible, predicts the dependent variable values as a function of the independent variables. The adjective simple refers to the fact that the outcome variable is related to a single predictor.[8]

5.1.1 Classifier and Results

I use all the variables into input. I import LinearRegression from sklearn, then using my X_trian and y_train to make my classifier. Then I predict my model using this classifier.

```
 0.78896981,  0.75315956,  0.09411097,  0.10911562,  1.00206447,
 0.29126439,  0.09437507,  0.07599871,  0.66738568,  0.97122642,
 0.10336966,  0.09411097,  0.86689311,  0.23467192,  0.682823  ,
 0.8915252 ,  0.69150553,  0.96432253,  0.70033455,  0.7332772 ,
 0.0627659 ,  0.72742911,  0.47958389,  0.26603115,  0.64882951,
 0.73333159,  0.68273003,  0.79119732,  0.8233727 ,  0.15321657,
-0.04323123, -0.2086883 ,  0.02079293,  0.23890089,  0.77441778,
 0.10322858,  0.08598222,  0.88011966,  0.82657068,  0.28100153,
 0.67766791,  0.11216333,  0.96894756,  0.24219268,  0.05005507,
 0.25439901,  0.85939219,  0.7437948 ,  0.81397612,  0.97494289,
```

Figure.12

5.2 Naïve Bayes

In machine learning, naïve Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. They are among the simplest Bayesian network models. But they could be coupled with Kernel density estimation and achieve higher accuracy levels. [9]

5.2.1 Classifiers and Results

The classifier is similar with the classifier below, and I import GaussianNB from sklearn. Then I compare the results, prediction from X_test, with the y_test, using confusion matrix. The result shows in figure.15. From this graph, the result is good enough for prediction. From this graph, for example, 217 means that 217 passengers were predicted not survival which result is right.

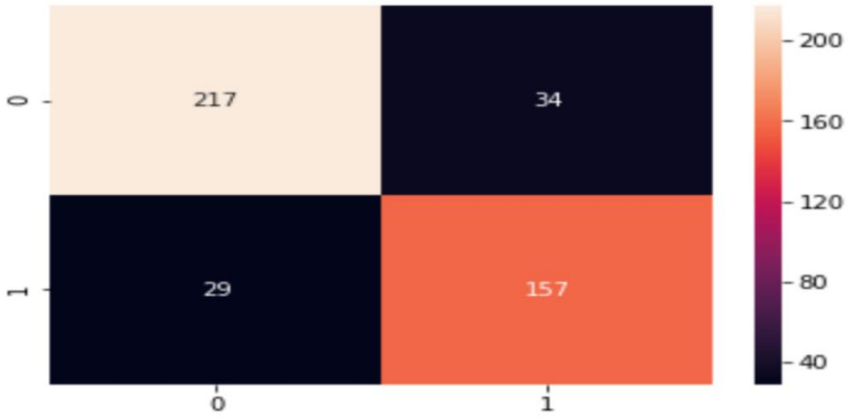


Figure.13

5.3 Logistic Regression

In statistics, the logistic model (or logit model) is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick. This can be extended to model several classes of events such as determining whether an image contains a cat, dog, lion, etc. Each object being detected in the image would be assigned a probability between 0 and 1, with a sum of one.[10]

5.3.1 Classifier and Results

The classifier is similar with the classifier below, and I import LogisticiRegression from sklearn. Then I compare the results, prediction from X_test, with the y_test, using confusion matrix. The result shows in Figure.16. From this graph, the result is good enough for prediction. From this graph, for example, 224 means that 224 passengers was predicted not survival which result is right.

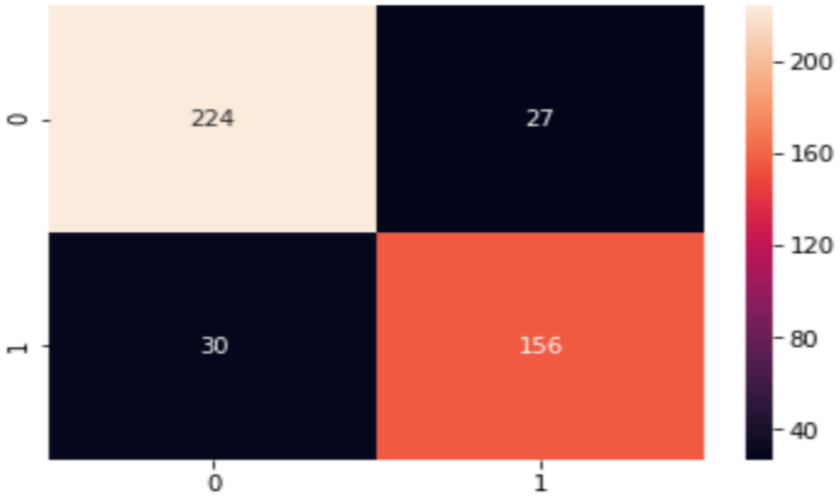


Figure.14

6.Long Short-Term Memory (LSTM) Model

6.1 Recurrent Neural Network

Recurrent neural network is the repetition of a single cell, like the figure below. So, this network first implements the computations for a single time step, then goes through the whole network. [11]

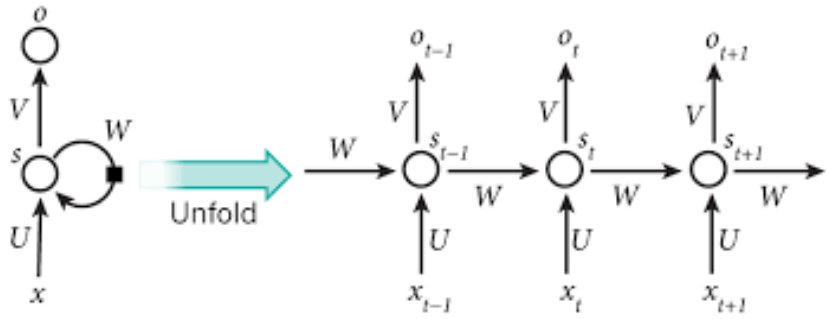


Figure.15

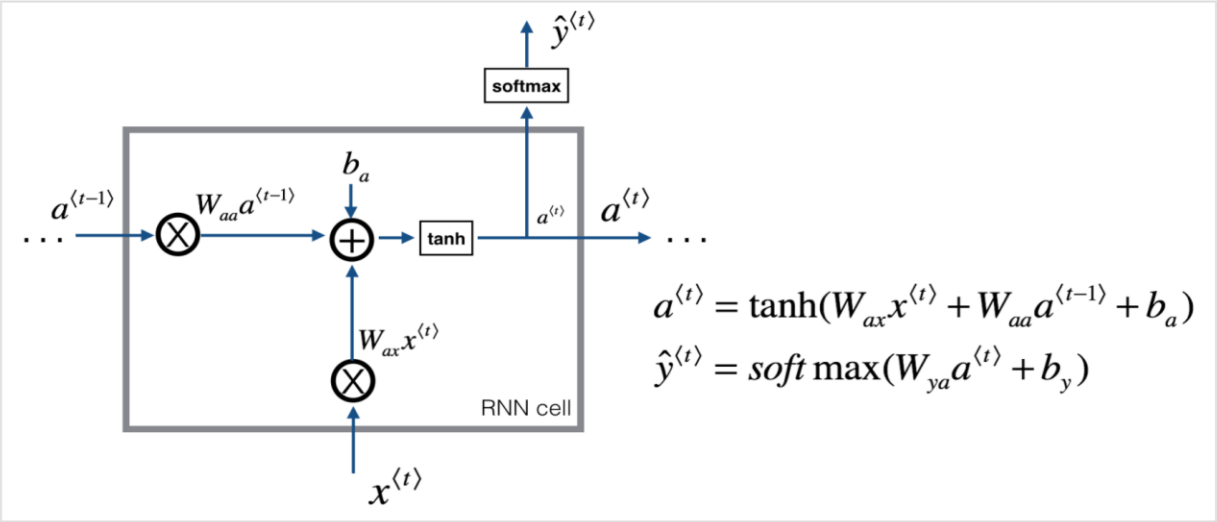


Figure16. Basic RNN cell [11]

This is the basic algorithm for the single RNN cell, and takes as input $x^{[t]}$ (current input) and $a^{[t-1]}$ (previous hidden cell containing information from the past), and outputs $a^{[t]}$ which is given to the next RNN cell and also used to predict $y^{[t]}$.[11]

Using this network, it can work well for the multiple information in input and combining useful to pass through whole network. Even though, with the huge dataset, it still could help delete redundant information, which save the algorithm space and time. It is advantages for the deep learning, especially for the big data analysis.

6.2 Long Short-Term Memory

Long Short-Term Memory is a significant branch of recurrent neural network (RNN), and it is effective in many sequence problems. Thus, many kinds of LSTM models, such as Vanilla LSTMs, Stacked LSTMs, CNN LSTMs, Encoder-Decoder LSTMs, and so on. In my project, I mainly carry out the vanilla LSTMs.

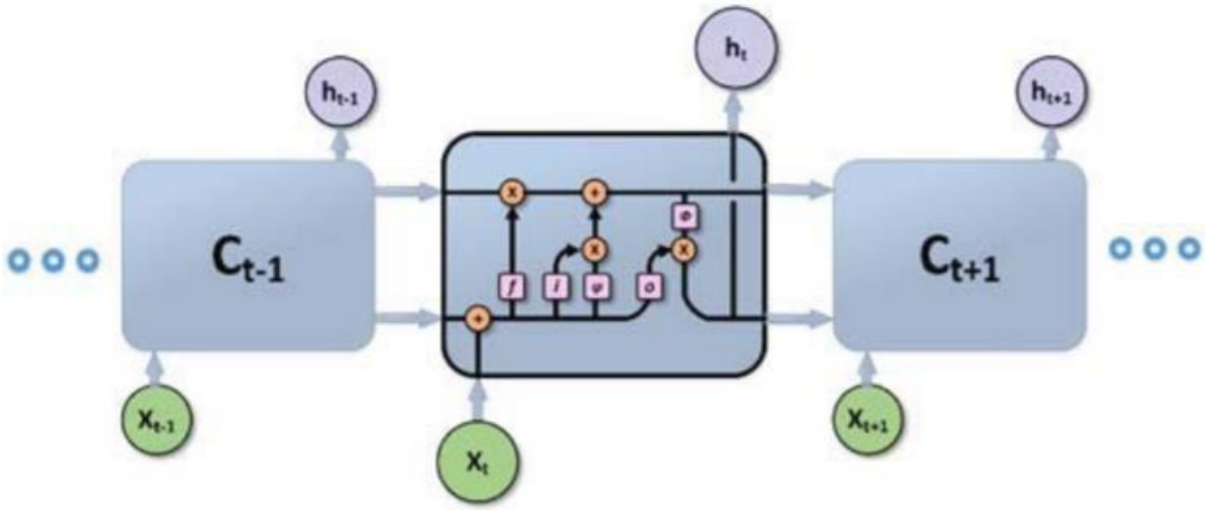


Figure17. Vanilla LSTM

6.2.1 Concept of LSTM

The forget gate(ft) decides which historical information will be discarded from the cell state, the input gate(it) decides which states will be updated and the output gate(ot) decides which part of the cell states will be outputted. In this way, LSTM can remove or add information to the cell state, instead of the mechanism of completely overriding cell states taken by classical RNN. [12] The figure below is the single LSTM memory cell, including forget gate, input gate, and output gate, and in the whole LSTM, there would be a sequence of LSTM combining.

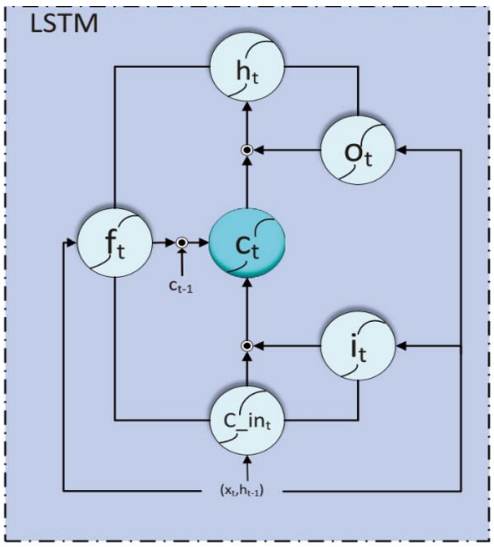


Figure18. The architecture of LSTM memory cell

6.2.2 Algorithm of LSTM

$$
\begin{aligned}
\boldsymbol{g}_l^{(t)} &= \phi(W_l^{\mathrm{gx}} \boldsymbol{h}_{l-1}^{(t)} + W_l^{\mathrm{gh}} \boldsymbol{h}_l^{(t-1)} + \boldsymbol{b}_l^{\mathrm{g}}) \\
\boldsymbol{i}_l^{(t)} &= \sigma(W_l^{\mathrm{ix}} \boldsymbol{h}_{l-1}^{(t)} + W_l^{\mathrm{ih}} \boldsymbol{h}_l^{(t-1)} + \boldsymbol{b}_l^{\mathrm{i}}) \\
\boldsymbol{f}_l^{(t)} &= \sigma(W_l^{\mathrm{fx}} \boldsymbol{h}_{l-1}^{(t)} + W_l^{\mathrm{fh}} \boldsymbol{h}_l^{(t-1)} + \boldsymbol{b}_l^{\mathrm{f}}) \\
\boldsymbol{o}_l^{(t)} &= \sigma(W_l^{\mathrm{ox}} \boldsymbol{h}_{l-1}^{(t)} + W_l^{\mathrm{oh}} \boldsymbol{h}_l^{(t-1)} + \boldsymbol{b}_l^{\mathrm{o}}) \\
\boldsymbol{s}_l^{(t)} &= \boldsymbol{g}_l^{(t)} \odot \boldsymbol{i}_l^{(i)} + \boldsymbol{s}_l^{(t-1)} \odot \boldsymbol{f}_l^{(t)} \\
\boldsymbol{h}_l^{(t)} &= \phi(\boldsymbol{s}_l^{(t)}) \odot \boldsymbol{o}_l^{(t)}.
\end{aligned}
$$

Figure.19

Given a series of observations x (1), ..., x(T), we learn a classifier to generate hypotheses yˆ of the true labels y. Here, t indexes sequence steps, and for any example, T stands for the length of the sequence. Our proposed LSTM-RNN uses memory cells with forget gates.[13]

In these equations, σ stands for an element-wise application of the sigmoid (logistic) function, ϕ stands for an element-wise application of the tanh function and is the Hadamard (elementwise) product. The input, output, and forget gates are denoted by i, o, and f respectively, while g is the input node and has a tanh activation.[13]

6.4 Build Vanilla LSTM Model

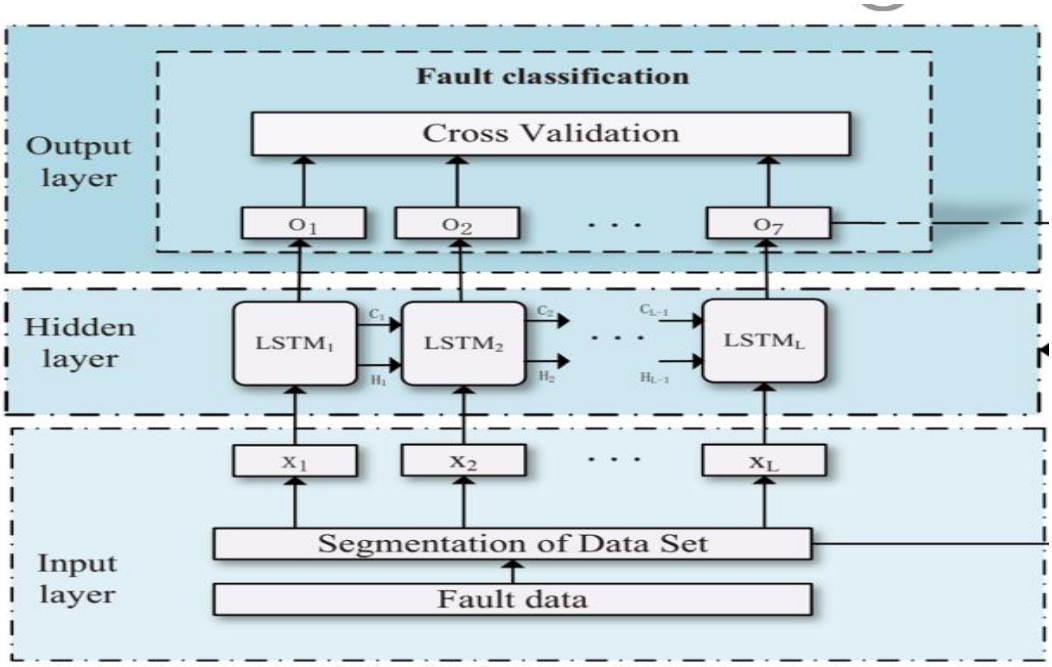


Figure.20

Hidden layer number: In the figure above, because of dataset is not so huge, I build three hidden layers, and each layer's epoch I build is 100.

Dropout regularization: In the hidden layer, I eliminate some nodes, then it can remove some redundant items in case of overfitting. In my model, I drop out at every hidden layer, and drop rate is nearly 10%.

6.5 Tune LSTMs

After training data n Keras, I diagnostic plots to diagnose whether my RNN-LSTM model is underfitting or overfitting. For the definition, when model is underfitting, it performs well at training set but poor on the test dataset. In contrast, when model is overfitting, it performs well at test dataset but poor on the training set. When I implement my model, both works well.

But I still change the drop rate, hidden layers number, but it shows that the model I build illustrate highest accuracy. And then I change the activation for hidden layers

and dense rate, and it shows that when activation equation is ReLu gets the highest results.

6.6 Further Research

Nowadays, there are many other model architectures which add to the LSTM model. In the figure below is architecture combining LSTM and FCN (Fully convolutional network). This model is for multivariate time series classification.

A time series dataset can be univariate, where a sequence of measurements from the same variable are collected, or multivariate, where a sequence of measurements from multiple variables or sensors are collected. [14] So, for the multivariate time series classification, the model below will be better than the single LSTM models.

So there are many different situations, for the further research, I could add many other different models with LSTM or other deep learning model.

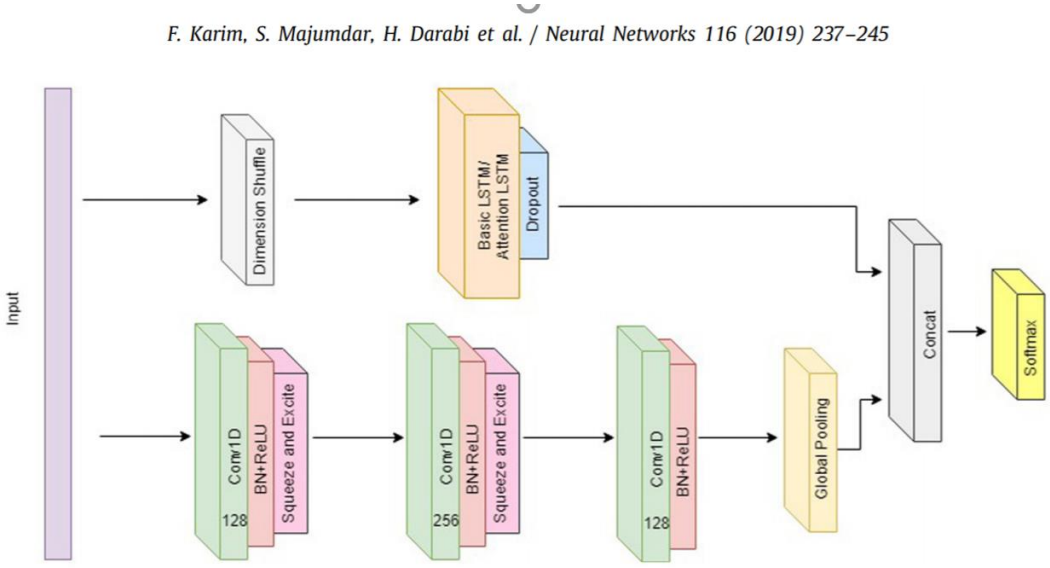


Figure.21

7.Conclusion

	Simple Linear regression	Naïve Bayes	Logistic Regression	LSTM
Accuracy	0.84668	0.8260	0.83981	0.858123

[1] https://en.wikipedia.org/wiki/Titanic

[2] https://www.kaggle.com/c/titanic/data

[3][4] https://en.wikipedia.org/wiki/Training,_validation,_and_test_sets#cite_noteBrownlee-5

[5] https://en.wikipedia.org/wiki/Missing_data

[6] https://en.wikipedia.org/wiki/Feature_scaling#:~:text=Feature%20scaling%20is%20a%20 method, during%20the%20data%20preprocessing%20step.

[7] https://en.wikipedia.org/wiki/Dummy_variable_(statistics)

[8] https://en.wikipedia.org/wiki/Simple_linear_regression

[9] https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[10] https://en.wikipedia.org/wiki/Logistic_regression

[11] https://snaildove.github.io/2018/06/02/Building+a+Recurrent+Neural+Network+-+Step+by+Step+-+v3/

[12] Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network (2016 IEEE/CSAA International Conference on Aircraft Utility Systems (AUS) October 10-12,2016 Beijing, China)

[13] Learning to Diagnose with LSTM Recurrent Neural Networks Zachary C. Lipton, David C. Kale, Charles Elkan, Randall Wetzel

[14] Multivariate LSTM-FCNs for time series classification, Fazle Karim a, Somshubra Majumdar b , Houshang Darabi a,∗ , Samuel Harford a