**Assignment 1**

**Titanic- Machine Learning from Disaster**

**Hongyi Hao**



**1 Problem Description**

* 1. Problem 1:

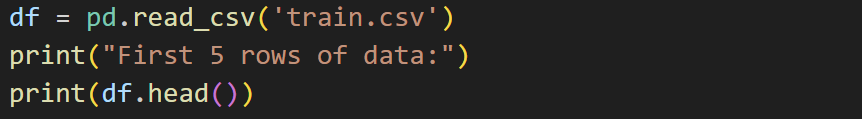
Which feature do you believe is the most important for the model’s performance? Justify your answer with evidence, such as data analysis, visualizations, or feature importance scores.

* 1. Problem 2:

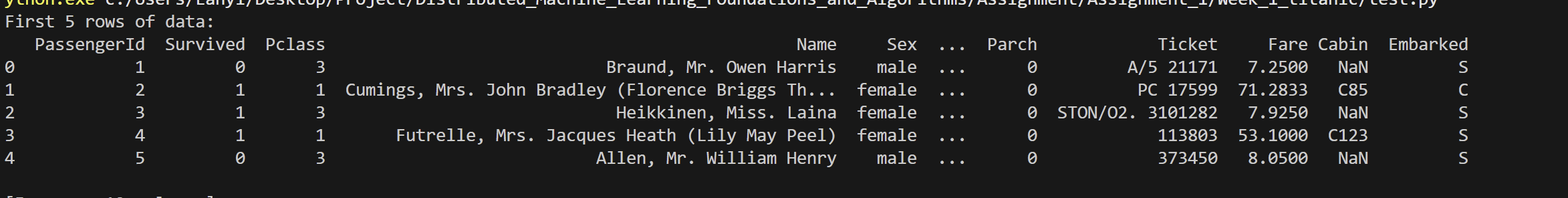
For the "Age" feature, what alternative methods could be used to handle missing values instead of filling them with 0? Explain how these methods could improve the model’s performance.

1. **Data Preparation**
   1. Read and Visualize Data
2. Read the train.csv file.
3. Print out the first 5 samples.

To begin with, I used the ***pandas.read\_csv*** function in Python to read the train.csv data. And then print out the first 5 samples.



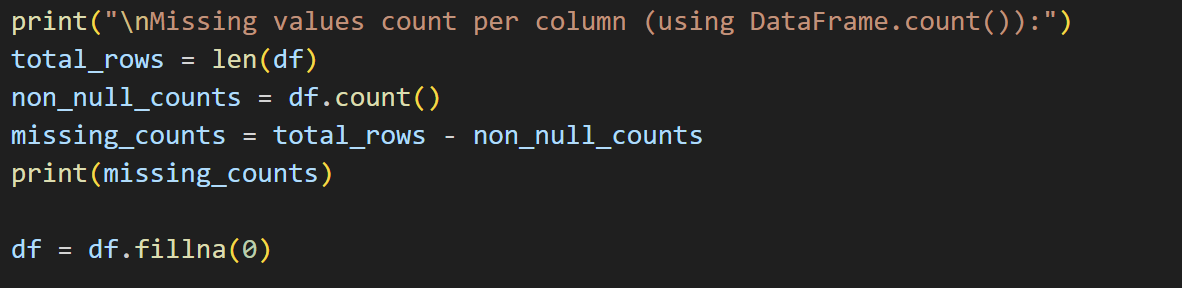
**Output:**



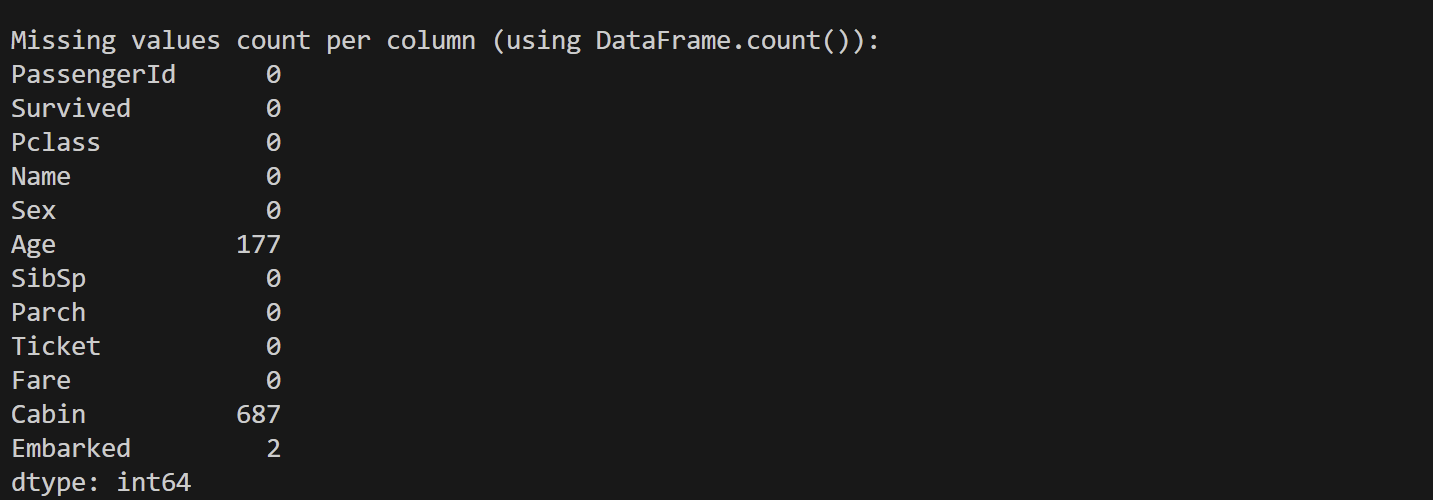
* 1. Handling Missing Data

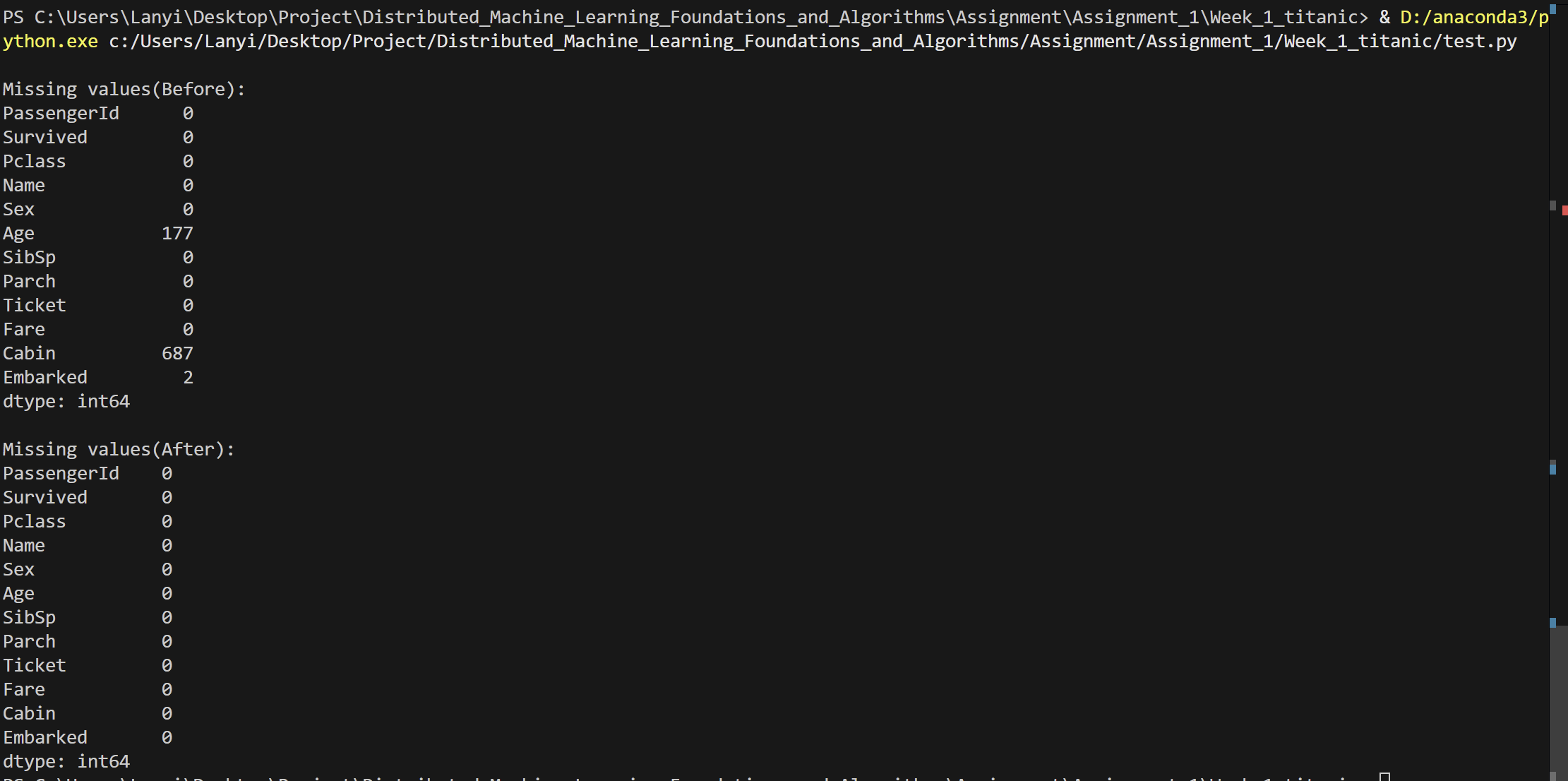
1. Print out the count of missing values in each column.
2. Impute missing entries with zero.

Then, I used the ***pandas.DataFrame.count*** and ***pandas.DataFrame.fillna*** function to count the number of values in each column and impute missing entries with 0.



**Output:**

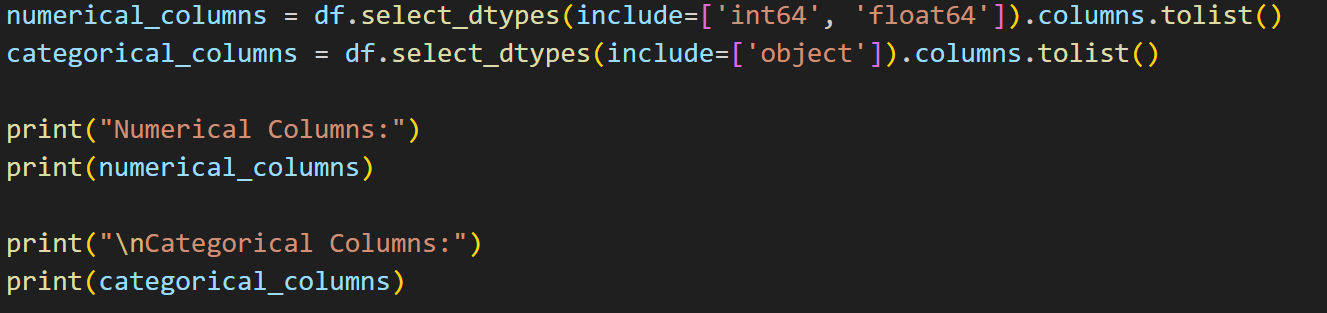


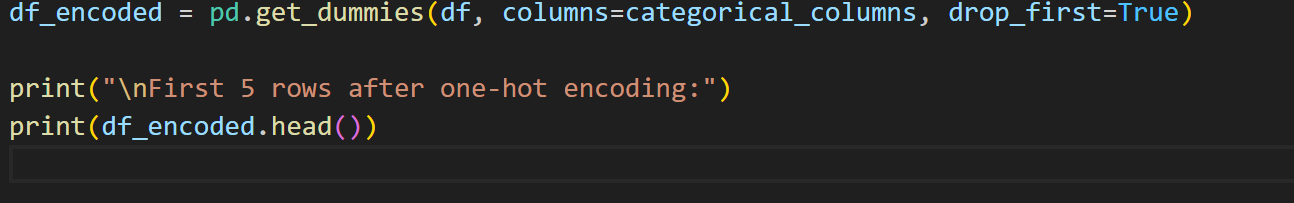
**Comparison Before And After Filling:**

* 1. Handling numerical and categorical values.

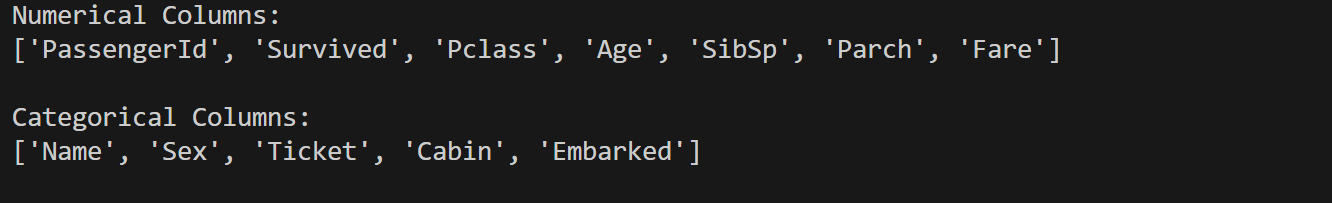
1. Identify and write down which columns are numerical and which are categori cal.
2. Convert the categorical columns into one-hot encoded columns. Print out the first 5 samples to verify.

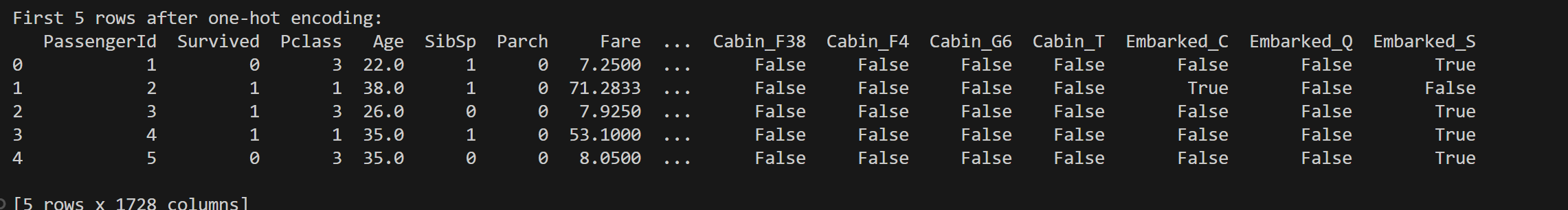
Additionally, I used the ***pandas.DataFrame.head*** function and ***pandas.get\_dummies*** function to Identify and write down the type of the values convert the categorical columns into one-hot encoded columns.





**Output:**

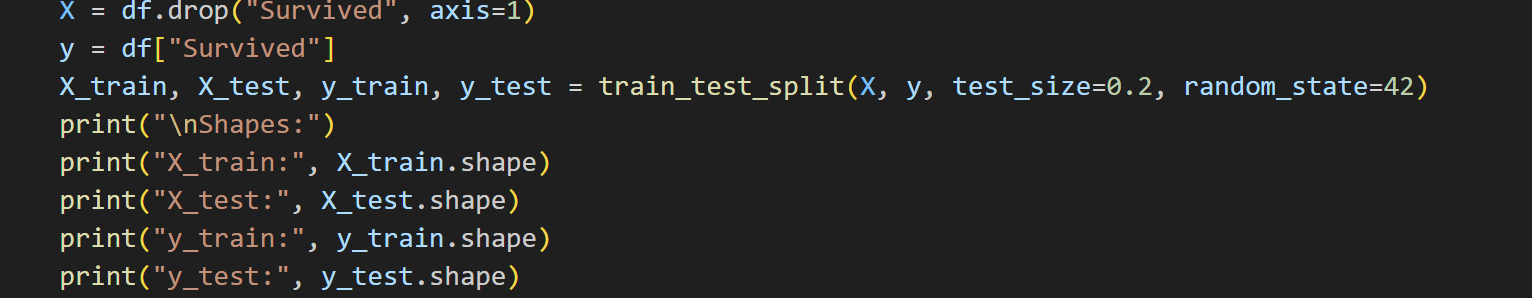




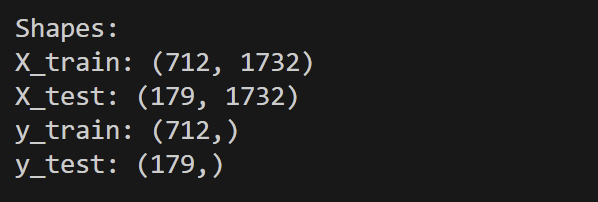
* 1. Dataset Splitting

(a) Divide the cleaned dataset into training features (x\_train), training targets (y train), testing features (x\_test), and test targets (y\_test).

(b) Print out each set’s shape.



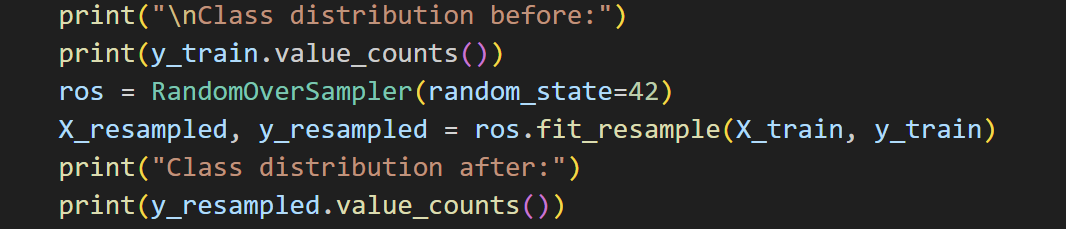
**Output:**

****

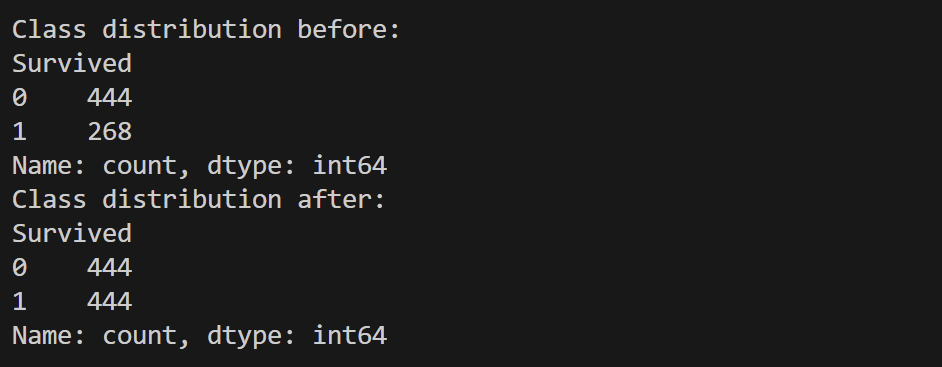
* 1. Addressing Imbalanced Labels for Training Set

(a) How many samples are there for survival and non-survival?

(b) Use oversampling technique to handle imbalanced classes.

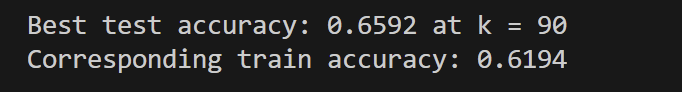


**Output:**

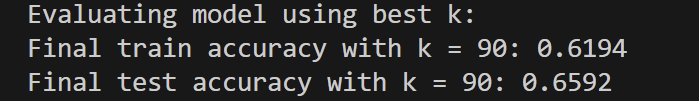


1. **Model Training and Validation**
   1. Training a k-nearest neighbors classifier using the training dataset you have from prev section.
   2. Print out your model’s prediction accuracy on the training dataset.
   3. Print out your model’s prediction accuracy on the test dataset
   4. Plot the training/validation accuracy with respect to different values of k. The range of k should be [1, 100]. Save the plotted to your pdf submission file.
2. def train\_knn(X\_train, y\_train, X\_test, y\_test):
3. accuracies\_train = []
4. accuracies\_test = []
5. k\_range = range(1, 101)
6. print("KNN Start...\n")
7. print(f"{'k':<5} {'Train Acc':<12} {'Test Acc':<12}")
8. print("-" \* 30)
9. for k in k\_range:
10. model = KNeighborsClassifier(n\_neighbors=k)
11. model.fit(X\_train, y\_train)
12. y\_pred\_train = model.predict(X\_train)
13. y\_pred\_test = model.predict(X\_test)
14. acc\_train = accuracy\_score(y\_train, y\_pred\_train)
15. acc\_test = accuracy\_score(y\_test, y\_pred\_test)
16. accuracies\_train.append(acc\_train)
17. accuracies\_test.append(acc\_test)
18. print(f"{k:<5} {acc\_train:<12.4f} {acc\_test:<12.4f}")
19. # Find the best k
20. best\_k = accuracies\_test.index(max(accuracies\_test)) + 1
21. best\_train\_acc = accuracies\_train[best\_k - 1]
22. best\_test\_acc = accuracies\_test[best\_k - 1]
23. # Paint the graph
24. plt.figure(figsize=(10,6))
25. plt.plot(k\_range, accuracies\_train, label='Train Accuracy')
26. plt.plot(k\_range, accuracies\_test, label='Test Accuracy')
27. plt.xlabel("k")
28. plt.ylabel("Accuracy")
29. plt.title("KNN Accuracy vs k")
30. plt.legend()
31. plt.grid()
32. plt.savefig("screenshots/knn\_accuracy.png")
33. plt.show()
34. print(f"\n Best test accuracy: {best\_test\_acc:.4f} at k = {best\_k}")
35. print(f" Corresponding train accuracy: {best\_train\_acc:.4f}")
36. print("\n Evaluating model using best k:")
37. best\_model = KNeighborsClassifier(n\_neighbors=best\_k)
38. best\_model.fit(X\_train, y\_train)
39. final\_train\_pred = best\_model.predict(X\_train)
40. final\_test\_pred = best\_model.predict(X\_test)
41. final\_train\_acc = accuracy\_score(y\_train, final\_train\_pred)
42. final\_test\_acc = accuracy\_score(y\_test, final\_test\_pred)
43. print(f" Final train accuracy with k = {best\_k}: {final\_train\_acc:.4f}")
44. print(f" Final test accuracy with k = {best\_k}: {final\_test\_acc:.4f}")

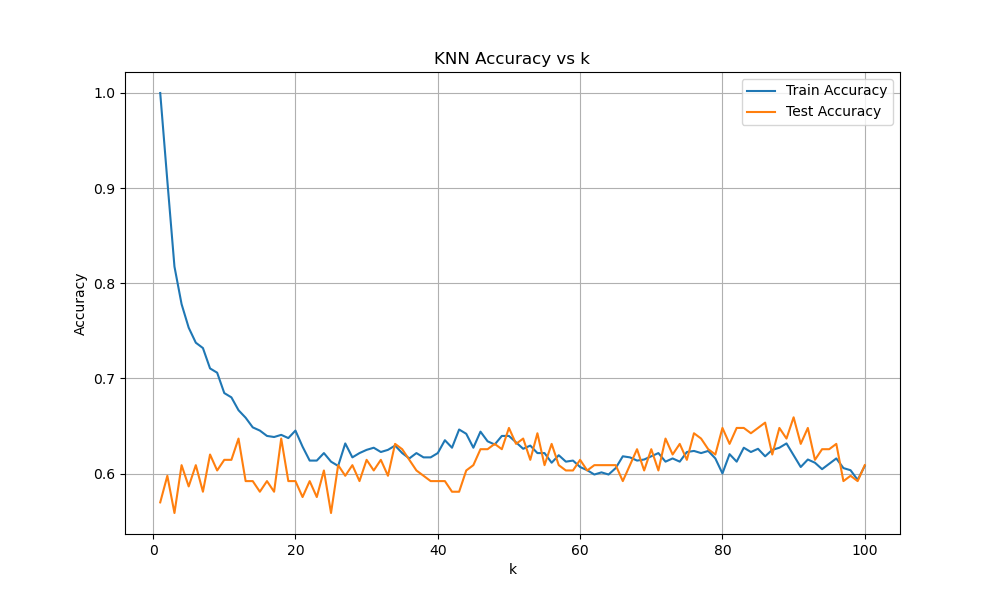
According to the test and experiment the model’s prediction accuracy on the training dataset and the model’s prediction accuracy on the test dataset are as followed:



All in all, evaluating model using best k:

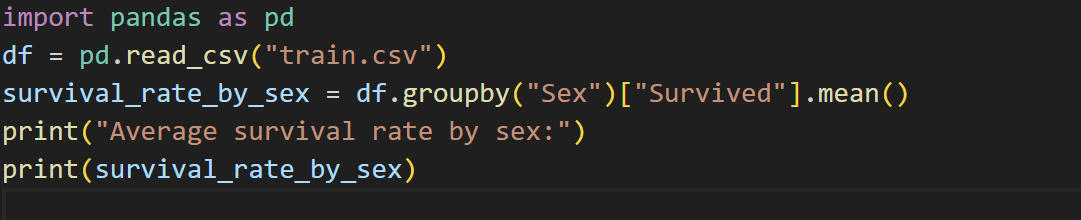


And then, the training/validation accuracy with respect to different values of k. The range of k should be [1, 100].

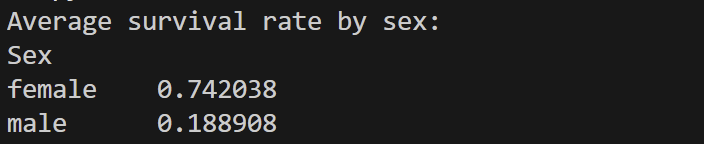


1. **Analytical Questions**
2. Which feature do you believe is the most important for the model’s performance?

In my opinion, the most important factor affecting model performance is 'Sex', as analyzed using Python based on the original dataset:



**Output**:



The output shows a significant disparity in survival rates between passengers of different genders, indicating that the survival rate for women was much higher than that for men. This also reflects the prevailing rescue strategy of 'women and children first', which contributed to the higher survival rate of women in the Titanic disaster. At the same time, After OneHot encoding, the discriminative ability of the KNN model can be significantly improved, and compared to other high-dimensional sparse features, the information density is higher and the effect is more clear.

1. For the "Age" feature, what alternative methods could be used to handle missing values instead of filling them with 0?

When dealing with missing values for the 'Age' feature, directly filling them with 0 can introduce outliers and potentially mislead the model's judgments, which may affect subsequent model training. There are 4 other methods to deal with this feature.

1. **Fill with median:**

Use the median of the Age from the entire dataset to fill in missing values, which reduces the impact of outliers. The median is more robust and can alleviate the disturbance to the model caused by filling in 0 or the mean value.



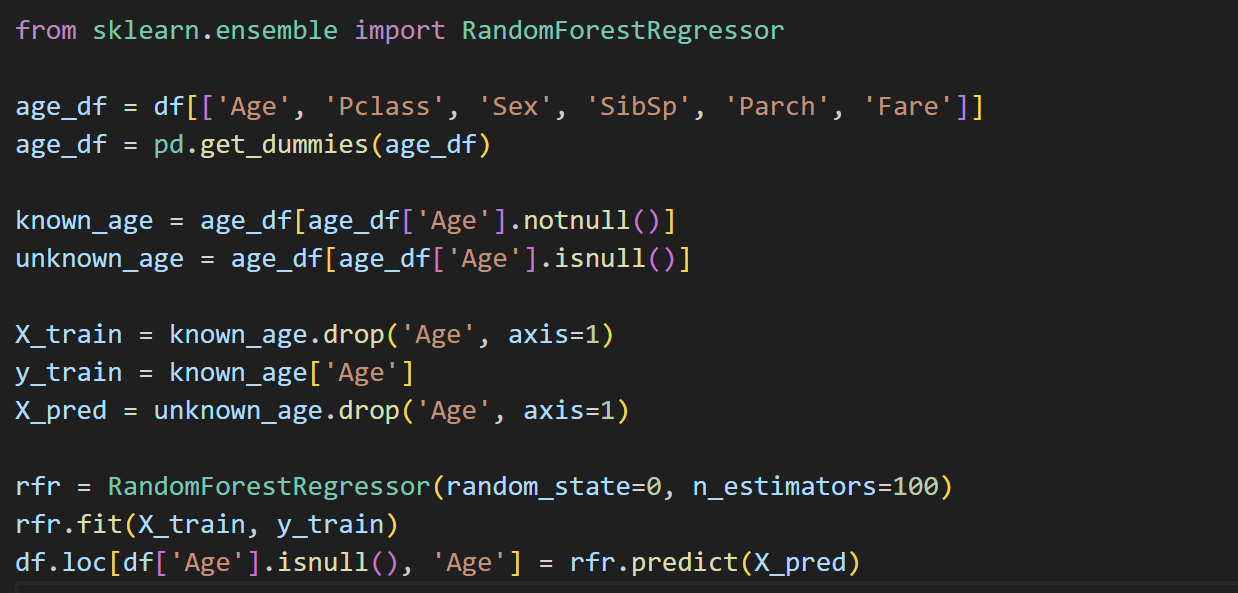
1. **Fill in using the median grouped by 'Pclass':**

Fill in the missing ages by calculating the median age for each group based on passengers' Pclass. This method takes advantage of the potential correlation between 'cabin class and age', filling in based on categorical features, which is more in line with the true data distribution.



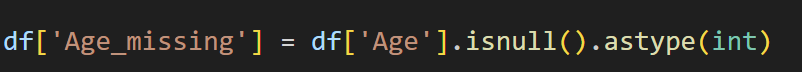
1. **Using regression models to predict missing values:**

Use known features such as “Pclass”, Sex, and Fare to train a regression model to predict Age. This is a machine learning method based on the relationships between other features, which is more flexible than simple statistics and can provide personalized fill values, improving the quality of the imputation.



1. **Add 'Is Age Missing' as a new feature:**

While filling in Age = 0, add a new feature “Age\_missing” to indicate whether it is missing. This approach explicitly informs the model that 'this is missing data,' allowing the model to learn whether 'missingness' itself is related to the prediction target (Survived).



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method | median | grouped by 'Pclass' | regression models | Add Age Missing |
| Accuracy | 0.5866 | 0.6034 | 0.6201 | 0.5866 |

Form 1: Accuracy Comparison

More reasonable methods for handling missing values, such as filling with group medians, regression prediction, and missing indicator features, are not only more in line with the data distribution, but also significantly enhance the model's ability to learn real situations, thereby improving classification performance. Especially when using regression prediction for filling, the accuracy compared to other methods and the original method shows a significant improvement in the final accuracy.