

**T1. What is the accuracy of Model A?**

ANS:

$$\text{Accuracy} = \text{Correct Pred} / \text{Total} = 70/100 = 0.7$$

**T2. Consider cats as ‘class 1’ (positive) and dogs as ‘class 0’ (negative), calculate the precision, recall, and F1.**

ANS:

$$TP = 40, TN = 30, FP = 20, FN = 10$$

$$\text{Precision} = TP / (TP + FP) = 40/60 = 0.67$$

$$\text{Recall} = TP / (TP + FN) = 40/50 = 0.8$$

$$F1 = 2 * (\text{Precision} * \text{recall}) / (\text{Precision} + \text{Recall}) = 0.729$$

**T3. Consider class cat as ‘class 0’ and class dog as ‘class 1’, calculate the precision, recall, and F1.**

ANS:

$$TP = 30, TN = 40, FP = 20, FN = 10$$

$$\text{Precision} = TP / (TP + FP) = 30/40 = 0.75$$

$$\text{Recall} = TP / (TP + FN) = 30/50 = 0.6$$

$$F1 = 2 * (\text{Precision} * \text{recall}) / (\text{Precision} + \text{Recall}) = 0.67$$

**T4. Now consider a lopsided population where there are 80% cats. What is the accuracy of Model A? Using dog as the positive class, what is the precision, recall, and F1? Explain how and why these numbers change (or does not change) from the previous questions.**

**ANS:**

Consider Actual 80 Cats and 20 Dogs, And the prediction rate still the same

$$\begin{aligned} \text{TP} &= 20 * 0.6 \text{ (rate of correctly identify dog)} = 12, \quad \text{TN} = 80 * 0.8 \text{ (rate of correctly identify cat)} = 64, \\ \text{FP} &= 80 - 64 = 16, \quad \text{FN} = 20 - 12 = 8 \end{aligned}$$

$$\text{Accuracy} = 0.76, \text{ Precision} = 0.43, \text{ Recall} = 0.6, \text{ F1} = 0.5$$

Accuracy increase as this model is better at identifying cat, increasing number of cats would make the weight better, resulting in more accuracy

Precision drop heavily from this imbalance, with more cats, the false positive overwhelm the calculation

Recall only relate to dog identification, which stay the same throughout the change,

**OT1. Consider the equations for accuracy and F1**

**ANS:**

Using given equation we can rewrite both equation for comparison as

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{F1} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

The only different in F1 and F1 would be the TP and TN

If  $\text{TP} == \text{TN}$  : Then  $\text{ACC} = \text{F1}$

If  $\text{TP} > \text{TN}$ : Then  $\text{ACC} < \text{F1}$

If  $\text{TP} < \text{TN}$ : Then  $\text{ACC} > \text{F1}$

So ACC will be equal to F1 when  $\text{TP} == \text{TN}$

ACC will be less than F1 when  $\text{TP} > \text{TN}$

ACC will be more than F1 when  $\text{TP} < \text{TN}$

The code relate to T5-T7, OT2 will be after this section's text answer

**T5. 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.**

ANS: Output cell in the code section

**T6. If the starting points are (-3,-3), (2,2), and (-7,-7), what happens?**

ANS: Output cell in the code section

**T7. 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?**

**ANS:**

From the result in T5 and T6 , without any visualization or standard method, T6 might look better from it's minimal iteration with only 2 iterations until convergence while T5 require 3 iterations, showing that the starting position is already close to a centroid

Using more standardize method to actually measure ‘goodness’ we can use Within-Cluster Sum of Squares (WCSS), Silhouette Score, Inter-Cluster Distance, Average Distance from Centroid. All of the measure are implement is code section, the result are below.

WCSS: Showing how each point are close to their centroid, the lower the better. T5 gives 29.33 and T6 gives 77.83, Showing that T6 Cluster is more fuzzy than T5

Silhouette Score: Showing how close it is to their centroid compare to other centroid, closer to 1 is better, -1 is worst. T5 gives 0.67 and T6 gives 0.5, Showing that T5 is giving more compact and distinct clustering.

Inter-Cluster Distance: Showing how far it is from other centroid, higher mean clusters are separated well. T5 and T6 give the same number at around 10.7-10.9

Average Distance from Centroid: Same as WCSS but interpret directly. T5 gives 1.47 and T6 gives 2.44, showing that on average T6 point is position farer than T5.

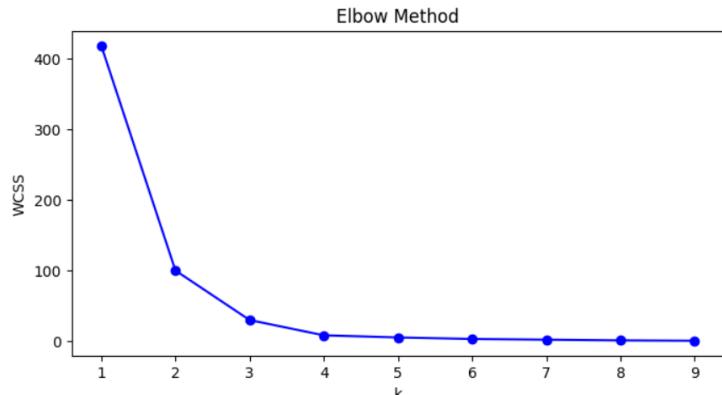
From standardize result, we can see that T5 starting centroid is significantly better than T6.

Ultimately, the starting centroid should also depend on the bussiness use cases by selecting the centroids that would show expected insight from bussiness views.

**OT2. What would be the best K for this question? Describe your reasoning.**

**ANS:**

We can determine the beat K using elbow method. The code for elbow method is in code section



We can select k to be around 3 or 4 for optimal k.

The code relate to T8-T11, OT 3-4 will be after this section's text answer

T8. What is the median age of the training set? You can easily modify the age in the dataframe by

ANS: 24

T9. 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

T10. Write a logistic regression classifier using gradient descent as learned in class. Use PClass, Sex, Age, and Embarked as input features.

ANS: I implemented T9-T10 in code section (see code section), The accuracy on training is 79.69%

T11. Submit a screenshot of your submission (with the scores). Upload your code to courseville.

ANS: Test score is 76.79%

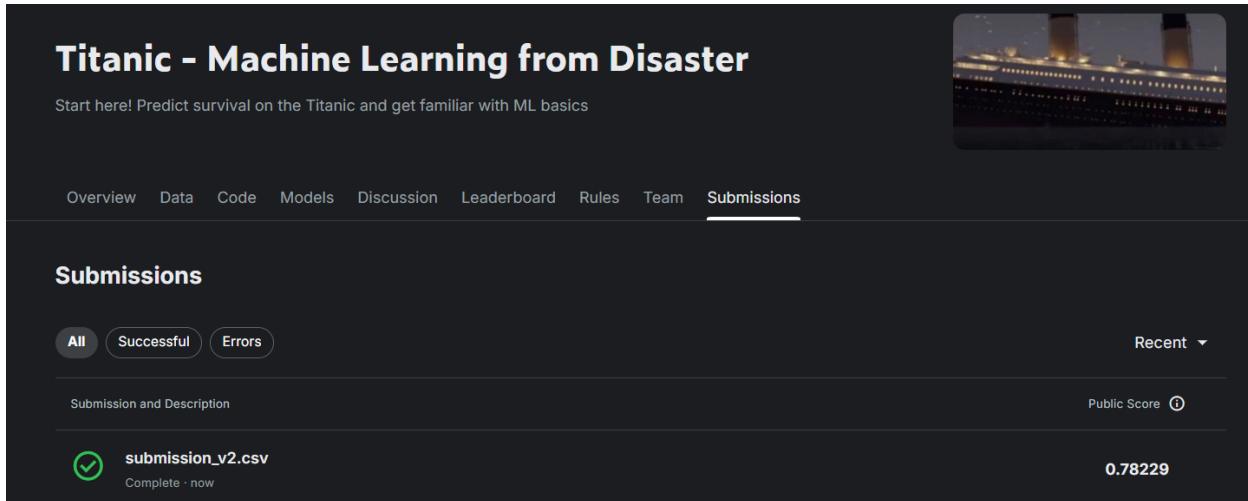
The screenshot shows the 'Submissions' page of the 'Titanic - Machine Learning from Disaster' competition. At the top, there is a banner with the text 'Start here! Predict survival on the Titanic and get familiar with ML basics' and an image of the Titanic ship. Below the banner, there is a navigation bar with links: Overview, Data, Code, Models, Discussion, Leaderboard, Rules, Team, and Submissions. The 'Submissions' link is underlined, indicating it is the active page. The main content area is titled 'Submissions'. It features a filter bar with 'All', 'Successful', and 'Errors' buttons, and a dropdown menu set to 'Recent'. Below the filter, there is a table with one row. The table has columns for 'Submission and Description', 'Public Score' (with a tooltip icon), and a 'Recent' button. The row contains a green checkmark icon, the file name 'submission.csv', the status 'Complete - now', and the score '0.76794'.

**T12. Try adding some higher order features to your training ( $x_21$ ,  $x_1x_2$ ,...).**

**Does this model has better accuracy on the training set? How does it perform on the test set?**

**ANS:** The code for this version will be shown after this text answer (only change in model and predict part)

The accuracy on training set is 82.15%, slightly better than non-higher order data.



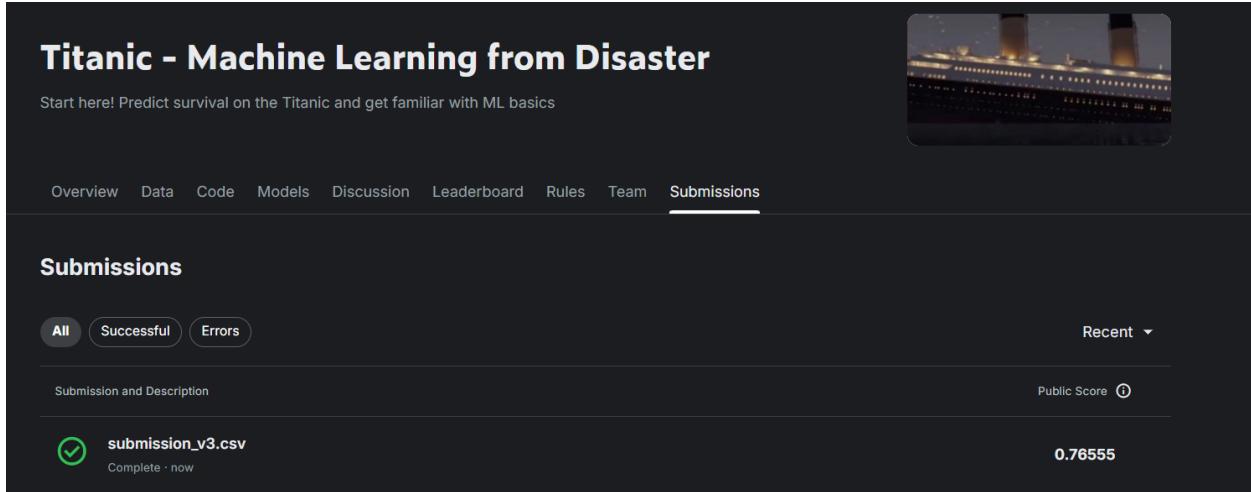
The screenshot shows the Kaggle interface for the "Titanic - Machine Learning from Disaster" competition. The page title is "Titanic - Machine Learning from Disaster". Below it, a sub-header says "Start here! Predict survival on the Titanic and get familiar with ML basics". On the right, there's a small image of the Titanic ship at night. A navigation bar at the top includes links for Overview, Data, Code, Models, Discussion, Leaderboard, Rules, Team, and Submissions, with "Submissions" being the active tab. Below the navigation, the word "Submissions" is centered. Underneath, there are three filter buttons: "All" (selected), "Successful", and "Errors". To the right of these filters is a "Recent" dropdown menu. The main content area displays a table with two columns: "Submission and Description" and "Public Score ⓘ". The first row in the table shows a green checkmark icon next to the file name "submission\_v2.csv", followed by the text "Complete · now" and the score "0.78229".

Test data accuracy is also higher at 78.23%

**T13. What happens if you reduce the amount of features to just Sex and Age?**

**ANS:** The code for this version will be shown after this text answer (only change in model and predict part)

The accuracy on training set is 78.68%, slightly better than normal but lower than higher order data.



The screenshot shows the Kaggle interface for the "Titanic - Machine Learning from Disaster" competition. The page title is "Titanic - Machine Learning from Disaster". Below it is a subtitle: "Start here! Predict survival on the Titanic and get familiar with ML basics". On the right, there is a small image of the Titanic ship at night. The navigation bar includes links for Overview, Data, Code, Models, Discussion, Leaderboard, Rules, Team, and Submissions, with "Submissions" being the active tab. Below the navigation bar, the section title "Submissions" is displayed. Underneath, there are three filter buttons: "All", "Successful", and "Errors", with "All" being selected. To the right of these filters is a "Recent" dropdown menu. The main content area shows a table with one row of data. The columns are "Submission and Description" and "Public Score ⓘ". The row contains a green checkmark icon, the file name "submission\_v3.csv", the status "Complete · now", and the score "0.76555".

But the test result is worst than normal case, only 76.55%

**OT3. 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.**

**ANS:** The code for this version will be shown after this text answer (only change in model and predict part)

The accuracy on training set is 78.68%

The screenshot shows the Kaggle interface for the 'Titanic - Machine Learning from Disaster' competition. The 'Submissions' tab is selected. A successful submission is listed with the file name 'submission\_v4.csv', a completion status of 'Complete - now', and a public score of 0.76555. The interface includes navigation links like Overview, Data, Code, Models, Discussion, Leaderboard, Rules, Team, and Submissions, along with a search bar and filter options for All, Successful, and Errors.

The test result is the same as only using age and sex, only 76.55%

**OT4. 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.**

**ANS:**

```
theta_matrix = np.linalg.inv(X_b.T.dot(X_b)).dot(X_b.T).dot(y)

print("Theta (Gradient Descent):", theta)
print("Theta (Matrix Inversion):", theta_matrix)

weight_mse = np.mean((theta - theta_matrix) ** 2)
print(f"\nMSE of the weight difference: {weight_mse}")

Theta (Gradient Descent): [ 1.02648941 -0.94338079  2.53270066 -0.01848152  0.31728152]
Theta (Matrix Inversion): [ 0.76512686 -0.18828708  0.49299994 -0.00478358  0.04513561]

MSE of the weight difference: 0.9746213905435914
```

**OT5-OT7 will be prove in hand writing below**