Titanic - Machine Learning from Disaster

TEAM 12: Mey Yeh (102012805), Huey-Chii Liang (107011153), Chih-Mei Young (109032805)

I. Introduction

The sinking of the Titanic is undoubtedly one of the deadliest commercial peacetime maritime disasters in modern history. This infamous shipwreck took more than 1500 lives from an estimated 2,224 passengers and crew aboard the ship. Learning from the tragedy, we found that certain groups of people seemed more likely to survive than others. We are motivated by this indication to explore whether the survival of passengers encountering this disaster is predictable based on the known background information via machine learning technique.

In this final project, we aimed to investigate the effects of feature combination and labeling on the accuracy of survival prediction. By adopting two generally used machine learning algorithms to train the preprocessed data, we successfully predicted the survival of the passengers on Titanic with the accuracy of 79.7% testing the public data and 82.6% upon applying to the private testing data.

II. Materials and Methods

Apparently, there were two crucial factors that contributed to the accuracy of prediction, which were how the data has been preprocessed and the mathematical models having been utilized to predict the survival.

A. Data-preprocessing

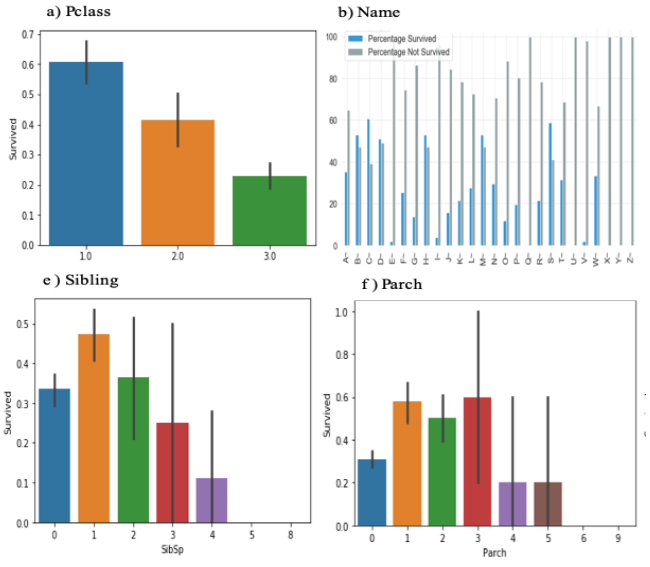
(1) Background Information

In our training data set, there is a total of 891 passengers. Unfortunately, some of their background information was missing, adding to the difficulties in preprocessing and training. The Age feature, which may largely affect the survival, is missing approximately 21.2 % of its values. We substituted these missing values with the mean age. For the Cabin feature, about 77.4% of its values were missing. Due to the complexity to fill in the missing values, we dropped them from our dataset. The missing 0.22% values of the Embarked feature is so trivial that the effect of these missing data can be neglected. To label the Ticket feature is indeed a tough task. We presumed that the information included in the Ticket feature were associated with the features Fare and Pclass.

(2) Visualization of the Data

To comprehensively visualize and analyze the training data, eight histograms were constructed to interpret the relation between the survivors and their background.

As shown in **figure 1a**, the passengers with higher socioeconomic class had a higher rate of survival (60.8% vs. 41.4% vs. 23%). **Figure 1b** indicates that the passengers with some specific first names were more possible to survive. According to **Figure 1c**, the survival of females was obviously higher than that of males, implying that the Sex feature is influential in predicting the survival. Depicted in **figure 1d**, it is apparent that infants are more likely to survive than any other age group. The relation between the number of family members on board and survival was revealed in **figure 1e** and **1f**. **Figure 1e** manifests that the survival rate of the passengers traveling alone or with more than three siblings or spouses was lower than those with one or two. Similarly, passengers being aboard with few parents or children were more likely to survive compared to those with none or many as demonstrated in **figure 1f**. In general, the passengers who were aboard Titanic either alone or with more than three family members were highly possible to die in this disaster. Therefore, we created two features, which specified whether the passengers traveled alone or not and the number of their relatives on board respectively. In **figure 1g**, it can clearly be seen that ticket fares were positively correlated with their survival. For the missing data in the Embarked feature, since the majority of people embarked in Southampton from our observation, we replaced the missing values with S as illustrated in **figure 1h**. Detailed description of how the features were labelled was exhibited in **table 1**.



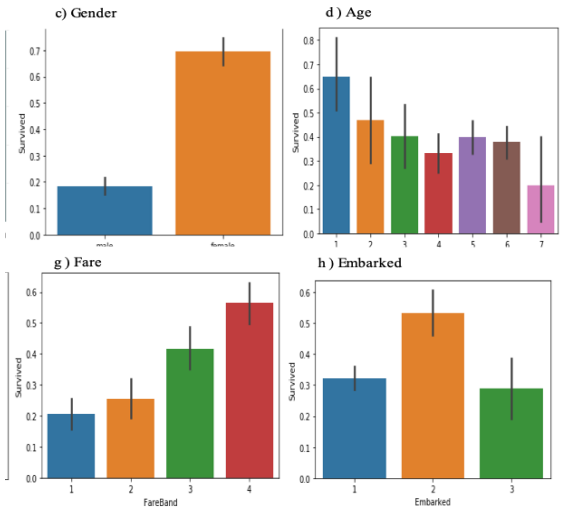


Figure 1. Visualization of the Data via histograms

Table 1. Labelling methods of each feature

Features	Description
Pclass	1=1st; 2=2nd; 3=3rd
Name	The first letter of last name. If the letter is A, label as 1.
Sex	Women=1, Man=0.
Age	1st: 6 group, 0-5years:1; 5-12 years:2; 12-18years:3; 18-24years:4; 24-35years:5; 35-60years:6; over 60years:7 ; 2nd: real number
SibSp	real number
Parch	real number
Fare	1st:real number; 2nd:fare per person (fare / number of relatives)
Embarked	C=0, Q=1, S=2

(3) Correlation between the Features

Correlation is a term used to represent the statistical measurement of dependence between two different variables. The correlation between multiple variables can be demonstrated in a correlation matrix.

Pearson correlation coefficient measures how the value of two different variables vary with respect to each other. The formula given below (eq.1) represents Pearson correlation coefficient.

$$\frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \quad \text{(eq.1)}$$

Noted that:

- If the value is 1, the two variables are positively correlated.
- If the value is -1, the two variables are negatively correlated.
- If the value is 0, there is no correlation between the two variables.

The correlation between the features was given by Seaborn library. The correlation heatmap shown in **figure 2** suggested that the survival is mostly related to the features "Sex," "Pclass," "Fare," and "Embarked."

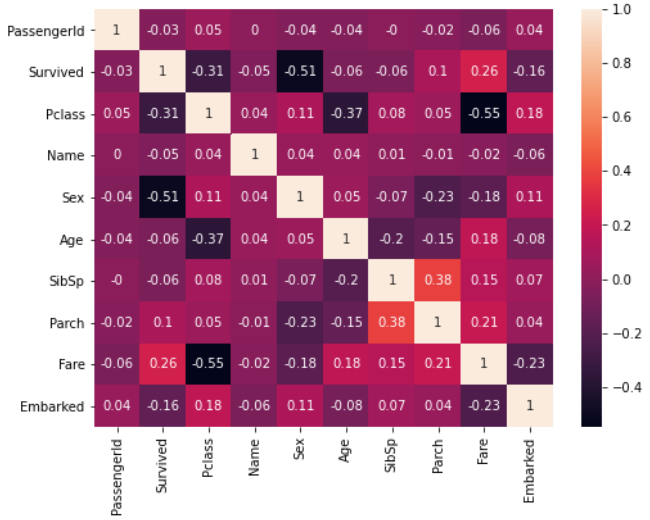


Figure 2. Correlation heatmap between the features

B. Selection of Models

In this project, two of the most commonly adopted machine learning models, Gaussian Naïve Bayes Classifier (NBC) and K-nearest Neighbor (KNN), were adopted. The related mathematical algorithms will be discussed in the following sections.

(1) Gaussian Naïve Bayes Classifier

Gaussian Naïve Bayes Classifier (NBC) is defined as a collection of classification algorithms based on Bayes' theorem, where y = class variable, X = (x_1, x_2, x_3, x_4, …, x_n) a dependent feature vector (of size n) in eq.2.

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)} \quad \text{(eq.2)}$$

Using Bayesian probability terminology, this equation can also be written as

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}} \quad \text{(eq.3)}$$

Naïve assumption, P(A∩B) = P(A)P(B), assumes that each feature makes an equal and independent contribution to the outcome.

Combining Bayes' theorem and Naïve assumption we get the following equation eq.4

$$P(y|x_1, x_2, x_3, x_4, \ldots, x_n) = \frac{P(x_1|y)P(x_2|y)P(x_3|y)\ldots P(x_n|y)\,P(y)}{P(x_1)P(x_2)P(x_3)\ldots P(x_n)} \quad \text{(eq.4)}$$

For Gaussian NBC, continuous values associated with each feature are assumed to be distributed according to a

Gaussian distribution. The likelihood of the features is assumed to be Gaussian; hence, conditional probability is given by

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(xi-\mu_y)^2}{2\sigma_y^2}\right) \quad \text{(eq.5)}$$

The code from scratch is shown in **figure 3**. First, we input x_train and y_train into the subroutine "split_classes" as shown in **figure 4**, which can divide input into class 1(survived) and class 2 (dead). Next, we calculated the mean, standard deviation and probability of each class, and computed the discriminant function of each class. If the discriminant function of class 1 is greater than class 2, we predict that the passenger will survive; otherwise, we predict that he will die.

```
def Gaussian_NB(X_train, y_train, X_test):
    y_pred = np.zeros(len(X_test))
    X_train_C1, X_train_C2, y_train_C1, y_train_C2 = split_classes(X_train, y_train)
    C1_mean = np.mean(X_train_C1, axis=0)
    C2_mean = np.mean(X_train_C2, axis=0)
    C1_std = np.std(X_train_C1, axis=0, ddof=1)
    C2_std = np.std(X_train_C2, axis=0, ddof=1)
    C1_prob = len(y_train_C1) / len(y_train)
    C2_prob = len(y_train_C2) / len(y_train)
    for i in range(len(X_test)):
        g1 = ln(C1_prob) - 0.5 * np.sum(( (X_test[i] - C1_mean) / C1_std )**2)
        g2 = ln(C2_prob) - 0.5 * np.sum(( (X_test[i] - C2_mean) / C2_std )**2)
        if g1 > g2:
            y_pred[i] = 1
        else:
            y_pred[i] = 0
    return y_pred

y_pred = Gaussian_NB(X_train, y_train, X_test)
```

Figure 3. Code from scratch (Gaussian NB)

```
def split_classes(X_train, y_train):
    C1_count = (y_train == 1).sum()
    C2_count = (y_train == 0).sum()
    X_train_C1 = np.zeros([C1_count, X_train.shape[1]])
    X_train_C2 = np.zeros([C2_count, X_train.shape[1]])
    y_train_C1 = np.zeros([C1_count, 1])
    y_train_C2 = np.zeros([C2_count, 1])
    c1 = 0
    c2 = 0
    for i in range(len(y_train)):
        if y_train[i] == 1.:
            X_train_C1[c1] = X_train[i]
            y_train_C1[c1] = y_train[i]
            c1 = c1 + 1
        else:
            X_train_C2[c2] = X_train[i]
            y_train_C2[c2] = y_train[i]
            c2 = c2 + 1
    return X_train_C1, X_train_C2, y_train_C1, y_train_C2
```

Figure 4. Subroutine that divides data into two classes

(2) K-nearest Neighbor

K Nearest Neighbor (KNN) is a simple algorithm that stores all the available cases and classifies the new data or case based on a similarity measure. It is mostly used to classifies a data point based on how its neighbors are classified. The construction of KNN model can be simplified into three steps: calculating Euclidean distance according to eq.5, sifting out the nearest neighbors (number of nearest neighbors, K=1,3,5,7, …) and making predictions.

$$Euclidean\ Disctance = \sqrt{\sum_{i}^{N} (x1_i - x2_i)^2} \quad \text{(eq.6)}$$

The scratched code is shown in **figure 5** including comparison of various k values and normalization of training data. **Figure 6** compares the accuracy obtained from different K values. It is shown that the highest accuracy occurred when k equals to 5, indicating the most applicable k value. To eliminate redundancy and increase the integrity, normalizing the training data is requisite. Normalization of the preprocessed data comprises (1) subtraction of the average value from each data and (2) division of each data by their standard deviation.

```
#%% KNN
K_neighbors = 5
X_train = ( X_train - np.mean(X_train, axis=0) ) / np.std(X_train, axis=0, ddof=1)
X_test = ( X_test - np.mean(X_test, axis=0) ) / np.std(X_test, axis=0, ddof=1)

def KNN(X_train, y_train, X_test, k):
    y_pred = np.zeros(len(X_test))
    distances = pd.DataFrame(np.zeros([len(X_train),2]), columns=['y','dist'])
    distances['y'] = y_train
    for i in range(len(X_test)):
        for j in range(len(X_train)):
            distances['dist'][j] = np.sqrt(np.sum((X_test[i]-X_train[j])**2))
        dist = distances.sort_values(by=['dist'])[:k]
        y_pred[i] = dist['y'].value_counts().idxmax()
    return y_pred

```

Figure 5. Code from scratch (KNN)

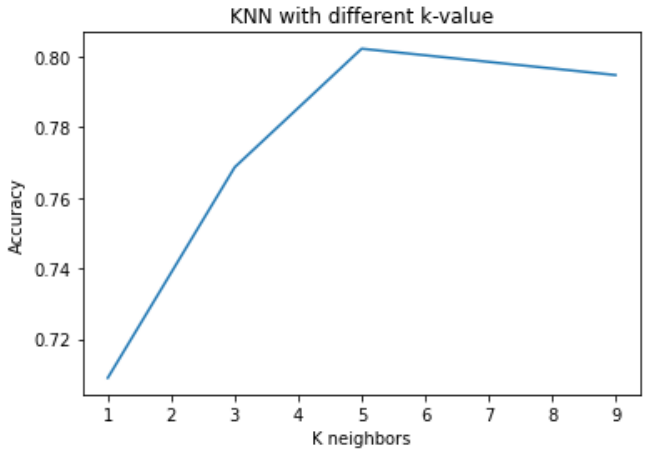


Figure 6. Accuracy versus k-neighbors

III. Results and Discussion

To sum up, the best results were obtained by training the preprocessed data, Combination A, using Gaussian Naïve Bayes classifier. The accuracy of prediction gained from two KNN model (with k=3 and k=5) were mostly lower than that gained from Naïve Bayes model since KNN is sensitive to noise of certain features whereas NBC takes all the information of features into consideration.

Due to our eagerness to single out the features or modifications of features that affect the result the most, we targeted at Combination B at first and then and extended to a wide variety of modified datasets. The results were shown in **table 2**. In **table 2**, the training dataset, Combination B0, took the Name feature into account, while Combination B1 aborted this feature. The accuracies of prediction attained from these two datasets were identical, implying that the Name feature did not affect the result. The difference between datasets B1 and B2 is that the Age feature was labelled by groups in B1 whereas the Age feature was trained by their original value in B2. In comparison, the prediction is more accurate if we divide Age into groups and label in accordance with the groups. On

the contrary, for the Fare feature, higher accuracy is reached if the original value of this feature is retained.

As the standard deviation of the Fare feature is larger than the Age feature, the Fare feature is more discrete and informative. Hence, the Fare feature do not need additional partitioning. For the dataset B5, we labelled the Pclass feature depending on the Fare feature. The equivalent results calculated from B0 and B5 attested that the Fare feature originally contains abundant information so that the methods to label this feature do not impact the accuracy of prediction.

Though the results trained from datasets B0, B1 and B5 were equal, we firmly hold the opinion that the dataset B5 is better for preprocessing, training and prediction for the reason that B5 provides more information and the Pclass feature was systematically labelled. This preprocessing method might be more adaptive to other unknown datasets and might realize more accurate predictions.

Table 2. The Results from different combination of features.

Features	A	B	C	D	E	F
Pclass	observed	observed	observed	observed	Depend on Fare	Depend on Fare
Name	A->1, B->2 …	A->1, B->2 …	-	-	-	-
Sex	Male=0 Female=1	Male=0 Female=1	Male=0 Female=1	Male=0 Female=1	Male=0 Female=1	Male=0 Female=1
Age	Mean age	Age group (1-6)	Age group (1-6)	Age group (1-6)	Age group (1-6)	Age group (1-6)
SibSp	+	-	-	+	+	+
Parch	+	-	-	+	+	+
Relatives	-	+	+	+	+	-
Ticket	-	-	-	-	-	-
Fare	+	+	Fare band (4)	Fare band (4)	Fare per person	Fare per person
Cabin	-	-	-	-	-	-
Embarked	S:1, C:2, Q:3	S:1, C:2, Q:3	S:1, C:2, Q:3	S:1, C:2, Q:3	S:1, C:2, Q:3	S:1, C:2, Q:3
Not alone	-	+	+	+	+	-
KNN (k=3)	0.79640	0.76047	0.76646	0.76047	0.75449	0.73053
KNN (k=5)	0.80239	0.77844	0.76646	0.76646	0.73056	0.75449
Naïve Bayes	0.81437	0.82634	0.77844	0.74850	0.77245	0.76047

Features	B0	B 1	B2	B 3	B 4	B5
Pclass	observed	observed	observed	observed	observed	Depend on Fare
Name	+	--	--	+	+	+
Age	group(1-6)	group(1-6)	+	group(1-6)	group(1-6)	group(1-6)
Fare	+	+	+	Fare Band (4)	Fare per person	+
Naïve Bayes	0.83	0.83	0.81	0.79	0.80	0.83

IV. CONCLUSIONS

By comparing the results obtained from two models and a variety of ways to interpret the features, the methods to combine and label the features were found to largely affect the ultimate results. Therefore, in order to achieve better accuracy of prediction, further investigation on the approaches to manipulate and label the features is in need.

REFERENCE

[1] https://vitalflux.com/correlation-heatmap-with-seaborn-pandas/

[2] https://betterprogramming.pub/titanic-survival-prediction-using-machine-learning-4c5ff1e3fa16

[3] https://www.kaggle.com/nadintamer/titanic-survival-predictions-beginner