Predicting Survivors of Titanic Shipwreck

Heath Thompson

Computer Science

CS 477 Spring 2021

05/03/2021

Introduction

This project focuses on determining survivors of the infamous Titanic shipwreck. There are three datasets provided by https://www.kaggle.com/c/titanic/data : train.csv, test.csv, and gender_submission.csv. The test dataset holds the same column headers as the train dataset but lacks the “Survived” column. This column is the target and is what we wish the model to predict accurately and precisely. The rest of the columns include information about each of the passengers on the ship. Some of these include name, age, class, fare, and whether the passenger had children, a spouse, or parents. Each row in the datasets represents a single passenger on the ship.

Approach

Because we would like to predict if a passenger survived or died after the wreck, the passengers can be classified as such using logistic regression. I chose this approach because it was one we discussed in class and was the one I felt most confident with implementing in the project. Before I can apply logistic regression, the data must be prepared by removing/filling null valued cells in the data, removing columns not applicable to the problem, and finally converting all qualitative data to quantitative data. Preprocess->train->test->tune->retest.

Progress

So far, I have the data cleaned and ready to be used in logistic regression. I received an error when trying to fit the logistic regression model to the data. It mentioned increasing the max_iter variable in the function, which worked, but I am still trying to understand what this is telling me. Currently, I have a model that can be used on the data, but it is not as accurate as planned. I know I have a long way to go, but the results oh my model are provided below.

Experimental Results

Below is a figure of the training and testing accuracies of my current model. As of right now, the model could use a lot of work. Currently, it has just been fitted to the datasets and nothing more.

```
print("Training Accuracy: ", sum(y_train_prediction==y_train)/y_train.shape[0])
print("Testing Accuracy: ", sum(y_test_prediction==y_test)/y_test.shape[0])
```

```
Training Accuracy:  0.8019662921348315
Testing Accuracy:  0.8044692737430168
```

Below is a classification report generated from the results of my model compared to what is true. Again, the model has just been fitted and nothing more.

```
y_train_prediction = model.predict(X_train)
y_test_prediction = model.predict(X_test)

print(classification_report(y_test, y_test_prediction))
```

```
              precision    recall  f1-score   support

           0       0.84      0.85      0.84       110
           1       0.75      0.74      0.74        69

    accuracy                           0.80       179
   macro avg       0.79      0.79      0.79       179
weighted avg       0.80      0.80      0.80       179
```