CS146: Machine Learning

Homework 1

Chaojie Feng
Oct.18, 2018

1. Problem 1

Please check CCLE for answers

2. Problem 2: Entropy and Information

(a) Problem 2(a)

Since $\frac{p_k}{p_k+n_k}$ is constant for all k, $\frac{p_k}{p_k+n_k} = \frac{p}{p+n}$

$$Gain = H[S] - \sum_{i=1}^{K} \frac{1}{K} H[S_k]$$

Where:

$$\sum_{i=1}^{K} \frac{1}{K} H[S_k] = -K * \frac{1}{K} * \frac{p_k}{p_k+n_k} \log\left(\frac{p_k}{p_k+n_k}\right) - K * \frac{1}{K} * \frac{n_k}{p_k+n_k} \log\left(\frac{n_k}{p_k+n_k}\right)$$

$$= -\left(\frac{p_k}{p_k+n_k}\right) \log\left(\frac{p_k}{p_k+n_k}\right) - \left(\frac{n_k}{p_k+n_k}\right) \log\left(\frac{n_k}{p_k+n_k}\right)$$

$$= -\left(\frac{p}{p+n}\right) \log\left(\frac{p}{p+n}\right) - \left(\frac{p}{p+n}\right) \log\left(\frac{n}{p+n}\right)$$

$$= H[S]$$

As a result, Gain = H[s] – H[s] = 0

3. K-Nearest Neighbor and Cross-Validation

(a) Problem 3(a)

To minimize training set error, K is selected to be 1. The resulting training error is 0 because the nearest neighbor for every observation is its own.

(b) Problem 3(b)

If K is too large, test data will absorb not only all error data near to itself, but also correctly labeled data near to it. In voting process, the test data will not be voted correctly for its label, which causes generalization or underfitting. If K is too small, then although the performance of training might be good, it causes over-fitting because the nearest neighbors might not be representative for the whole dataset.

(c) Problem 3(c)

K = 5 or 7. The resulting error is 4/14.

4. Programming Exercise

(a) Problem 4(a)

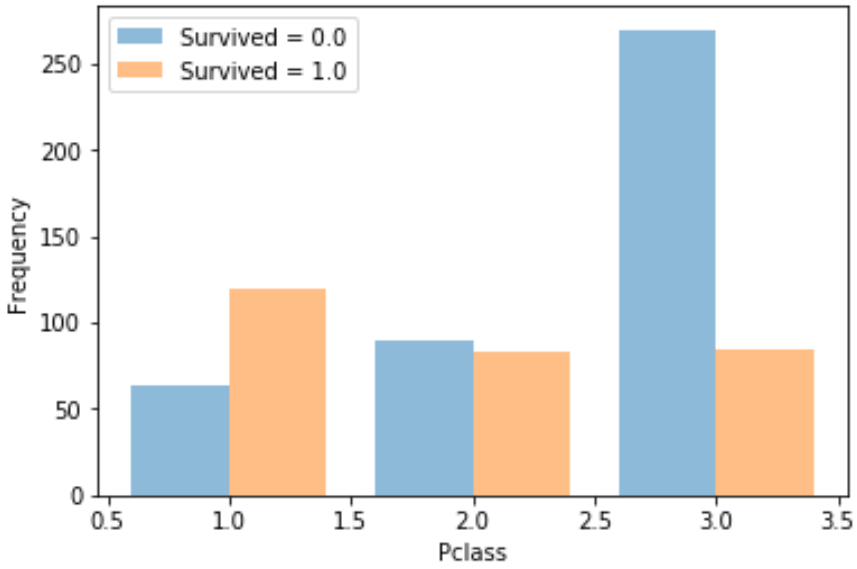

Figure 1. Ticket Class

The first feature is ticket class. 1 is for 1st class, 2 is for 2nd class and 3 is for 3rd class. It can be concluded that survival rate is decreasing as ticket class downgrades, and survival rate for 3rd class is significantly lower than 1st and 2nd class. Also, among all the passengers not survived, 3rd class passengers contributed most. However, survived passengers are distributed almost evenly among classes.

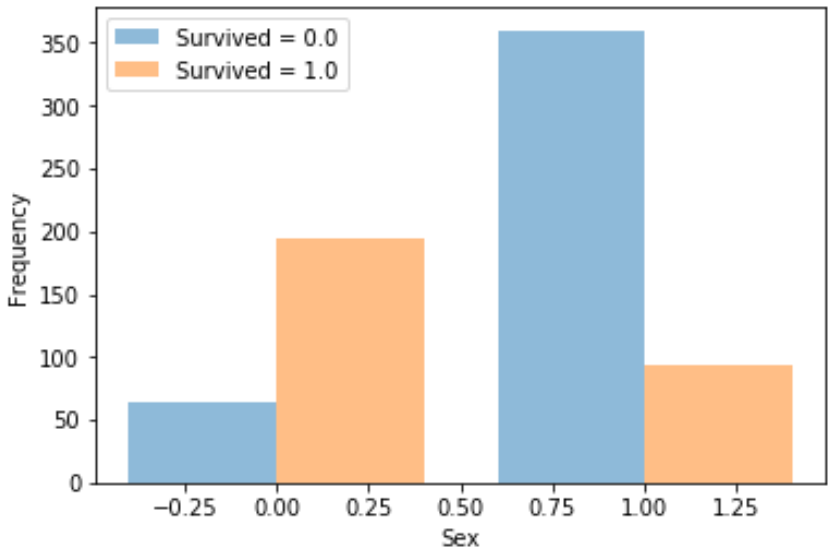

Figure 2. Sex

The second feature is sex. Female is much more likely to survive as male. Also, we can tell that there are more male passengers boarding than female passengers.

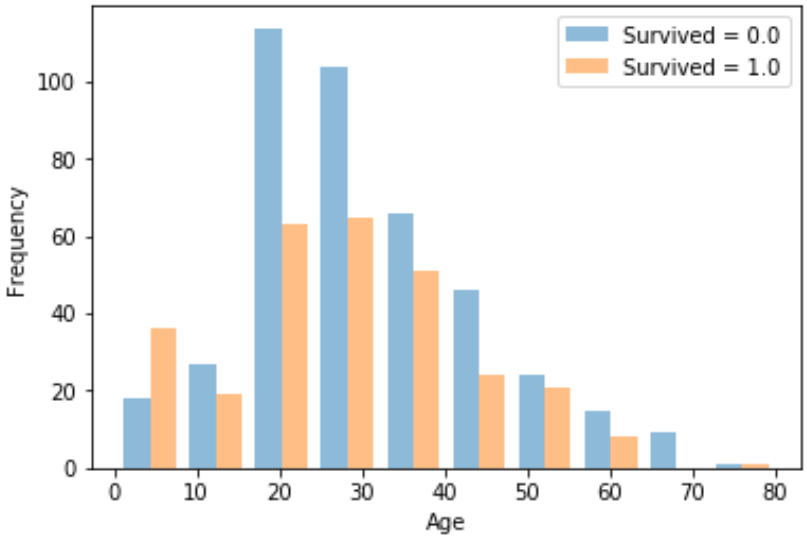

Figure 3. Age

The third feature is age. It is not surprising that young children are more likely to survive because they will have priority. For all other ages, passengers are less likely to survive.

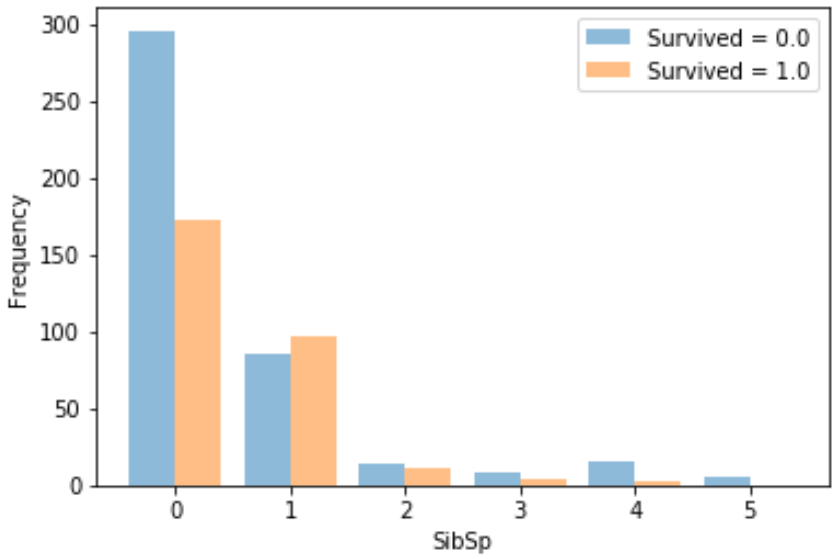

Figure 4. # of siblings and spouses

The fourth feature is # of siblings and spouses aboard. Obviously, as more siblings and spouses come with you, you might get help from them, which makes you more likely to survive.

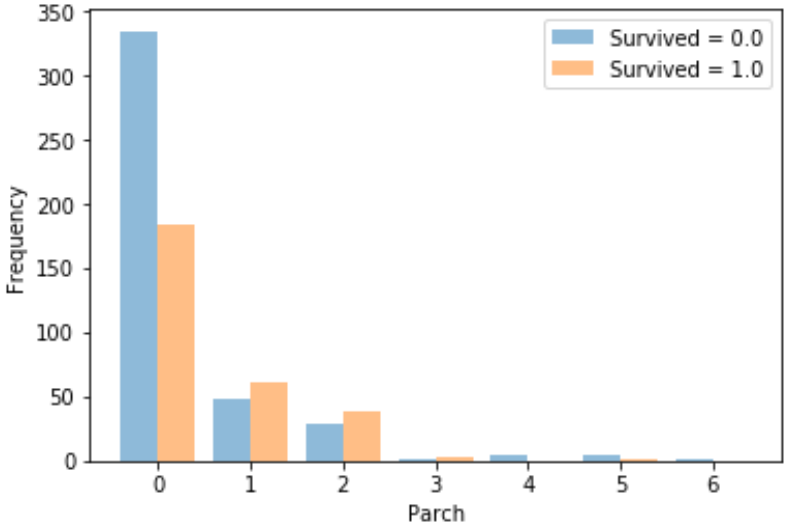

Figure 5. # of parents and children

The fifth feature is # of parents and children passenger brings aboard. Trend is pretty similar as previous case, where a single passenger is less likely to survive.

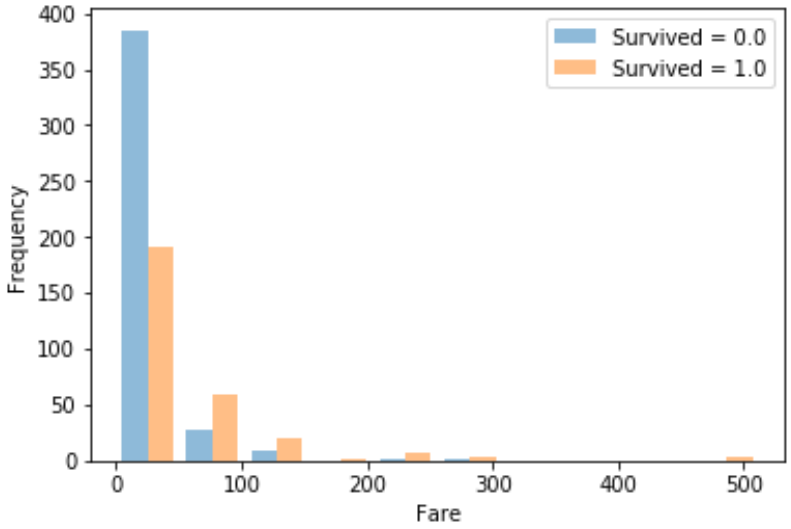

Figure 6. Fare

The sixth feature is fare. It makes sense that as passenger fare increases, the chance of surviving gets greater.

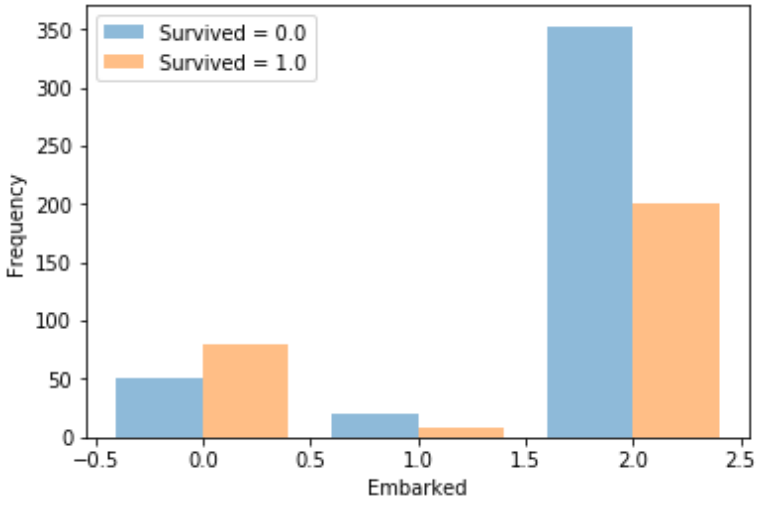

Figure 7. Embarked

The seventh feature is port of embarkation. For those who embark at Cherbourg, they are more likely to survive than those who embarked at other two location.

(b) The output of error is 0.485 as suggested in problem description

```
    ### ========== TODO : START ========== ###
    # part b: set self.probabilities_ according to the training set
    n,d = X.shape
    dead = Counter(y)[0]
    surv = Counter(y)[1]


    self.probabilities_ = [dead/(surv+dead),surv/(surv+dead)]

    ### ========== TODO : END ========== ###

  ### ========== TODO : START ========== ###
  # part b: predict the class for each test example
  # hint: use np.random.choice (be careful of the parameters)
  n,d = X.shape
  y = np.random.choice([0,1],size = n, p = self.probabilities_)

  ### ========== TODO : END ========== ###
```

Figure 8. Random Classifier

(c) The training error is 0.014 for decision tree classifier using entropy criterion.

```
### ========== TODO : START ========== ###
# part c: evaluate training error of Decision Tree classifier
# use criterion of "entropy" for Information gain
print('Classifying using Decision Tree...')
clf = DecisionTreeClassifier(criterion = "entropy")
clf.fit(X,y)
y_pred = clf.predict(X)
train_error = 1 - metrics.accuracy_score(y, y_pred, normalize=True)
print('\t-- training error: %.3f' % train_error)

### ========== TODO : END ========== ###
```

Figure 9. Decision Tree Classifier

(d) When k = 3, the training error is 0.167. When k = 5, the training error is 0.201. When k = 7, the training error is 0.240. The section of code is provided below.

```
### ========== TODO : START ========== ###
# part d: evaluate training error of k-Nearest Neighbors classifier
# use k = 3, 5, 7 for n_neighbors

print('Classifying using k-Nearest Neighbors...')

# n can be chosen in {3,5,7}
clf = KNeighborsClassifier(n_neighbors = 7)
clf.fit(X,y)
y_pred = clf.predict(X)
train_error = 1 - metrics.accuracy_score(y, y_pred, normalize=True)
print('\t-- training error: %.3f' % train_error)

### ========== TODO : END ========== ###
```

Figure 10. KNN Classifier

(e) Average training and test error of each classifier

	Majority Vote	Random	Decision Tree	KNN
Train error	0.404	0.489	0.012	0.213
Test error	0.407	0.486	0.241	0.315

(f) Average error is 0.303. It can be seen that the error is very high when K is 1. This is because when K is low, the model tends to overfit the dataset. The best K is 7.

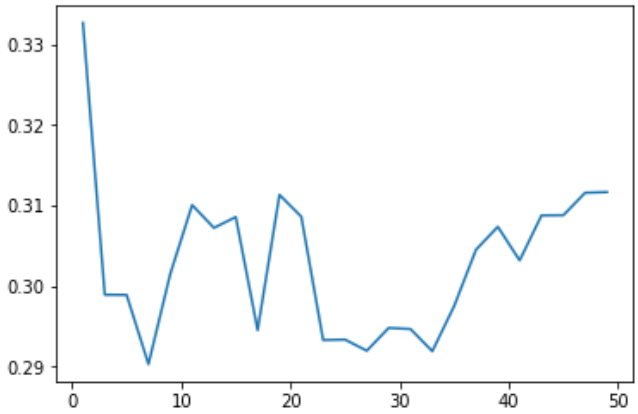

Figure 11. Average error vs. K

(g) Clearly, as depth limit increases, the classifier tends to overfit the data as training error is decreasing but testing error still remain at a high level. As a result, we need to choose the best classifier based on testing error. In this situation, the best depth is 3.

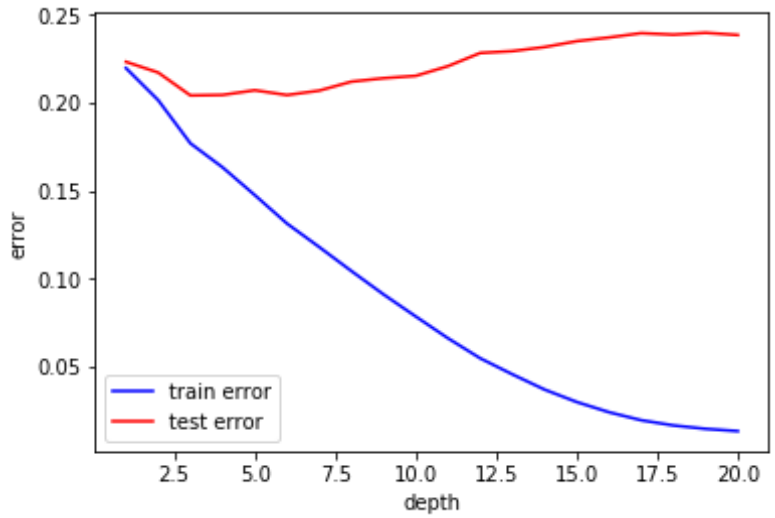

Figure 12. error for different depth using decision tree classifier

(h) The learning curve is plotted using average error over 100 trials, with specific random state to control the dataset. As can be seen from figure 13 and 14, both test errors for DT and KNN are decreasing, which shows the learning method is valid. From the figures we can also conclude that for this dataset decision tree is a better model than KNN because the average testing error is lower and the variance (gap between training and testing error) is smaller.

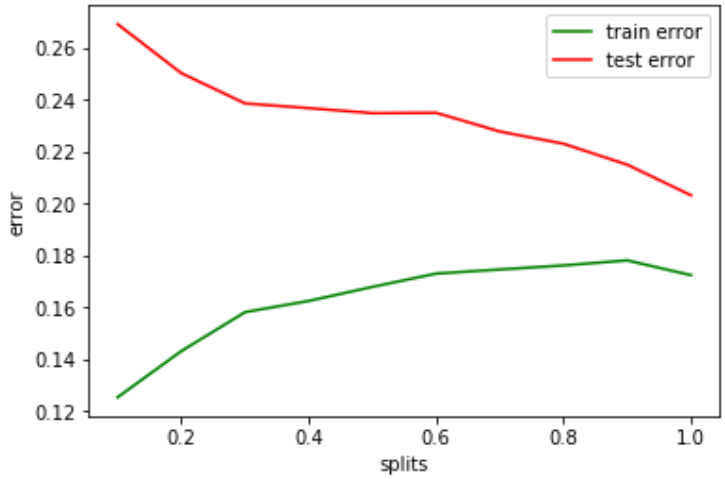

Figure 13. training and test error for decision tree over splits

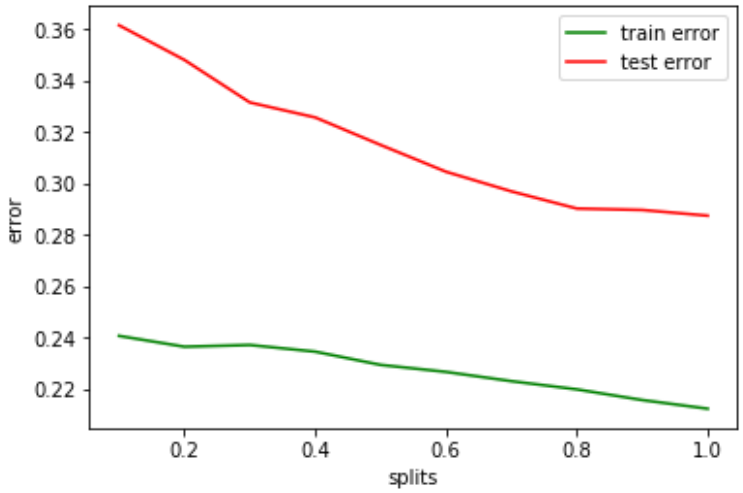

Figure 14. training and testing error for KNN over splits