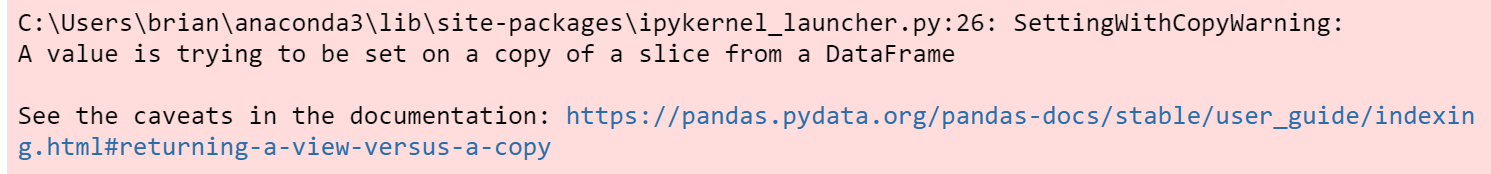
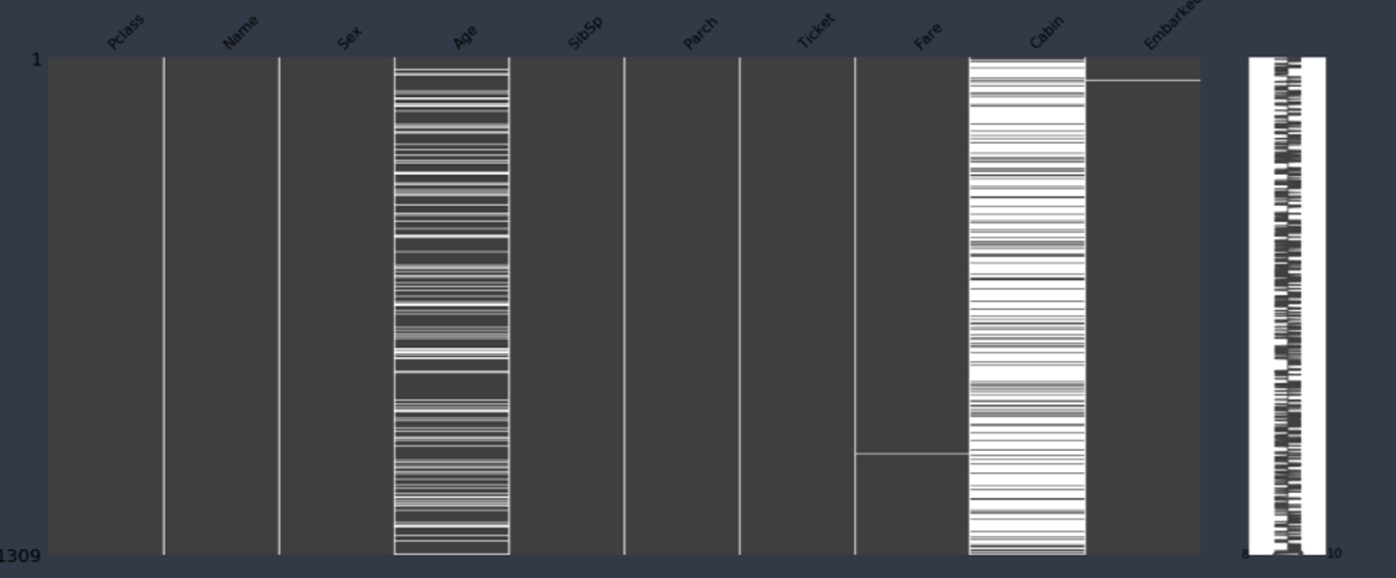
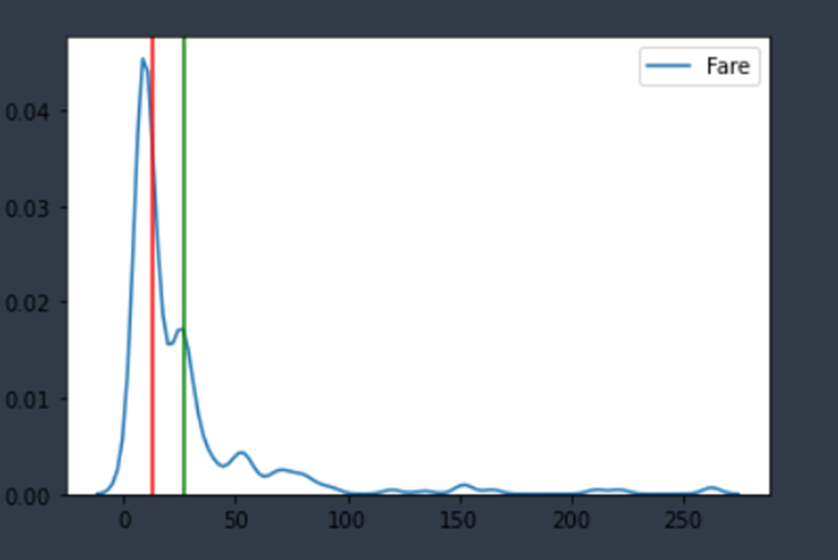
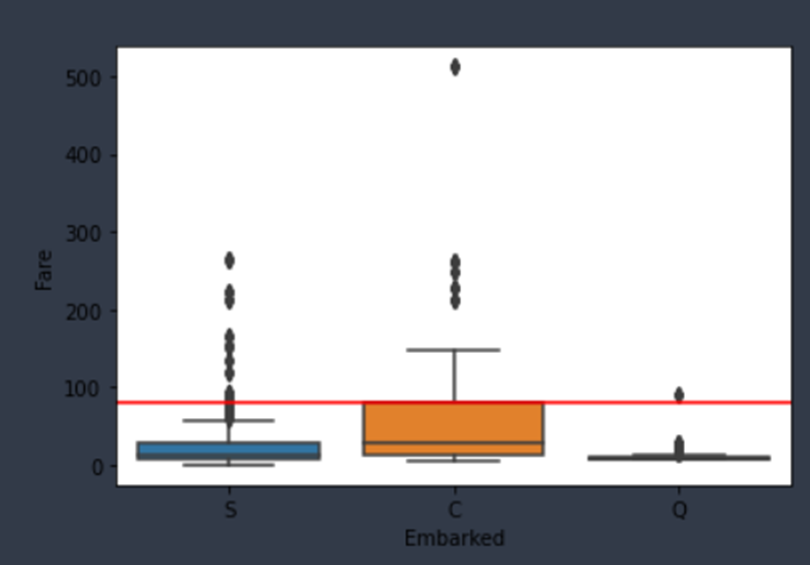
1. Details of different submission attempts:
   1. Trial 1 (fill empty NAN, on Age dataset, with the median Age of the same P\_Class):
      1. We used median of Age, with respect to the same P\_Class, to fill the NAN of the initial Age dataset.
      2. This approach is to make a more accurate prediction on filling the NAN of the Age dataset, by grouping the median Age of the same P\_Class instead of the whole median of the dataset.
   2. Trial 2 (fill empty NAN, on Age dataset, with KNN method):





1. Models Tried:
   1. Random Forest
   2. SVC
2. Data-Cleaning:



* Features w/ null values shown by missingno library.
  1. Cabin & Deck Feature
     1. As shown in the image above, the Cabin feature has the most number of missing/null values.
     2. Ultimately, we used One Hot Encoding w/ Cabin to help categorize the deck feature.
     3. Empty Deck based on null Cabin Assigned is assigned to Deck\_N.
     4. In the end, we drop the Cabin Column as we no longer need it.
  2. Age Feature
     1. For the Age Feature, we used K-Mean Clustering or KNN.
     2. We used the KNNImputer from the Scikit-learn library.
  3. Fare Feature
     1. We used a density graph to visualize the fares of people embarked.
     2. 
     3. From the graph above, Green means Mean & Red means Median.
     4. Based on the graph above, we decided to use Median for cleaning the Fare data.
  4. Embarked Feature
     1. We used a boxplot to determine how to clean the Embarked feature.
     2. 
     3. Based on the boxplot above, we found that passengers were mainly embarked from port C, or from Cherbourg Port. So, we cleaned embarked feature with port C.

1. Feature Engineering:
   1. We need to classify some of the datasets into different classes so that it can be easily read by the model:
      1. Sex:
         1. 0 for male.
         2. 1 for female.
      2. Embarked:
         1. 0 for port ‘S’.
         2. 1 for port ‘C’.
         3. 2 for port ‘Q’.
      3. Age:
         1. 0 for ages 0-16.
         2. 1 for ages 16-32.
         3. 2 for ages 32-48.
         4. 3 for ages 38-64
         5. 4 for ages greater than 64.
      4. Fare:
         1. 0 for 0-7.91.
         2. 1 for 7.91-14.45.
         3. 2 for 14.45-31.
         4. 3 for greater than 31.
      5. Next, it is important to understand that the people having a family might have a bigger chance of survival with their family, so it is important to make a column to classify that. We will call it 'isAlone'. It can be calculated by taking the sum of 'SibSp' and 'Parch' columns and adding it with 1 (for the person itself). We can classify it as 0 for alone and 1 for not alone.
2. Machine Learning Model:
   1. Random Forest:
      1. Random Forest is an ensemble method for classification which uses multiple decision tree during training and outputs the class that is the mode or the mean of the individual trees.
         1. Reference: Wikipedia.
      2. Why are we using Random Forest:
         1. It is unexcelled in accuracy.
         2. It runs effectively on a big dataset.
         3. It estimates properly which feature is important for classification.
         4. It corrects the Decision Tree's habit of overfitting.
         5. It maintains accuracy even if data is missing.
         6. It can handle several input features effectively.
3. Team’s workload distribution:
   1. Andrew Thenedi: Codes, Report
   2. Briann: Codes
   3. Srijan: Codes, PowerPoint
   4. Vincent: Codes