De La Salle University - Manila
Gokongwei College of Engineering
Computer Engineering Department

Project 1

Titanic Machine Learning from Disaster

Submitted by:

Jadman, Daine Janssen R.
Yap, Kalvin Kenjo H.
Landayan, Jed Gabriel C.

In partial fulfillment of the course
Computer Architecture and Organization Laboratory
LBOEC3B

December 6, 2022

I. Objectives

The objective of this project is to build a predictive model that answers the question: “what sorts of people were more likely to survive?” using passenger data (ie name, age, gender, socio-economic class, etc). The researchers intend to test the use of different models to see what model is more efficient to use in terms of its resulting accuracy.

II. Introduction

The British passenger liner RMS Titanic launched back in May 1911. It was managed by the White Star Liner which had a voyage from Southampton, England with a destination of New York City in the United States. The RMS Titanic sank in April 1912 after hitting an iceberg in the North Atlantic Ocean while it was on its expedition (History, 2009). The sinking of the Titanic resulted in the death of over a thousand people that includes both passengers and crew members. Fortunately, there were passengers as well as crew members who were able to survive the wreck.

For this project, the task and problem is to be able to calculate and predict the death or survival of a specific passenger of the Titanic that is based on a given set of variables and characteristics that describe the said passenger, such as their age, sex, and their passenger class while they were on the ship.

III. Statistical Analysis of the Dataset

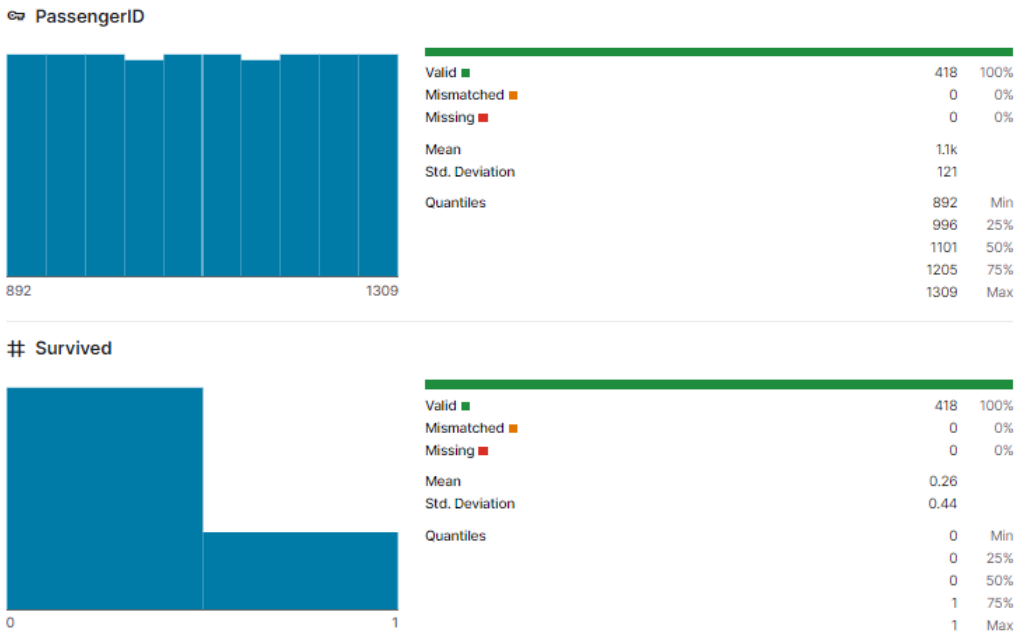

Figure 1. Decision Tree Model

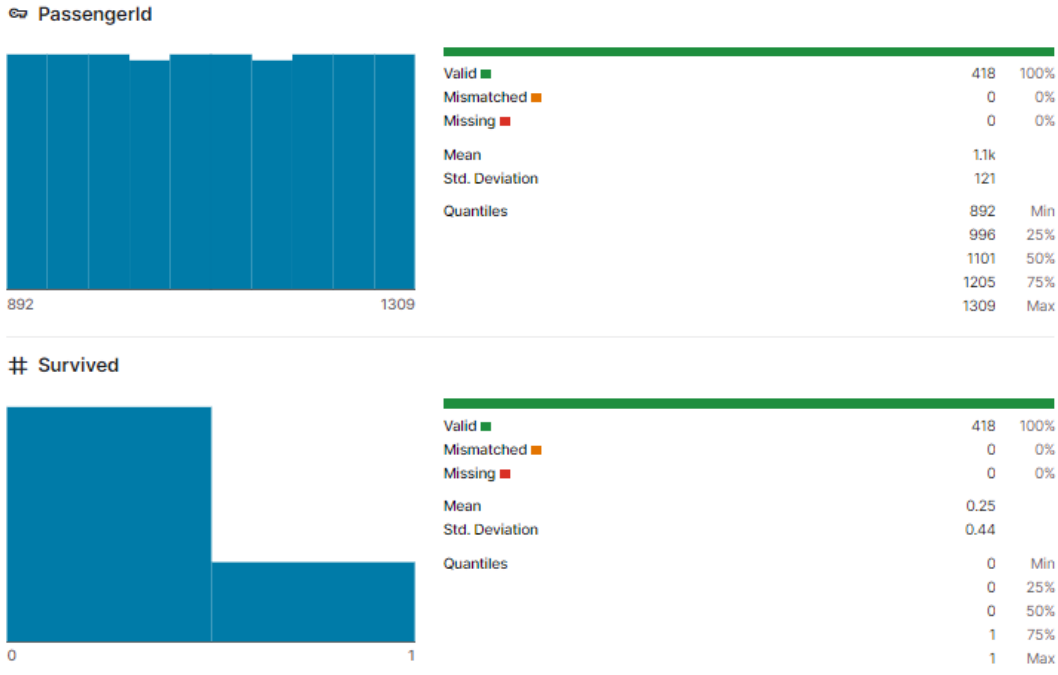

Figure 2. Classification KNN

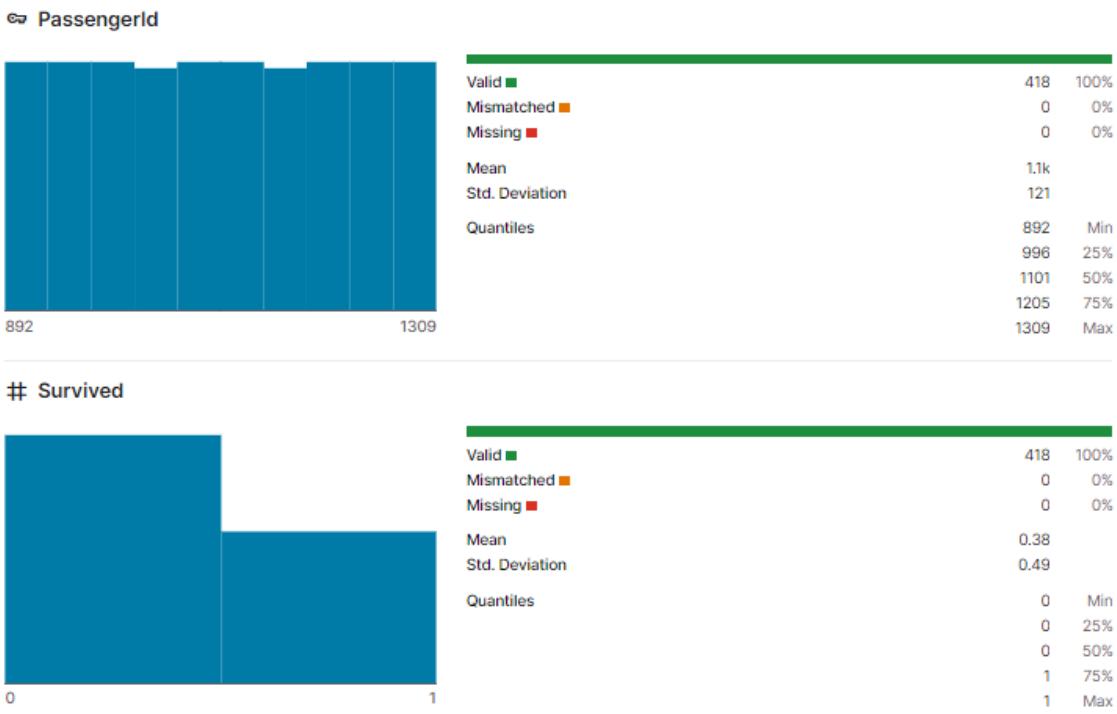

Figure 3. Logistic Regression

For the statistical analysis of the dataset, the researchers have decided to use the implementation of the decision tree model due to its ease of accessibility to interpret and visualize. Another good factor of this implementation is that it can handle any sorts of data, along with the fact that normalization and feature scaling is not exactly required for this type of model. Consequently, the preparation of the data is not much compared to other models which may require a seldom multitude of data.

IV. Data Parsing and Wrangling

The data parsed from the training file consisted of survived, passenger class, gender, and age data rows. The other data provided such as their names, number of siblings, number of parents or children, ticket number, passenger fare, cabin number, and port of embarkation were not parsed since some variables could not be classified or read by the training model. The data wrangled was the ages of the passengers since some rows were empty. To mitigate this issue, these blanks were filled with the average age of all the other filled rows. Afterwards, ages were discretized into five categories which were 0, 15, 25, 60, and 100. Furthermore, the data for gender and age were created into an index vector. Additionally, the same process was used for logistic regression.

Regarding the data parsed from the testing data set, the passenger class, gender, age, and passenger ID, embarked, and fare were parsed as well. Subsequently, the other information that was provided was disregarded. However, KNN classification focused on the conversion of values to readable input for the machine learning model. Furthermore, a sample wrangling procedure implemented was that the researchers implemented a constant to fill in the blanks of the embarked and fare data rows.

V. Model Testing and Hyperparameter Tuning

The first model utilized for the titanic project was the Decision Tree model which obtained the highest accuracy. The fitctree function trains the data provided based on the information contained in passenger class, gender, and age, and the output which is survived. The function cvloss then returns the optimal pruning level for the output of fitctree. Subsequently, pruning the decision tree is an efficient strategy to combat overfitting. Lastly, the predictions from the test data set is produced by using the passenger class, gender, age and predict function with the information from the pruning tree.

The second model utilized for the titanic project was the use of KNN Classification which obtained an accuracy percentage lower than the Decision Tree model. In this case, the predict function is meant to return a vector of the predicted class labels for the predictor data in the matrix. Furthermore, training is still performed because it is meant to specify all the classifier options and train them accordingly.

The third model utilized for the titanic project was the Logistic Regression which used the fitglm function to train the training data set. The predict function is then used to predict the response values of the linear regression model to the data in the survived row. The output of this is then rounded to the nearest integer and the confusion chart is generated from there. The testing dataset is then tested by using the output of the fitglm function and the chosen data from the test dataset. However, this model obtained an accuracy percentage which is lower than both the Decision Tree and KNN classification model.

The methodological process implemented was first based on the hypothesis of the desirable variables. The first variable used was the sex of the passenger since women were prioritized during the evacuation. The age and passenger class were great suspects as well for the survivability factor. Afterwards, the kaggle score when tested was 77.5% for the decision tree model. When the embarked data row was added, it had decreased the score; therefore, it was removed. Fare was tested as well, however the score still had not

increased. This led to testing the age discretization values as it had increased the score slightly which provided a possibility of reaching the target of 79%. The exact same process was used in logistic regression, but it was 2.5% less accurate even with the same data and age discretization values. For the KNN classification model, the data rows embarked and fare were used which helped increase the score. However, it was not enough to reach the desired score.

VI. Data and Results (Confusion Matrix and Kaggle Score)

```
%read the train table
traintable = readtable('train.csv');
datatable = (table2cell(traintable));
survived = cell2mat(datatable(:,2));
pclass = cell2mat(datatable(:,3));
sex = datatable(:,5);
age = cell2mat(datatable(:,6));

%filling empty ages with average age
mean_age = mean(age,'omitnan');
age(isnan(age))= mean_age;

agediff = discretize(age, [0 15 25 60 100],'categorical',{'Under 15', '15-25', '25-60', 'Above 60'});
agediff = grp2idx(agediff);
sex = grp2idx(sex);
arrangedtable = table(pclass, sex, agediff, survived);

%decision tree
dectree = fitctree(arrangedtable(:,1:3),arrangedtable(:,4));

[~,~,~,BestLevel] = cvloss(dectree,'subtrees','all','treesize','min');
pruningtree = prune(dectree,'Level',BestLevel);
view(pruningtree,'mode','graph')

label = predict(pruningtree,arrangedtable(:,1:end-1));
conmattree = confusionmat(survived,label);
acctree = trace(conmattree)/sum(conmattree, 'all');

%read the test table
traintable = readtable('test.csv');
datatable = (table2cell(traintable));
```

Figure 1. Decision Tree Model Part 1

```
pclass = cell2mat(datatable(:,2));
sex = datatable(:,4);
age = cell2mat(datatable(:,5));
PassengerID = cell2mat(datatable(:,1));

%filling empty ages with average age
mean_age = mean(age,'omitnan');
age(isnan(age))= mean_age;

agediff = discretize(age, [0 15 25 60 100],'categorical',{'Under 15', '15-25', '25-60', 'Above 60'});
agediff = grp2idx(agediff);
sex = grp2idx(sex);
testtable = table(pclass, sex, agediff);

%decision tree
ytree = predict(pruningtree,testtable(:,1:end));
Survived = ytree;

out = table(PassengerID, Survived);
writetable(out,'predictions.csv');
```

Figure 2. Decision Tree Model Part 2

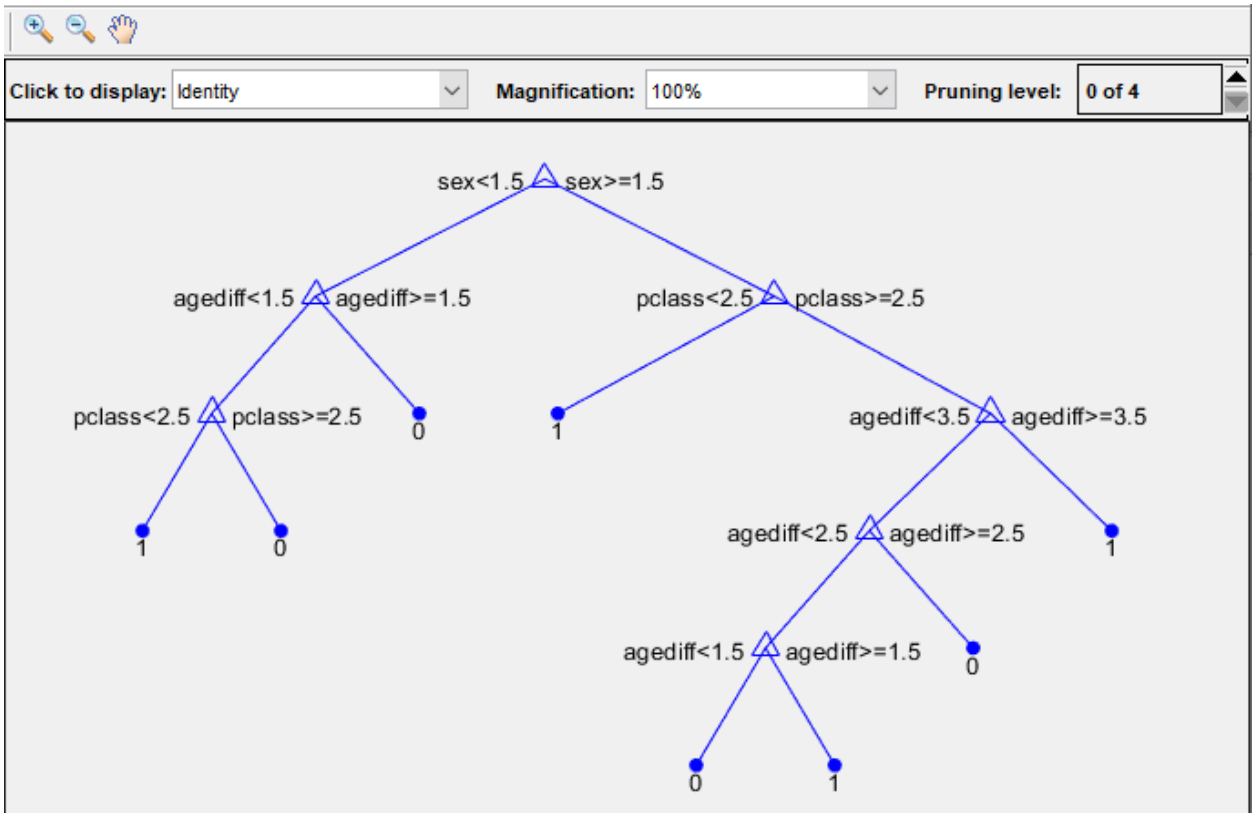

Figure 3. Decision Tree Result

0.79186

Figure 4. Kaggle Score using Decision Tree

```
traintable1 = readtable('train.csv');

traintable1.Name=[];
traintable1.Ticket=[];
traintable1.Cabin=[];

agenan=median(traintable1.Age,'omitnan');
traintable1.Age=fillmissing(traintable1.Age,"constant",agenan);

t = traintable1.Embarked;
conv = cell2mat(t);
modever = mode(conv);

traintable1.Embarked = fillmissing(traintable1.Embarked,'constant',modever);
traintable1.Embarked = categorical(traintable1.Embarked);
traintable1.Embarked = renamecats(traintable1.Embarked,{'S','C','Q'},{'1','2','3'});
traintable1.Embarked = str2double(string(traintable1.Embarked));

traintable1.Sex = categorical(traintable1.Sex);
traintable1.Sex = renamecats(traintable1.Sex,{'male','female'},{'1','2'});
traintable1.Sex = str2double(string(traintable1.Sex));

sum(ismissing(traintable1))
```

Figure 5. KNN Classification Model Part 1

```
traintable2 = readtable('test.csv');
passid = traintable2.PassengerId;

traintable2.Name=[];
traintable2.Ticket=[];
traintable2.Cabin=[];

agenan=median(traintable2.Age,'omitnan');
traintable2.Age=fillmissing(traintable2.Age,"constant",agenan);

a = traintable2.Embarked;
conv1 = cell2mat(a);
modever1 = mode(conv1);

traintable2.Embarked = fillmissing(traintable2.Embarked,'constant',modever1);
farenan=median(traintable2.Fare,'omitnan');
traintable2.Fare=fillmissing(traintable2.Fare,"constant",farenan);

traintable2.Sex = categorical(traintable2.Sex);
traintable2.Sex = renamecats(traintable2.Sex,{'male','female'},{'1','2'});
traintable2.Sex = str2double(string(traintable2.Sex));

traintable2.Embarked = categorical(traintable2.Embarked);
traintable2.Embarked = renamecats(traintable2.Embarked,{'S','C','Q'},{'1','2','3'});
traintable2.Embarked = str2double(string(traintable2.Embarked));

sum(ismissing(traintable2))

traintable1_train = traintable1(:,[1 3:8]);
traintable2_train =traintable1(:,2);

model = fitcknn(traintable1_train,traintable2_train,'OptimizeHyperparameters','auto');

prediction=predict(model,traintable2);

output=[passid prediction];

store=array2table(output,'VariableNames',{'PassengerId','Survived'});

writetable(store,'predictions.csv','Delimiter',',');
```

Figure 6. KNN Classification Model Part 2

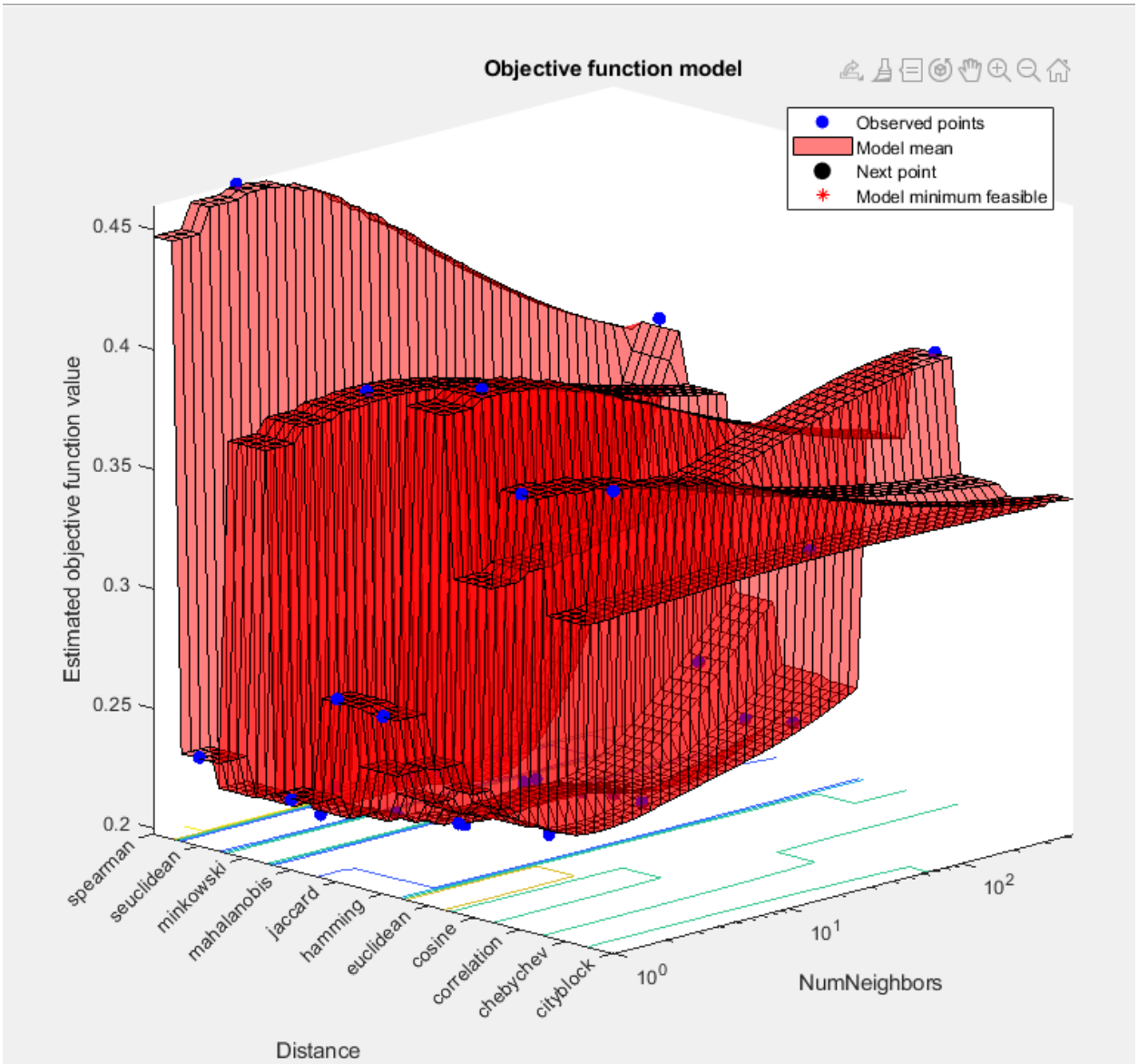

Figure 7. KNN Classification Program Result

0.77033

Figure 8. Kaggle Score using KNN Classification

```
%read the train table
traintable = readtable('train.csv');
datatable = (table2cell(traintable));
survived = cell2mat(datatable(:,2));
pclass = cell2mat(datatable(:,3));
sex = datatable(:,5);
age = cell2mat(datatable(:,6));

%filling empty ages with average age
mean_age = mean(age,'omitnan');
age(isnan(age))= mean_age;

agediff = discretize(age, [0 15 25 60 100],'categorical',{'Under 15', '15-25', '25-60', 'Above 60'});
agediff = grp2idx(agediff);
sex = grp2idx(sex);
arrangedtable = table(pclass, sex, agediff, survived);

%logistic regression
model = fitglm(arrangedtable, 'Distribution','binomial');
ypred = predict(model,arrangedtable(:,1:end-1));
ypred = round(ypred);
confusionchart(ypred,survived);

%read the test table
testtable = readtable('test.csv');
datatesttable = (table2cell(testtable));
```

Figure 9. Logistic Regression Model Part 1

```
pclass = cell2mat(datatesttable(:,2));
sex = datatesttable(:,4);
agetest = cell2mat(datatesttable(:,5));
PassengerID = cell2mat(datatesttable(:,1));

%filling empty ages with average age
mean_age_test = mean(agetest,'omitnan');
agetest(isnan(agetest))= mean_age_test;

agediff = discretize(agetest, [0 15 25 60 100],'categorical',{'Under 15', '15-25', '25-60', 'Above 60'});
agediff = grp2idx(agediff);
sex = grp2idx(sex);
testtable = table(pclass, sex, agediff);

%logistic regression
ypred2 = predict(model,testtable(:,1:end));
Survived = round(ypred2);


out = table(PassengerID, Survived);
writetable(out,'predictions.csv');
```

Figure 10. Logistic Regression Model Part 2

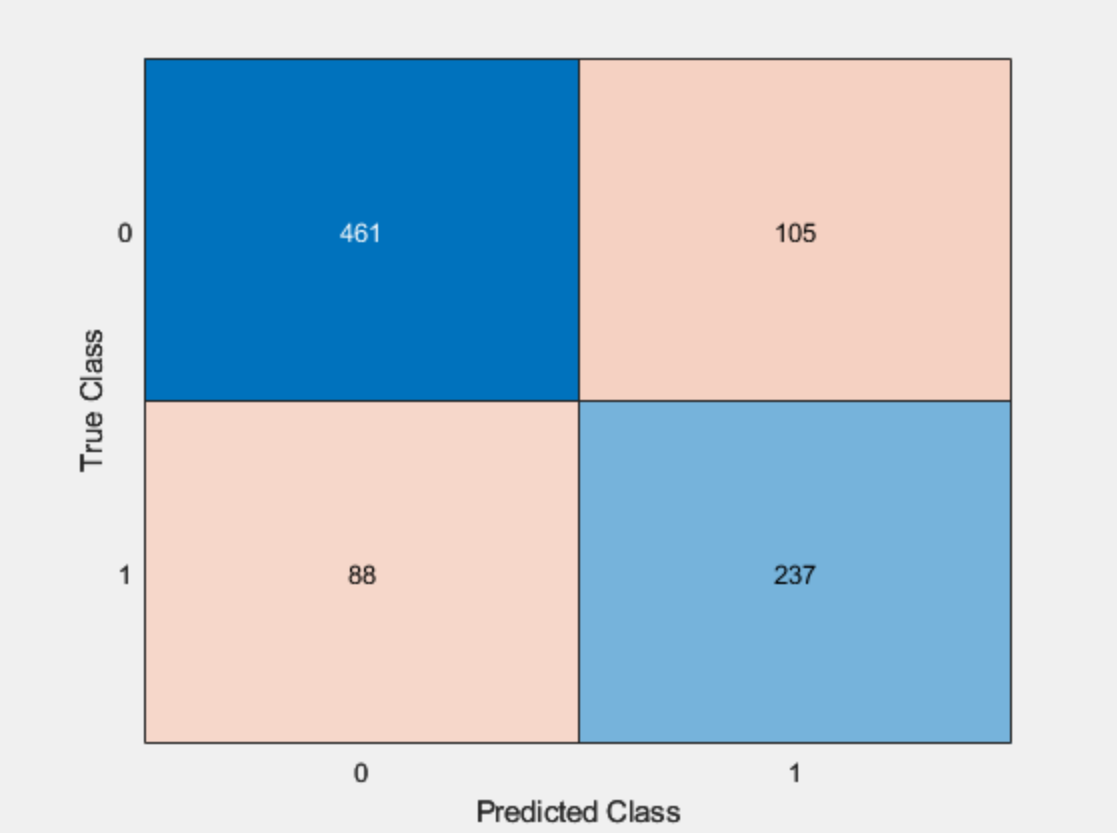

Figure 11. Logistic Regression Confusion Chart

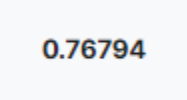

Figure 12. Kaggle Score using Logistic Regression

The researchers were able to conclude that using Decision Tree for the Titanic machine learning problem was the most efficient model to use due to its accuracy being 79.186% which is higher when compared to the other models such as Classification KNN having 77.033% and Logistic Regression obtaining a percentage of 76.794% accuracy. Additionally, the snippets displayed above show that using Decision Tree contained the highest accuracy percentage which further incentivizes its output to be more concise in terms of accuracy when tested compared to the two other models. Even though decision trees and logistic regression are highly identical aside from the training function utilized, the accuracy score displayed by kaggle shows a big gap. Therefore, this algorithm was the most suitable option.

VII. References

Decision tree advantages and disadvantages: Decision Tree Regressor. EDUCBA. (2021). Retrieved December 12, 2022, from https://www.educba.com/decision-tree-advantages-and-disadvantages/

History. (2009). *Titanic.* Retrieved from https://www.history.com/topics/early-20th-century-us/titanic

Mathworks. (n.d.). *Classification Trees.* Retrieved from https://www.mathworks.com/help/stats/classification-trees.html

Mathworks. (n.d.). *Fit Binary decision tree for multiclass classification.* Retrieved from https://www.mathworks.com/help/stats/fitctree.html

VIII. Appendix

Project 1 Models:

A. Decision Tree

```
%read the train table
traintable = readtable('train.csv');
datatable = (table2cell(traintable));
survived = cell2mat(datatable(:,2));
pclass = cell2mat(datatable(:,3));
sex = datatable(:,5);
age = cell2mat(datatable(:,6));
%filling empty ages with average age
mean_age = mean(age,'omitnan');
age(isnan(age))= mean_age;
```

```
agediff = discretize(age, [0 15 25 60 100],'categorical',{'Under 15', '15-25',
'25-60', 'Above 60'});

agediff = grp2idx(agediff);

sex = grp2idx(sex);

arrangedtable = table(pclass, sex, agediff, survived);

%decision tree

dectree = fitctree(arrangedtable(:,1:3),arrangedtable(:,4));

[~,~,~,BestLevel] = cvloss(dectree,'subtrees','all','treesize','min');

pruningtree = prune(dectree,'Level',BestLevel);

view(pruningtree,'mode','graph')

label = predict(pruningtree,arrangedtable(:,1:end-1));

conmattree = confusionmat(survived,label);

acctree = trace(conmattree)/sum(conmattree, 'all');

%read the test table

traintable = readtable('test.csv');

datatable = (table2cell(traintable));

pclass = cell2mat(datatable(:,2));

sex = datatable(:,4);

age = cell2mat(datatable(:,5));

PassengerID = cell2mat(datatable(:,1));

%filling empty ages with average age

mean_age = mean(age,'omitnan');

age(isnan(age))= mean_age;

agediff = discretize(age, [0 15 25 60 100],'categorical',{'Under 15', '15-25',
'25-60', 'Above 60'});

agediff = grp2idx(agediff);

sex = grp2idx(sex);

testtable = table(pclass, sex, agediff);

%decision tree
```

```
ytree = predict(pruningtree,testtable(:,1:end));

Survived = ytree;

out = table(PassengerID, Survived);

writetable(out,'predictions.csv');
```

B. Logistic Regression

```
%read the train table

traintable = readtable('train.csv');

datatable = (table2cell(traintable));

survived = cell2mat(datatable(:,2));

pclass = cell2mat(datatable(:,3));

sex = datatable(:,5);

age = cell2mat(datatable(:,6));

%filling empty ages with average age

mean_age = mean(age,'omitnan');

age(isnan(age))= mean_age;

agediff = discretize(age, [0 15 25 60 100],'categorical',{'Under 15', '15-25',
'25-60', 'Above 60'});

agediff = grp2idx(agediff);

sex = grp2idx(sex);

arrangedtable = table(pclass, sex, agediff, survived);

%logistic regression

model = fitglm(arrangedtable, 'Distribution','binomial');

ypred = predict(model,arrangedtable(:,1:end-1));

ypred = round(ypred);

confusionchart(ypred,survived);

%read the test table

testtable = readtable('test.csv');
```

```
datatesttable = (table2cell(testtable));

pclass = cell2mat(datatesttable(:,2));

sex = datatesttable(:,4);

agetest = cell2mat(datatesttable(:,5));

PassengerID = cell2mat(datatesttable(:,1));

%filling empty ages with average age

mean_age_test = mean(agetest,'omitnan');

agetest(isnan(agetest))= mean_age_test;

agediff = discretize(agetest, [0 15 25 60 100],'categorical',{'Under 15',
'15-25', '25-60', 'Above 60'});

agediff = grp2idx(agediff);

sex = grp2idx(sex);

testtable = table(pclass, sex, agediff);

%logistic regression

ypred2 = predict(model,testtable(:,1:end));

Survived = round(ypred2);

out = table(PassengerID, Survived);

writetable(out,'predictions.csv');
```

C. KNN Classification

```
traintable1 = readtable('train.csv');

traintable1.Name=[];

traintable1.Ticket=[];

traintable1.Cabin=[];

agenan=median(traintable1.Age,'omitnan');

traintable1.Age=fillmissing(traintable1.Age,"constant",agenan);

t = traintable1.Embarked;

conv = cell2mat(t);
```

```
modever = mode(conv);

traintable1.Embarked = fillmissing(traintable1.Embarked,'constant',modever);

traintable1.Embarked = categorical(traintable1.Embarked);

traintable1.Embarked =
renamecats(traintable1.Embarked,{'S','C','Q'},{'1','2','3'});

traintable1.Embarked = str2double(string(traintable1.Embarked));

traintable1.Sex = categorical(traintable1.Sex);

traintable1.Sex = renamecats(traintable1.Sex,{'male','female'},{'1','2'});

traintable1.Sex = str2double(string(traintable1.Sex));

sum(ismissing(traintable1))

traintable2 = readtable('test.csv');

passid = traintable2.PassengerId;

traintable2.Name=[];

traintable2.Ticket=[];

traintable2.Cabin=[];

agenan=median(traintable2.Age,'omitnan');

traintable2.Age=fillmissing(traintable2.Age,"constant",agenan);

a = traintable2.Embarked;

conv1 = cell2mat(a);

modever1 = mode(conv1);

traintable2.Embarked = fillmissing(traintable2.Embarked,'constant',modever1);

farenan=median(traintable2.Fare,'omitnan');

traintable2.Fare=fillmissing(traintable2.Fare,"constant",farenan);

traintable2.Sex = categorical(traintable2.Sex);

traintable2.Sex = renamecats(traintable2.Sex,{'male','female'},{'1','2'});

traintable2.Sex = str2double(string(traintable2.Sex));

traintable2.Embarked = categorical(traintable2.Embarked);

traintable2.Embarked =
renamecats(traintable2.Embarked,{'S','C','Q'},{'1','2','3'});
```

```
traintable2.Embarked = str2double(string(traintable2.Embarked));

sum(ismissing(traintable2))

traintable1_train = traintable1(:,[1 3:8]);

traintable2_train =traintable1(:,2);

model =
fitcknn(traintable1_train,traintable2_train,'OptimizeHyperparameters','auto');

prediction=predict(model,traintable2);

output=[passid prediction];

store=array2table(output,'VariableNames',{'PassengerId','Survived'});

writetable(store,'predictions.csv','Delimiter',',');
```