writing: 10/10

EE 510: Machine Learning Theory & Algorithms **Spring 2020**

Mini Project 1

Due: April 24, 2020, 11:59PM PT

Student Name: Andrew Huffman *Instructor Name: John Lipor*

Problem Description 5/5

In this project I worked with the Titanic dataset [1] from Kaggle. This dataset includes a training set and a test set. The training set includes passenger ID numbers with binary survival labels. For each passenger, there are 10 features: ticket class, sex, age, number of siblings/spouses abord the Titanic, number of parents/children aboard the Titanic, ticket number, passenger fare, cabin number, and port of embarkation. The test set consists of the same 10 features and passenger ID numbers but no labels. My goal was to train linear classifiers to predict which passengers in the test set survived. I implemented two linear classifiers, ridge regression and logistic regression, to predict passenger survival.

Exploratory Data Analysis (EDA) 16/20

I found Shubham Simar Tomar's tutorial [2] helpful in getting started with exploratory data analysis. I explored correlations between features and survival by looking at the percentage of survival for each feature. The strongest trends I found were between sex and survival and ticket class and survival, as shown in Figures 1 and 2. Overall, nearly 75% percent of female passengers survived while only around 19% of male passengers survived. There was also a disparity in survival rates across ticket classes—around 63% for first class, 47% for second class, and 24% for third class.

always use .eps for latex plots →

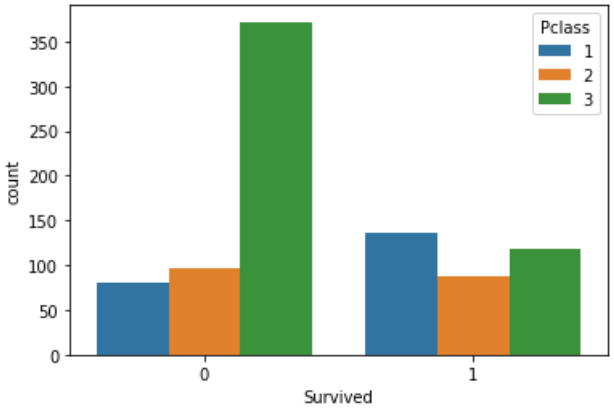

Figure 1: Survival count by sex

would've liked to see more data here to back up the claims you make in Approach section

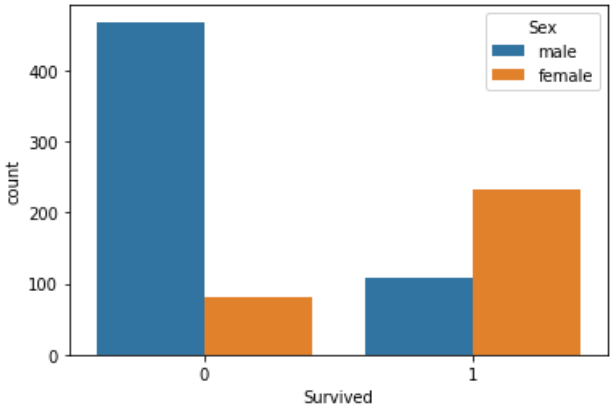

Figure 2: Survival count by ticket class

After some feature engineering and converting all used features to categorical representation by integers, I checked the correlation between features. The correlation matrix (with correlation coefficients rounded to one decimal place) is shown is figure 3.

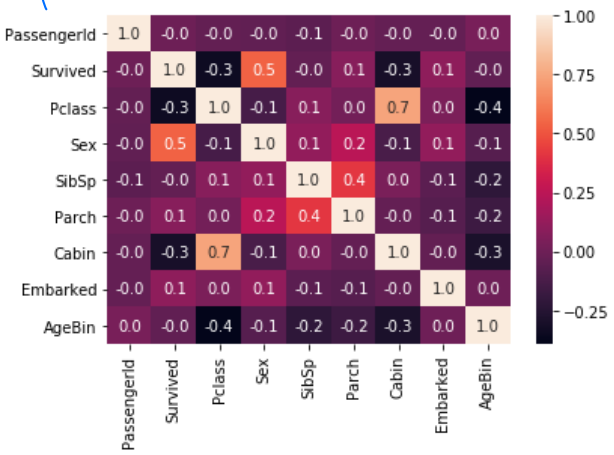

Figure 3: Feature correlation matrix

Challenges

I encountered a few challenges in trying to achieve useful prediction using linear classification on this dataset. The dataset was somewhat challenging because it had a lot of missing values for the Cabin feature (687 missing) and Age feature (177 missing). However, based on my analysis, these features do not seem to be crucial to performance of the classifiers I implemented. Another challenge was model performance, likely because this dataset is not linearly separable, and so the classifiers I implemented are not well-suited for classifying this dataset.

Approach 30/30

Data wrangling and feature engineering

Of the 10 features, I omitted three, name, ticket number, and passenger fare, in all analyses. There did not seem to be any value provided by the name feature. I did not find the last names to have a significant correlation with survival rate. Less than 3% of the titles associated with passenger name were something other than "Mr.", "Mrs.", "Miss", and "Master". Whatever relationship these titles might reflect with survival rate would actually be owed to the gender associated each of these titles. So, titles did not seem very informative. There seemed to be no discernible patterns in the ticket number feature, and I assumed it would not be helpful in predicting passenger survival. The passenger fare feature showed a high degree of variation with higher fares inconsistently corresponding to higher class tickets. As shown in Figure 3, ticket class has a relatively high correlation with survival. Given the erratic relationship between passenger fare and ticket class and the unlikeliness of such ticket fares being inherently indicative of access to lifeboats, it seemed reasonable to omit the ticket fare feature.

Out of 891 training examples 2 were missing port of embarkation data, 177 were missing age data, and 687 were missing cabin data. For age and cabin this a significant portion of training examples. I tried two methods for addressing missing values. For this portion of my analysis I took some guidance from a Kaggle tutorial [3] on the Titanic dataset.

In the first approach, I used the mode value to impute the 2 missing values for port of embarkation. Relative to other features, the correlation between age and ticket class was high so I imputed age value as the median age for the associated ticket class. Then, I grouped the ages into bins thinking that it would be more likely for a range of ages (e.g. 20-30) to have significance than for a single age value to have significance. For the cabin feature, I selected only the first letter from the given cabin designation, which corresponds to the deck that cabin was on, and assigned "U" for unknown to the examples for which the cabin information was missing.

In the second approach, I imputed the missing port of embarkment values as described above, but I deleted the age and cabin features. The correlation between survival and age indicates that age should not affect survival predictions significantly. Given that and a lack of an accurate method to impute missing ages, it seemed reasonable to omit age altogether in the model. The correlation between cabin name and survival is somewhat high (-0.3), but I had no good way of guessing which of 10 decks should be assigned to the passengers for which the cabin identification was missing. Notably, cabin name and ticket class have a high correlation (0.7), and so perhaps ticket class feature could be used to glean most of the information that could be conveyed by the cabin feature. Considering this and large uncertainty involved in attempting to guess the appropriate cabin decks, it seemed reasonable to omit the cabin feature from the data.

For both approaches to handling missing values, after that step, I converted all features to categorical, integer-labeled data. The only feature used that was not originally in categorical form was age. As described above, it seemed age ranges would better predict survival than continuously valued ages.

Model selection

I implemented two linear classifiers, ridge regression and logistic regression using stochastic gradient descent. For both classifiers I split the training data into training and validation sets to study the effect of hyper-parameters and number of iterations for the logistic classifier. I also tried using an offset and not using an offset for each classifier. I tried a range of regularization parameters, λ, for the ridge regression classifier. The validation results were not especially sensitive to the choice of λ (within a reasonable range). I chose to use $\lambda =$ 1e-3. I tried a range of values for the learning rate, μ, and training iterations for use in logistic regression with stochastic gradient descent. I selected $\mu =$ 1e-3 and 100 training iterations. Generally, the validation error flattened out (relatively) well before 100 iterations, but there was no sign of overfitting in the validation error. Figure 4 shows the validation error for 100 iterations.

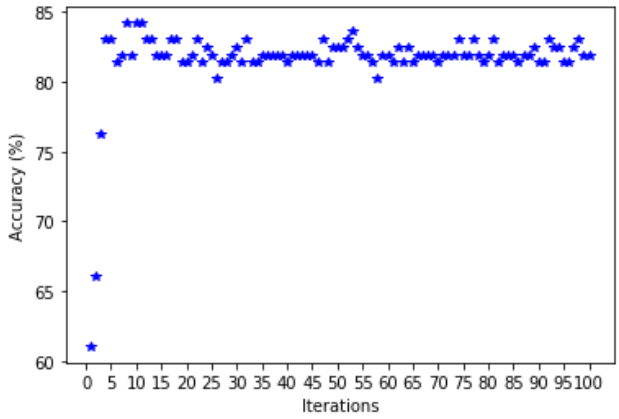

Figure 4: Logistic regression validation error

I did not find the use of an offset to be particularly helpful in this classification problem. The test accuracy was not significantly different when using an offset than when not using an offset.

Evaluation and Summary

Using the ridge regression classifier with λ = 1e-3 yielded validation accuracy of approximately 81% and test accuracy of approximately 77%.Using logistic regression with μ = 1e-3 yielded validation accuracy of approximately 81% and test accuracy of approximately 78%.The confusion matrices from the validation sets for the ridge regression and logistic regression classifiers are shown in Tables 1 and 2, respectively. The confusion matrices indicate that the models perform better when classifying an input correspond to a passenger that did not survive. Only around 38% of the passengers in the training set survived. Perhaps this imbalance between survived labels and not survived labels in the data contributed to the models bias toward predicting that passengers did not survive.

		Actual Labels	
		0	1
Predicted Labels	0	101	13
	1	20	43

Figure 5: Ridge regression confusion matrix

		Actual Labels	
		0	1
Predicted Labels	0	101	13
	1	20	43

Figure 6: Logistic regression confusion matrix

Based on examining survival counts vs category for each feature, it seemed that sex and ticket class were the strongest predictors of survival. Ticket number, passenger fare, and name seemed to contribute very little predictive strength. From the relatively few provided cabin identifications, it seems that the cabin deck might have been useful in predicting survival, but, unfortunately, that feature was missing for most of the passengers. The two classifiers essentially performed equally well, both predicting survival in the 77-78% accuracy range. The main limitations in predicting accuracy were using the linear classifiers and not having access to particularly insightful features. The structure of this dataset seems to be too complex for linear classifiers to handle well.

What I Learned 5|5

The main skills I learned through this project were how to perform exploratory data analysis, how to deal with missing data and how to perform basic feature engineering. The amount of missing data in the Titanic dataset and the variety in its features required these skills to be practiced. It was valuable to work with a dataset that was somewhat messy, in contrast to a dataset like MNIST in which the data are well curated and ready to be fed into a classifier. Also, this project provided a good opportunity for practicing methods for exploring relationships between several features and class of interest. I had not used tools like seaborn count plots and category plot prior to this project, and these tools seem like they are quite useful.

References

[1] (2018) Titanic: Machine learning from disaster. [Online]. Available: https://www.kaggle.com/c/titanic/overview

[2] (2016) A comprehensive introduction to data wrangling. [Online]. Available: https://www.springboard.com/blog/data-wrangling/

[3] (2020) Titanic tutorial for beginners, part 1-3. [Online]. Available: https://www.kaggle.com/kernelgenerator/titanic-tutorial-for-beginners-part-1