Nisaruj Rattanaaram 6031033521

Homework 1 Clustering and Regression

Source code for all tasks was uploaded to MyCourseVille and
https://colab.research.google.com/drive/1_EX5wlO0BMH05-NAPlap0wE5vdg0fdNc

T1. Prove that $\nabla_A tr(AB) = B^T$

Sol^n $\because$ $AB = \Sigma_m A_{i,m} B_{m,j}$

$$\nabla_A tr\left(\begin{bmatrix} \Sigma_m A_{1,m}B_{m,1} & \cdot & \cdot & \cdot \\ \cdot & \Sigma_m A_{2,m}B_{m,2} & \ddots & \\ \cdot & & & \\ \cdot & \cdot & \cdot & \Sigma_m A_{m,m}B_{m,m} \end{bmatrix}\right)$$

$$= \nabla_A \left(\Sigma_m A_{1,m}B_{m,1} + \Sigma_m A_{2,m}B_{m,2} + \dots + \Sigma_m A_{m,m}B_{m,m}\right)$$

$$= \nabla_A (\Sigma_m A_{1,m}B_{m,1}) + \nabla_A(\Sigma_m A_{2,m}B_{m,2}) + \dots + \nabla_A(\Sigma_m A_{m,m}B_{m,m})$$

$$= \begin{bmatrix} B_{1,1} & B_{2,1} & \cdots & B_{m,1} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & \cdots & 0 \\ B_{1,2} & B_{2,2} & \cdots & B_{m,2} \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & & 0 \end{bmatrix} + \dots + \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ B_{1,m} & B_{2,m} & \cdots & B_{m,m} \end{bmatrix}$$

$$= \begin{bmatrix} B_{1,1} & B_{2,1} & \cdots & B_{m,1} \\ B_{1,2} & B_{2,2} & \cdots & B_{m,2} \\ \vdots & & \cdots & \vdots \\ B_{1,m} & B_{2,m} & \cdots & B_{m,m} \end{bmatrix} = B^T \quad \square$$

T2. Prove that $\nabla_{A^T} f(A) = (\nabla_A f(A))^T$

Sol^n $\nabla_{A^T} f(A) = \left[(\nabla_{A^T} f(A))^T\right]^T = \left[\begin{bmatrix} \frac{\partial f}{\partial A_{1,1}} & \frac{\partial f}{\partial A_{2,1}} & \cdots & \frac{\partial f}{\partial A_{n,1}} \\ \frac{\partial f}{\partial A_{1,2}} & \frac{\partial f}{\partial A_{2,2}} & \cdots & \frac{\partial f}{\partial A_{n,2}} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f}{\partial A_{1,m}} & \frac{\partial f}{\partial A_{2,m}} & \cdots & \frac{\partial f}{\partial A_{n,m}} \end{bmatrix}^T\right]^T$

$$= \begin{bmatrix} \frac{\partial f}{\partial A_{1,1}} & \frac{\partial f}{\partial A_{1,2}} & \cdots & \frac{\partial f}{\partial A_{1,m}} \\ \frac{\partial f}{\partial A_{2,1}} & \frac{\partial f}{\partial A_{2,2}} & \cdots & \frac{\partial f}{\partial A_{2,m}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f}{\partial A_{n,1}} & \frac{\partial f}{\partial A_{n,2}} & \cdots & \frac{\partial f}{\partial A_{n,m}} \end{bmatrix}^T$$

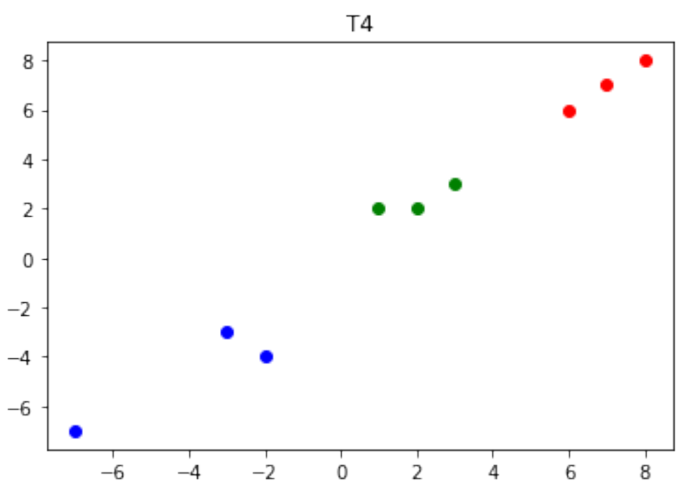
$$= (\nabla_A f(A))^T \quad \square$$

T4. If the starting points are (3,3), (2,2), and (-3,-3). Describe each assign and update step. What are the points assigned? What are the updated centroids? You may do this calculation by hand or write a program to do it.

```
Iter 0
Centroids: ( 3 , 3 ) ( 2 , 2 ) ( -3 , -3 )
Assign Step
Cluster 0 :
  ( 3 , 3 ) ( 8 , 8 ) ( 6 , 6 ) ( 7 , 7 )
Cluster 1 :
  ( 1 , 2 ) ( 2 , 2 )
Cluster 2 :
  ( -3 , -3 ) ( -2 , -4 ) ( -7 , -7 )

Iter 1
Centroids: ( 6.0 , 6.0 ) ( 1.5 , 2.0 ) ( -4.0 , -4.666666666666667 )
Assign Step
Cluster 0 :
  ( 8 , 8 ) ( 6 , 6 ) ( 7 , 7 )
Cluster 1 :
  ( 1 , 2 ) ( 3 , 3 ) ( 2 , 2 )
Cluster 2 :
  ( -3 , -3 ) ( -2 , -4 ) ( -7 , -7 )

Iter 2
Centroids: ( 7.0 , 7.0 ) ( 2.0 , 2.3333333333333335 ) ( -4.0 , -
4.666666666666667 )
Assign Step
Cluster 0 :
  ( 8 , 8 ) ( 6 , 6 ) ( 7 , 7 )
Cluster 1 :
  ( 1 , 2 ) ( 3 , 3 ) ( 2 , 2 )
Cluster 2 :
  ( -3 , -3 ) ( -2 , -4 ) ( -7 , -7 )
```

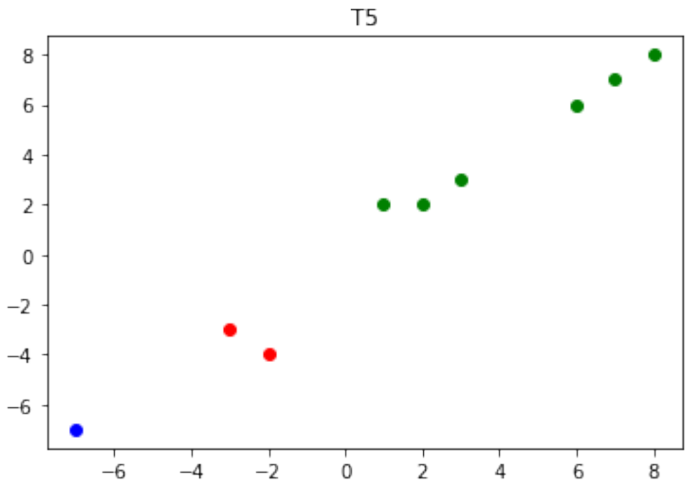


T5. If the starting points are (-3,-3), (2,2), and (-7,-7), what happens?

```
Iter 0
Centroids: ( -3 , -3 ) ( 2 , 2 ) ( -7 , -7 )
Assign Step
Cluster 0 :
  ( -3 , -3 ) ( -2 , -4 )
Cluster 1 :
  ( 1 , 2 ) ( 3 , 3 ) ( 2 , 2 ) ( 8 , 8 ) ( 6 , 6 ) ( 7 , 7 )
Cluster 2 :
  ( -7 , -7 )

Iter 1
Centroids: ( -2.5 , -3.5 ) ( 4.5 , 4.666666666666667 ) ( -7.0 , -
7.0 )
Assign Step
Cluster 0 :
  ( -3 , -3 ) ( -2 , -4 )
Cluster 1 :
  ( 1 , 2 ) ( 3 , 3 ) ( 2 , 2 ) ( 8 , 8 ) ( 6 , 6 ) ( 7 , 7 )
Cluster 2 :
  ( -7 , -7 )

Iter 2
Centroids: ( -2.5 , -3.5 ) ( 4.5 , 4.666666666666667 ) ( -7.0 , -
7.0 ) Assign Step
Cluster 0 :
  ( -3 , -3 ) ( -2 , -4 )
Cluster 1 :
  ( 1 , 2 ) ( 3 , 3 ) ( 2 , 2 ) ( 8 , 8 ) ( 6 , 6 ) ( 7 , 7 )
Cluster 2 :
  ( -7 , -7 )
```



T6. Between the two starting set of points in the previous two questions, which one do you think is better? How would you measure the 'goodness' quality of a set of starting points?

In general, it is important to try different sets of starting points when doing k-means.

We can use explained variance to measure clustering quality.

$$explained\ variance = \frac{between\_cluster\_var}{all\_data\_var}$$

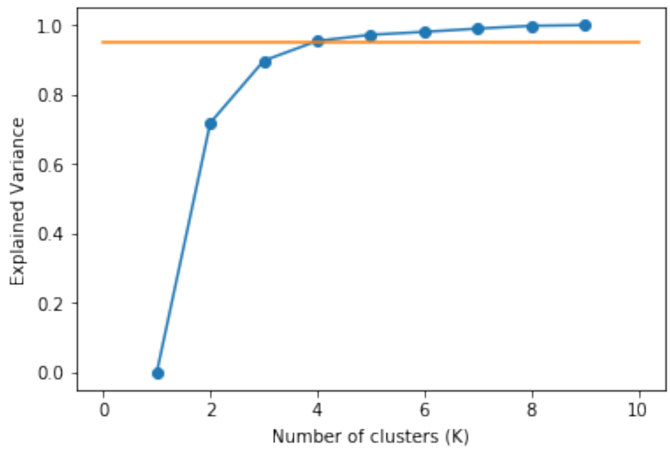
Since the explained variance of T4 is 0.93 while T5's is 0.81, we can say that T4 is better than T5.

OT1. What would be the best K for this question? Describe your reasoning.

To determine the best K for this question, we can use Elbow method though it isn't so accurate.

Elbow method chooses K where increasing complexity doesn't yield much in return. (i.e. minimal K that explains at least 95% of the all-data variance)

We find explained variance of each K by calculating average explained variance of K-mean clustering with different starting set of points.



From the result above, K=4 is the best K for this set of points.

T7. What is the median age of the training set? You can easily modify the age in the dataframe by `train["Age"] = train["Age"].fillna(train["Age"].median())`

Median = 28.0

T8. Some fields like 'Embarked' are categorical. They need to be converted to numbers first. We will represent S with 0, C with 1, and Q with 2. What is the mode of Embarked? Fill the missing values with the mode. You can set the value of Embarked easily with the following command: `train.loc[train["Embarked"] == "S", "Embarked"] = 0`

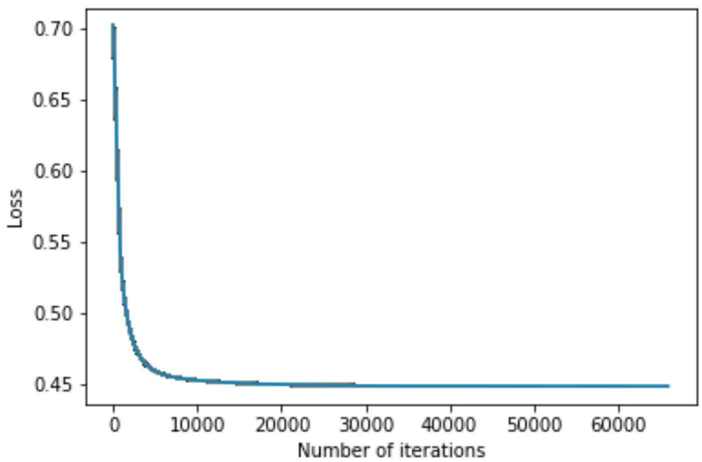
Mode = 0 (Southampton)

Do the same for Sex. (Male = 0, Female = 1)

Mode = 0 (Male)

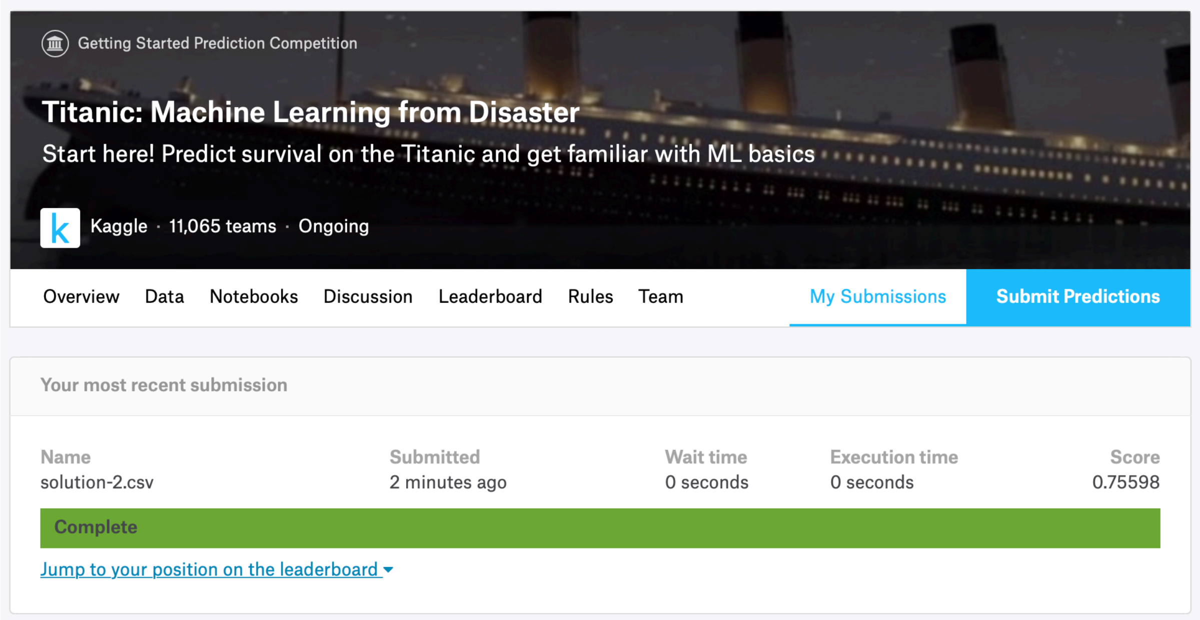
T9. Write a logistic regression classifier using gradient descent as learned in class. Use PClass, Sex, Age, and Embarked as input features.

weight ≈ [2.07070249, -1.19638948, 2.57579964, -0.03372561, 0.32077026]



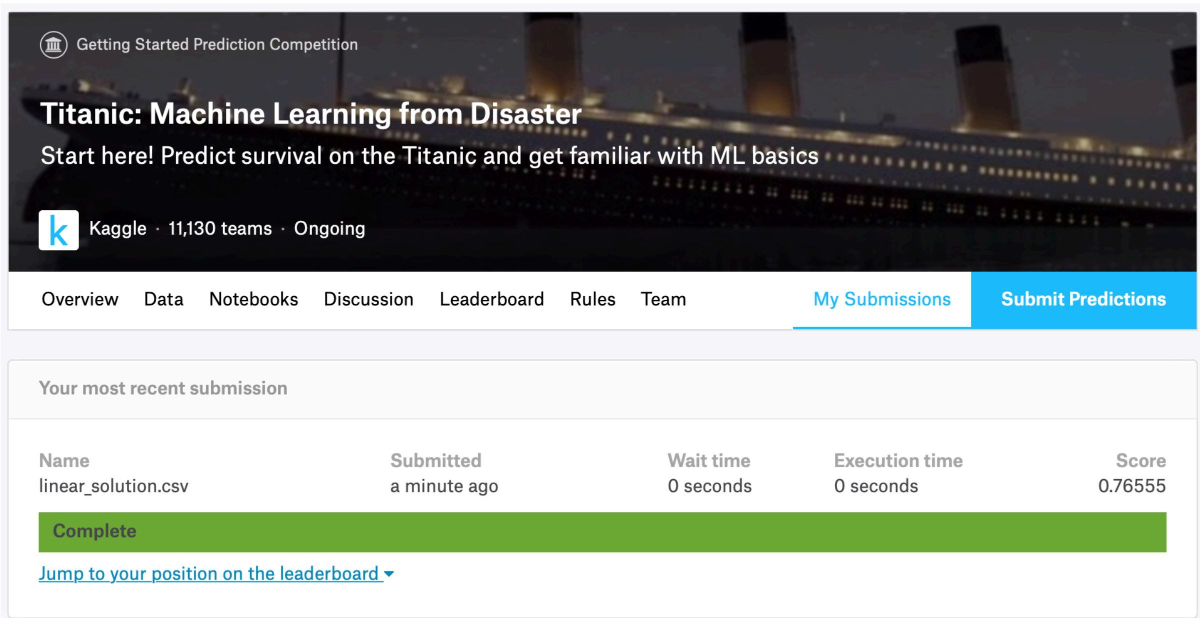
Log loss

T10. Submit a screenshot of your submission (with the scores). Upload your code to courseville.



OT2. We want to show that matrix inversion yields the same answer as the gradient descent method. However, there is no closed form solution for logistic regression. Thus, we will use normal linear regression instead. Re-do the Titanic task as a regression problem by using linear regression. Use the gradient descent method.

$$\text{weight} \approx [\ 0.60398843, -0.14915006, 0.52049079, -0.00288894\ , 0.05109822]$$



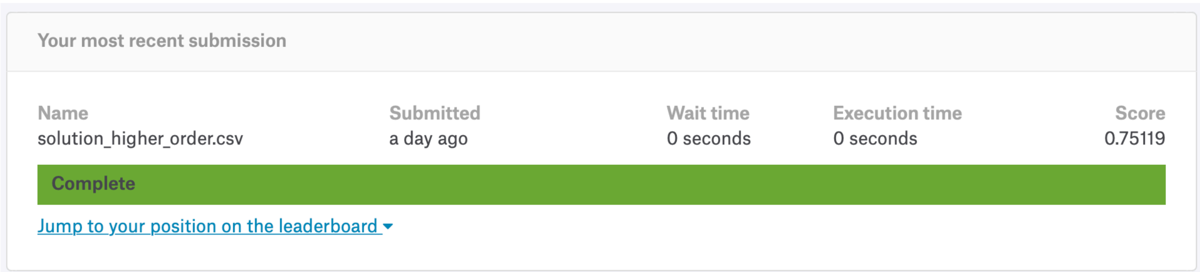
OT3. Now try using matrix inversion instead. However Are the weights learned from the two methods similar? Report the Mean Squared Errors (MSE) of the difference between the two weights.

weight = [0.776742, -0.18848969, 0.4908994, -0.00505977, 0.04907325]

The weights from two methods are similar.

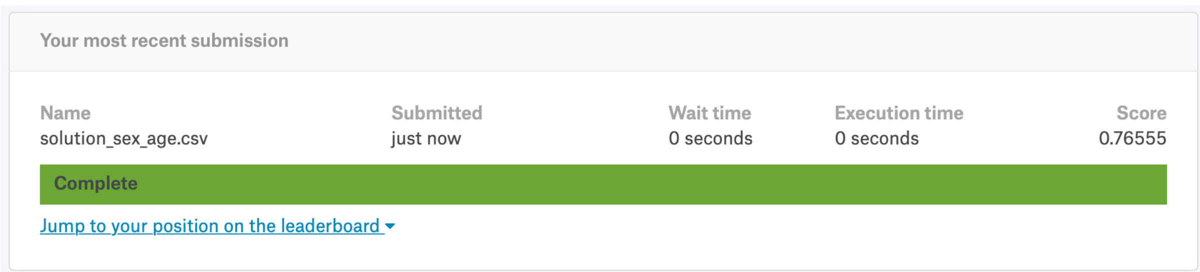
MSE = 0.006455173160960742

OT4. Try adding some higher order features to your training (x^2_1, x_1x_2,...). Does this model has better accuracy on the training set? How does it perform on the test set? (Use logistic regression)



We try to add higher order of Pclass but the score is a bit lower than the previous weight.

OT5. What happens if you reduce the amount of features to just Sex and Age? (Use logistic regression)



The score is equal to T10 (Logistic regression with gradient descent method). That means if we use too many features to train the model, it may drop the model's accuracy.