METIS PROJECT 3 SUBMITTED BY MIKE BERNARDO

Surviving the Titanic

Using Classification for Survival Prediction

53%

1317 out of 2453 passengers max capacity when launched

38%

Only 498 of the passengers survived.

THE QUESTION

Would I Survive the Titanic?

Classifying Passengers

Data Aquisition

data downloaded from Kaggle.com
Titanic Survivor dataset

Initial EDA

analyze the correlation and directionality of the data using python libraries

Split Data

60% Training. 20% Validation and 20% Testing data

Build Model

build classification models using training data via feature engineering, transformations etc

Train Model

score models and perform more feature engineering / transformations as needed

Test Model

Use and evaluate model on Test data. Perform more EDA, feature engineering as needed.

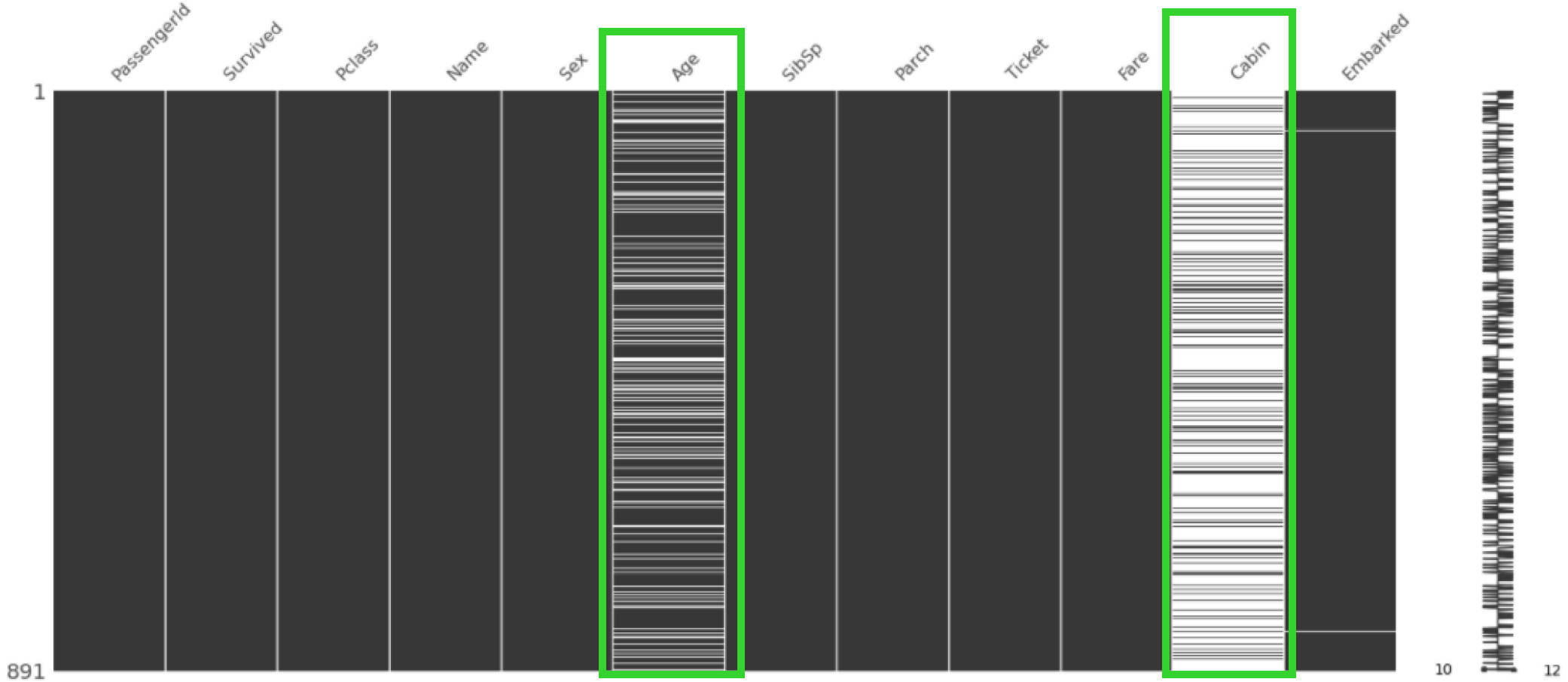
Deploy Model

Predict survivability based on new data.

The Data

Feature	Description
PassengerID	A unique index for passenger
Survived	refers to whether passenger survived. 1 = survived 0 = not survived
Pclass	Ticket class. 1 = First class ticket. 2 = Second class ticket. 3 = Third class
AgeGroup	Child (<13), Teen(<18), Adult(<65>, Senior(65+)
Sex	Passenger's gender. Male or Female.
SibSp	Number of siblings or spouses travelling with each passenger
Ticket	Number of siblings or spouses travelling with each passenger
Parch	Number of parents of children travelling with each passenger
Embarked	Southhampton(S),Cherbourg(C) or Queesnstown(Q)

EDA: Missing Data



EDA: Imputing Missing Data

Age:

- range from 0-80
- 177 missing values
- mean of ages to impute missing

Embarked:

- Southhampton(S),Cherbourg(C) or Queesnstown(Q)
- 2 missing values
- mode of Embarked to impute missing

Cabin:

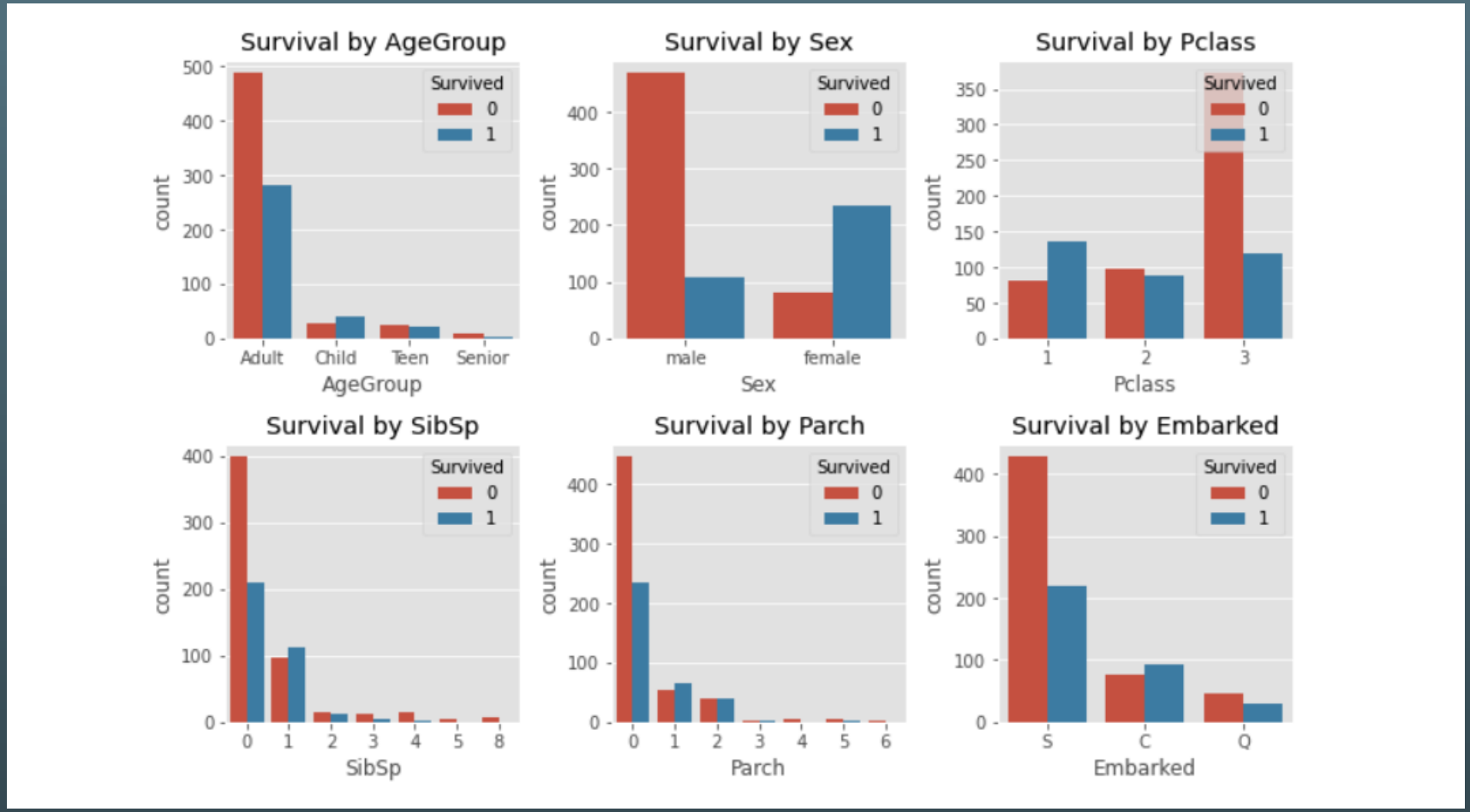
- 687 missing values
- removed from dataset

Visualizing the Data

Looking at Survival rates based on gender and other features

More Survivors:

- Adults
- Females
- First Class
- Solo
- Southampton



Data Preparation: Feature Engineering

Numerically encoding categorical data using sklearn.preprocessing LabelEncoder

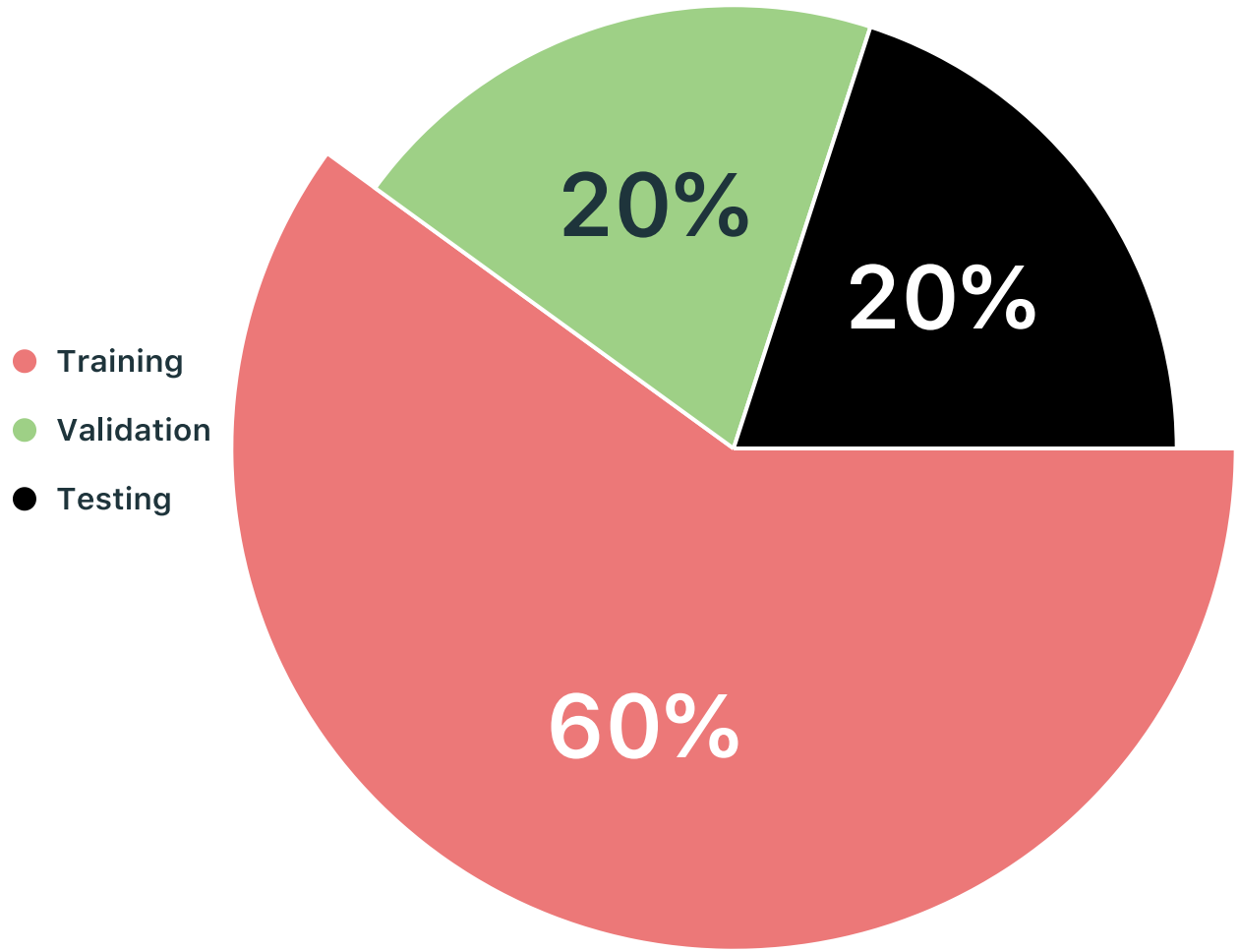
Tasks:

- Sex encoded as 0,1
- Age simplified to AgeGroup
- AgeGroup encoded to values 0,1,2,3
- Embarked encoded to 0,1,2
- Drop columns: passengerId, Name, Ticket

	Pclass	Sex	SibSp	Parch	Fare	Embarked	AgeGroup	Survived
0	3	1	1	0	7.2500	2	0	0
1	1	0	1	0	71.2833	0	0	1
2	3	0	0	0	7.9250	2	0	1
3	1	0	1	0	53.1000	2	0	1
4	3	1	0	0	8.0500	2	0	0
...	...	...	...	...	...	...	...	...
886	2	1	0	0	13.0000	2	0	0
887	1	0	0	0	30.0000	2	0	1
888	3	0	1	2	23.4500	2	0	0
889	1	1	0	0	30.0000	0	0	1
890	3	1	0	0	7.7500	1	0	0

Splitting Data

Before building any models, separate data into train, validation and test data sets



Model Selection: kNN

Try different algorithms on the training data and show confusion matrix

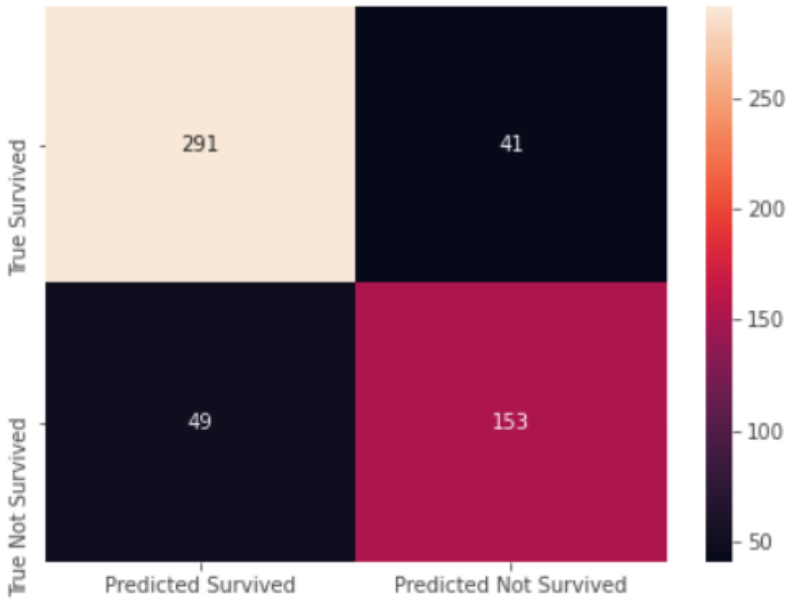
kNN Scores:

- Model Score: 83%
- Accuracy: 74%
- Precision: 69%
- Recall: 67%
- F1: 68%

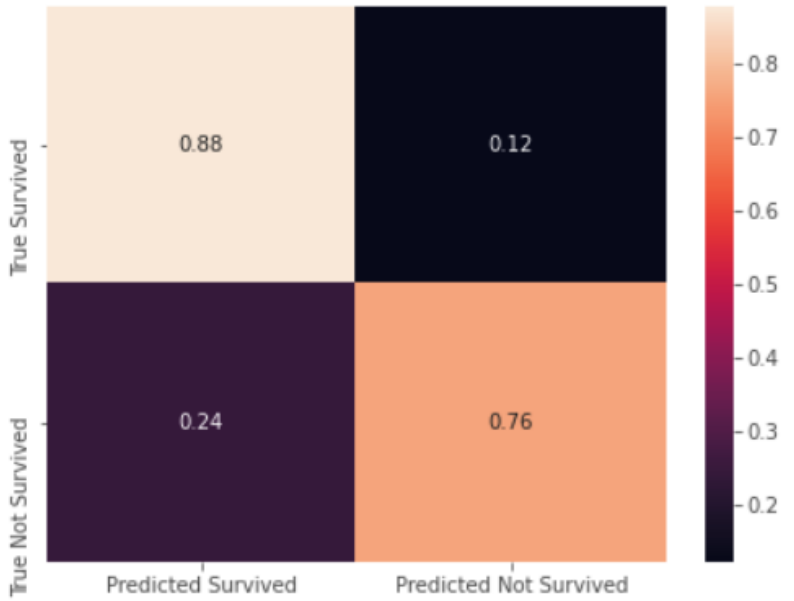
Hyperparameter: k

default value: 5

Confusion Matrix (values)



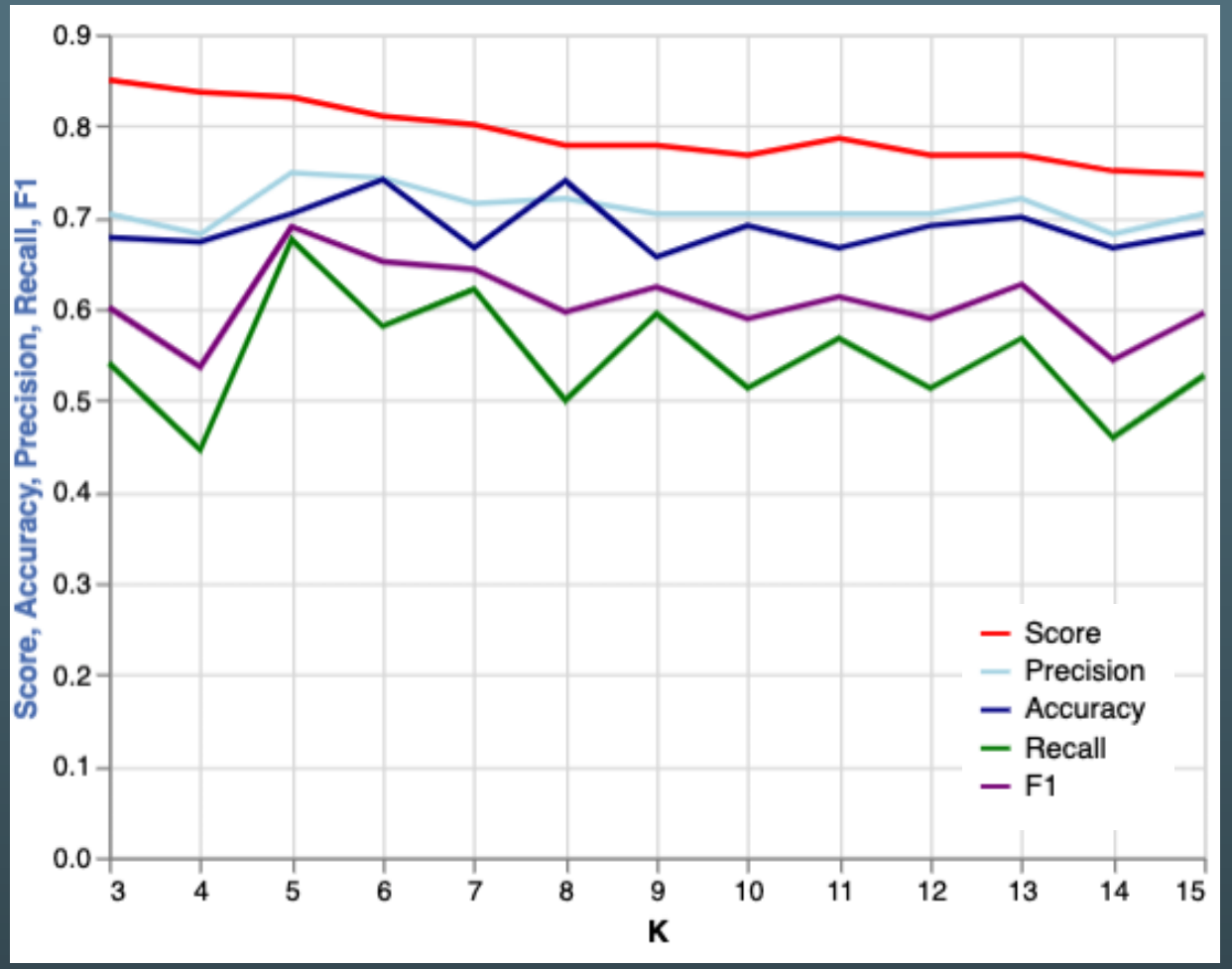
Confusion Matrix (%)



Model Tuning: kNN

Try different algorithms on the training data and show confusion matrix

	K	Score	Accuracy	Precision	Recall	F1
2	5.00	0.83	0.75	0.70	0.68	0.69
3	6.00	0.81	0.74	0.74	0.58	0.65
5	8.00	0.78	0.72	0.74	0.50	0.60
10	13.00	0.77	0.72	0.70	0.57	0.63
4	7.00	0.80	0.72	0.67	0.62	0.64
0	3.00	0.85	0.70	0.68	0.54	0.60
6	9.00	0.78	0.70	0.66	0.59	0.62
7	10.00	0.77	0.70	0.69	0.51	0.59
8	11.00	0.79	0.70	0.67	0.57	0.61
9	12.00	0.77	0.70	0.69	0.51	0.59
12	15.00	0.75	0.70	0.68	0.53	0.60
1	4.00	0.84	0.68	0.67	0.45	0.54
11	14.00	0.75	0.68	0.67	0.46	0.54



Model Tuning: Optimized kNN

Run model with k=3 (optimal value for highest Score)

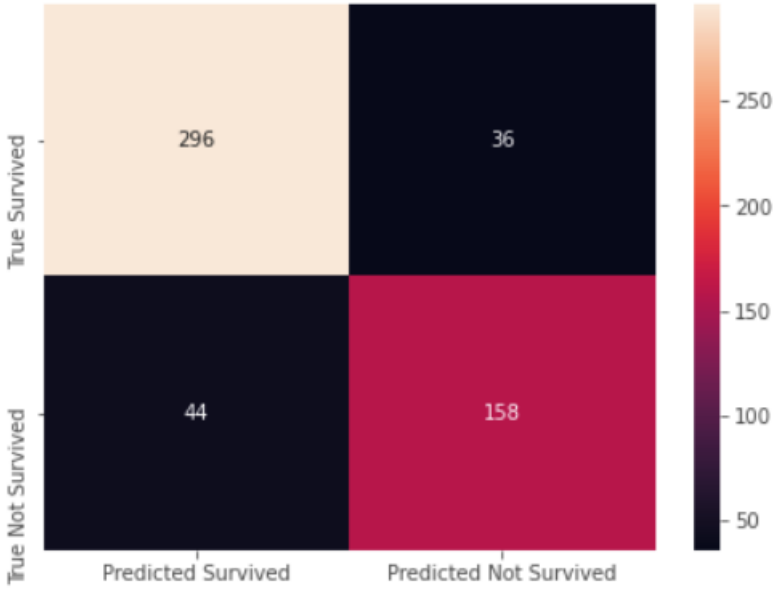
kNN Scores:

- Model Score: 85%
- Accuracy: 79%
- Precision: 76%
- Recall: 72%
- F1: 74%

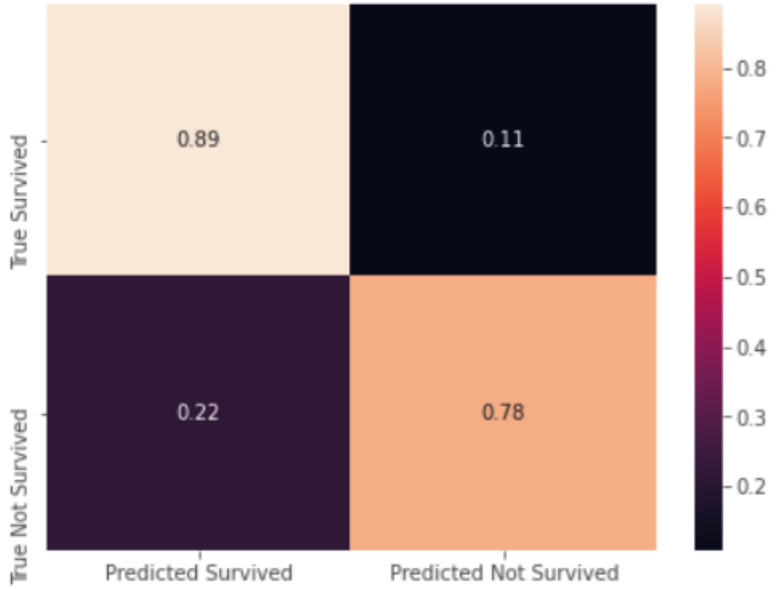
Hyperparameter: k

Optimal Value: 3

Confusion Matrix (values)



Confusion Matrix (%)



Model Selection: AdaBoost

Try different algorithms on the training data and show confusion matrix

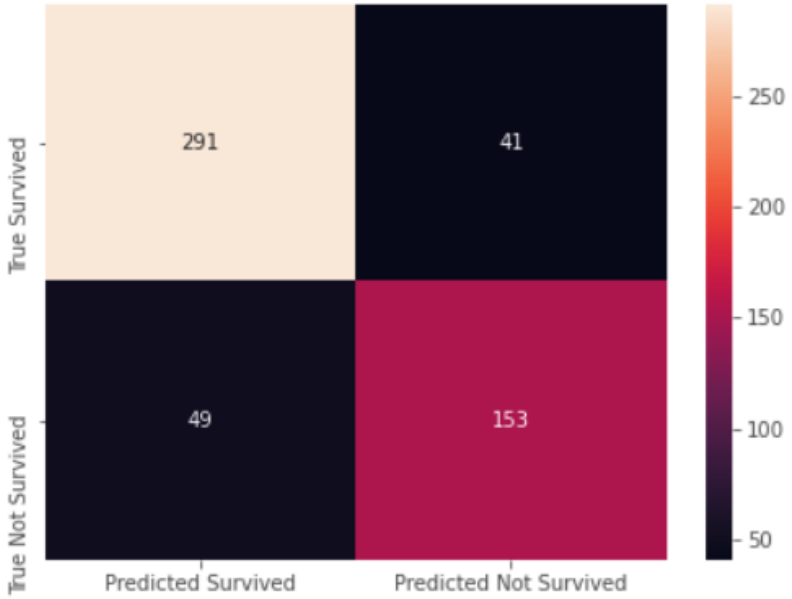
kNN Scores:

- Model Score: 83%
- Accuracy: 76%
- Precision: 69%
- Recall: 71%
- F1: 70%

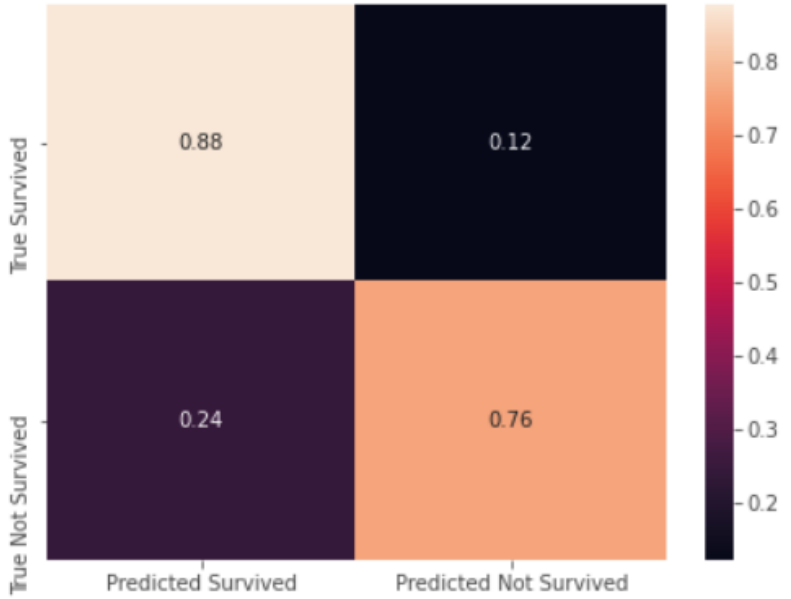
Hyperparameters:

- n_estimators = 150
- max_depth = 1
- Learning_rate = 1

Confusion Matrix (values)



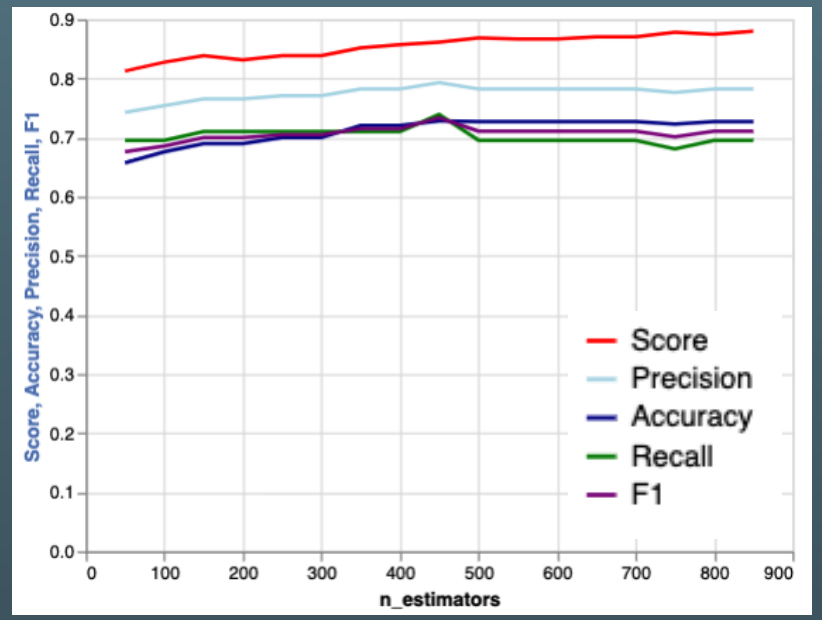
Confusion Matrix (%)

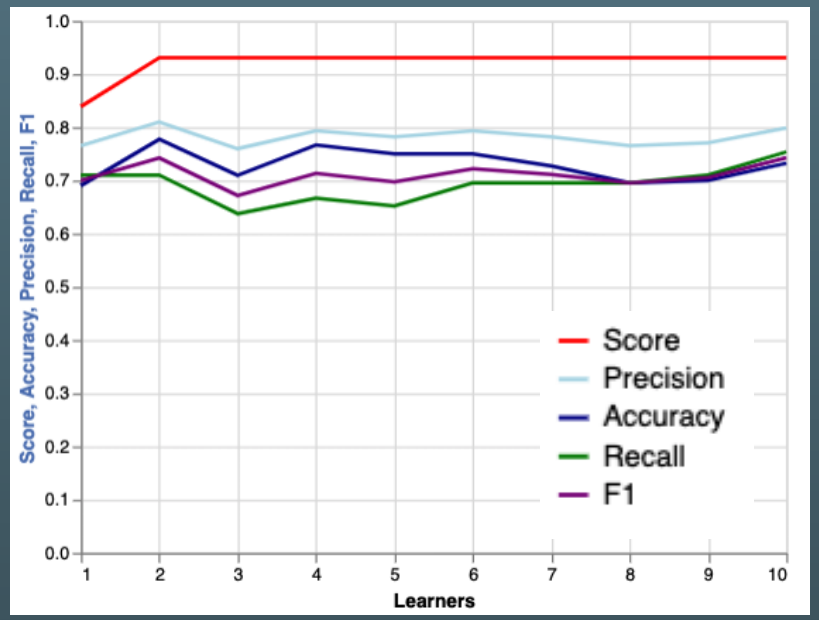


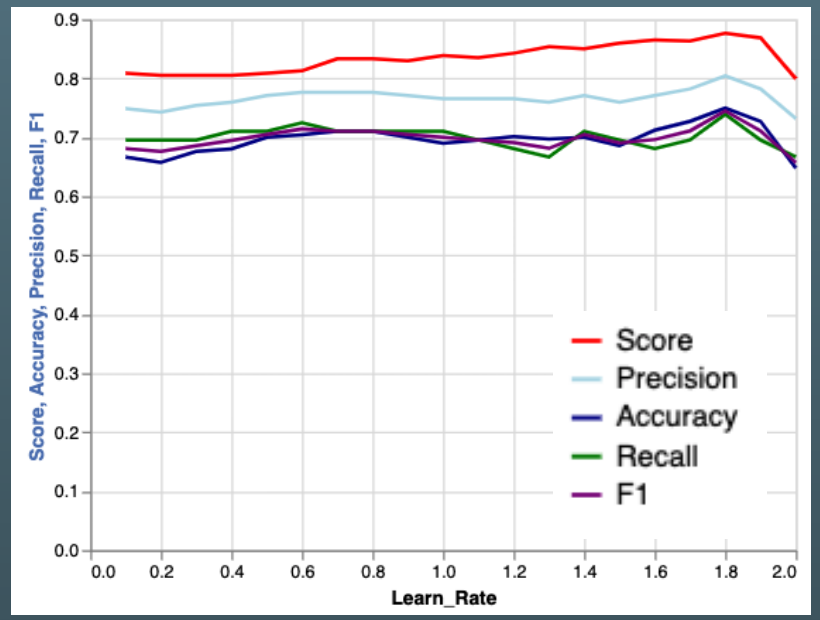
Model Tuning: AdaBoost

Try different algorithms on the training data and show confusion matrix

Optimized Hyperparameters:







- n_estimators = 450
- max_depth: 1
- learning_rate = 1.8

Model Tuning: Optimized AdaBoost

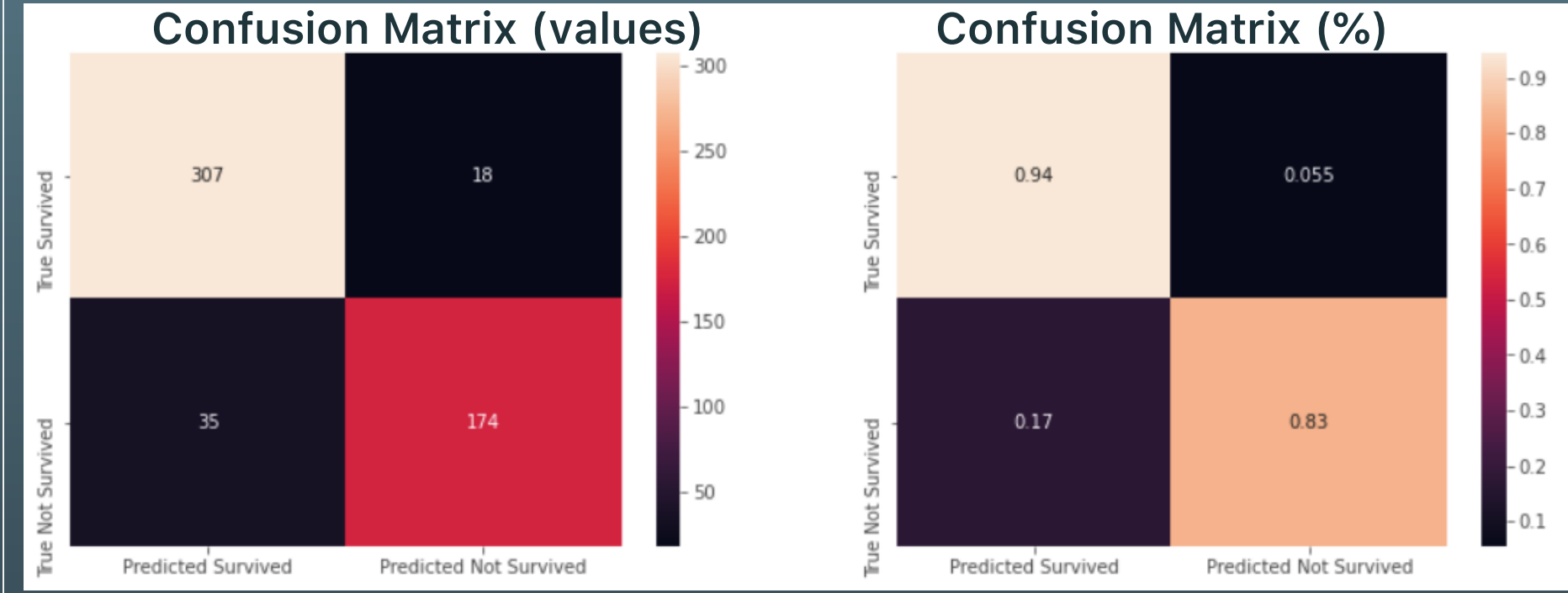
Run model with optimal hyperparameters

kNN Scores:

- Model Score: 90% (+7)
- Accuracy: 79% (+3)
- Precision: 74% (+5)
- Recall: 72% (+1)
- F1: 73% (+3)

Hyperparameters:

- n_estimators = 450
- max_depth = 1
- learn_rate = 1.8



Model Validation: AdaBoost

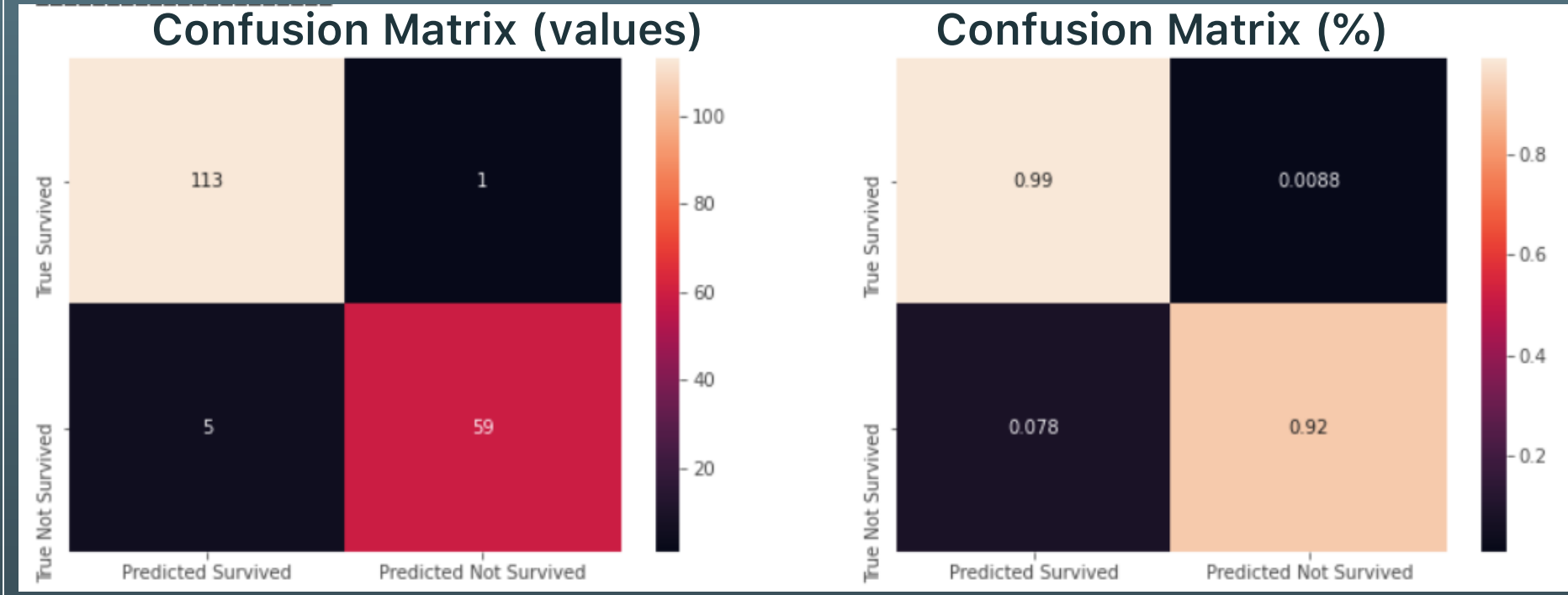
Run model with optimal hyperparameters on Validation data set

AdaBoost Scores:

- Model Score: 91% (+6)
- Accuracy: 72% (+3)
- Precision: 66% (+5)
- Recall: 64% (+1)
- F1: 65% (+3)

Hyperparameters:

- n_estimators = 450
- max_depth = 1
- learn_rate = 1.8



Model Testing: AdaBoost

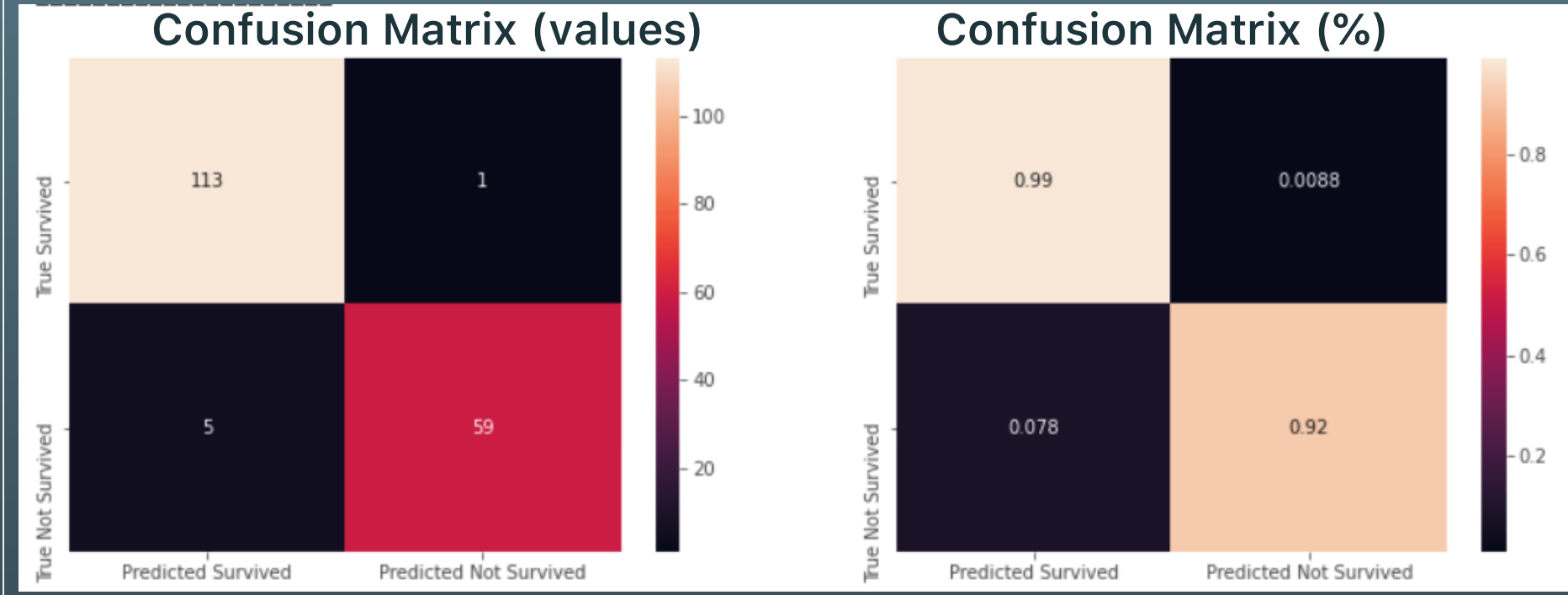
Run model with optimal hyperparameters on Test data set

AdaBoost Scores:

- Model Score: 91% (+6)
- Accuracy: 72% (+3)
- Precision: 66% (+5)
- Recall: 64% (+1)
- F1: 65% (+3)

Hyperparameters:

- n_estimators = 450
- max_depth = 1
- learn_rate = 1.8



Predicting Survivability on the Titanic

Conclusion

The model is able to survivability with an accuracy of