A Mini Project Report on

"Who Survived the Titanic shipwreck Prediction using Machine Learning"

Submitted *by*

Mr. Aditya Manish Limje (Roll No: 63)

submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor

in

COMPUTER ENGINEERING

For Academic Year 2023-2024

Under the guidance of

Prof. K.P.Birla

DEPARTMENT OF COMPUTER ENGINEERING

K.K.Wagh Institute of Engineering Education and Research,
Nashik – 422003

Certificate

This is to Certify that

Mr. Ankit Manish Limje (Roll No: 63)

has completed the necessary Mini Project work and prepared the report on

“Who Survived the Titanic shipwreck Prediction using Machine Learning”

in satisfactory manner as a fulfillment of the requirement of the award of degree of Bachelor of Computer Engineering in the Academic year 2023-2024

Prof.K.P.Birla ***Prof.Dr.S.S.Sane***

Project Guide ***Head of Department***

Acknowledgements

Every work is source which requires support from many people and areas. It gives us proud privilege to complete the Machine Learning Mini Project on “Who Survived the Titanic shipwreck Prediction using Machine Learning**”** under valuable guidance and encouragement of our guide **Prof. K.P.Birla.**

At last we would like to thank all the staff members and our students who directly or indirectly supported me without which the Mini Project work would not have been completed successfully.

by

Mr. Aditya Manish Limje

Table of Contents

Introduction

The goal of the project was to predict the survival of passengers based off a set of data. We used Kaggle competition "Titanic: Machine Learning from Disaster" (see https://www.kaggle.com/c/titanic/data) to retrieve necessary data and evaluate accuracy of our predictions. The historical data has been split into two groups, a 'training set' and a 'test set'. For the training set, we are provided with the outcome (whether or not a passenger survived). We used this set to build our model to generate predictions for the test set.

For each passenger in the test set, we had to predict whether or not they survived the sinking. Our score was the percentage of correctly predictions. In our work, we learned Programming language Python and its libraries NumPy (to perform matrix operations) and SciKit-Learn (to apply machine learning algorithms) Several machine learning algorithms (decision tree, random forests, extra trees, linear regression) Feature Engineering techniques We used Online integrated development environment Cloud 9 (https://c9.io) Python 2.7.6 with the libraries numpy, sklearn, and matplotlib Microsoft Excel

Work Plan

1. Learn programming language Python

2. Learn Shennon Entropy and write Python code to compute Shennon Entropy

3. Get familiar with Kaggle project and try using Pivot Tables in Microsoft Excel to analyze the data.
4. Learn to use SciKit-Learn library in Python, including
    a. Building decision tree
    b. Building Random Forests
    c. Building ExtraTrees
    d. Using Linear Regression algorithm

5. Performing Feature Engineering, applying machine learning algorithms, and analyzing results

Training and Test Data

Training and Test data come in CSV file and contain the following fields:

Passenger ID

Passenger

Class Name

Sex

Age

Number of passenger's siblings and spouses on board

Number of passenger's parents and children on board Ticket

Fare

Cabin

City where passenger embarked

```
#Loading Datasets
pd.set_option('display.max_columns',10,'display.width',1000)
train = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')
train.head()
```

	PassengerId	Survived	Pclass	Name	Sex	...	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	...	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	...	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	...	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	...	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	...	0	373450	8.0500	NaN	S

5 rows × 12 columns

```
#Description of dataset
train.describe(include="all")
```

	PassengerId	Survived	Pclass	Name	Sex	...	Parch	Ticket	Fare	Cabin	Embarked
count	891.000000	891.000000	891.000000	891	891	...	891.000000	891	891.000000	204	889
unique	NaN	NaN	NaN	891	2	...	NaN	681	NaN	147	3
top	NaN	NaN	NaN	Braund, Mr. Owen Harris	male	...	NaN	347082	NaN	B96 B98	S
freq	NaN	NaN	NaN	1	577	...	NaN	7	NaN	4	644
mean	446.000000	0.383838	2.308642	NaN	NaN	...	0.381594	NaN	32.204208	NaN	NaN
std	257.353842	0.486592	0.836071	NaN	NaN	...	0.806057	NaN	49.693429	NaN	NaN
min	1.000000	0.000000	1.000000	NaN	NaN	...	0.000000	NaN	0.000000	NaN	NaN
25%	223.500000	0.000000	2.000000	NaN	NaN	...	0.000000	NaN	7.910400	NaN	NaN
50%	446.000000	0.000000	3.000000	NaN	NaN	...	0.000000	NaN	14.454200	NaN	NaN
75%	668.500000	1.000000	3.000000	NaN	NaN	...	0.000000	NaN	31.000000	NaN	NaN
max	891.000000	1.000000	3.000000	NaN	NaN	...	6.000000	NaN	512.329200	NaN	NaN

11 rows × 12 columns

Feature Engineering

Since the data can have missing fields, incomplete fields, or fields containing hidden information, a crucial step in building any prediction system is Feature Engineering. For instance, the fields Age, Fare, and Embarked in the training and test data, had missing values that had to be filled in. The field Name while being useless itself, contained passenger's Title (Mr., Mrs., etc.), we also used passenger's surname to distinguish families on board of Titanic. Below is the list of all changes that has been made to the data.

Extracting Title from Name

The field Name in the training and test data has the form "Braund, Mr. Owen Harris". Since name is unique for each passenger, it is not useful for our prediction system. However, a passenger's title can be extracted from his or her name. We found 10 titles:

Index	Title	Number of occurrences
0	Col.	4
1	DR.	8
2	Lady	4
3	Master	61
4	Miss	262
5	Mr.	757
6	Mrs.	198
7	Ms.	2
8	Rev.	8
9	Sir	5

We can see that title may indicate passenger's sex (Mr. vs Mrs.), class (Lady vs Mrs.), age (Master vs Mr.), profession (Col., Dr., and Rev.).

Calculating Family Size

It seems advantageous to calculate family size as follows

Family_Size = Parents_Children + Siblings_Spouses + 1

Extracting Deck from Cabin

The field Cabin in the training and test data has the form "C85", "C125", where C refers to the deck label. We found 8 deck labels: A, B, C, D, E, F, G, T. We see deck label as a refinement of the passenger's class field since the decks A and B were intended for passengers of the first class, etc.

Extracting Ticket_Code from Ticket

The field Ticket in the training and test data has the form "A/5 21171". Although we couldn't understand meaning of letters in front of numbers in the field Ticket, we extracted those letters and used them in our prediction system. We found the following letters

Index	Ticket code	Number of occurrences
0	No code	961
1	A	42
2	C	77
3	F	13
4	L	1
5	P	98
6	S	98
7	W	19

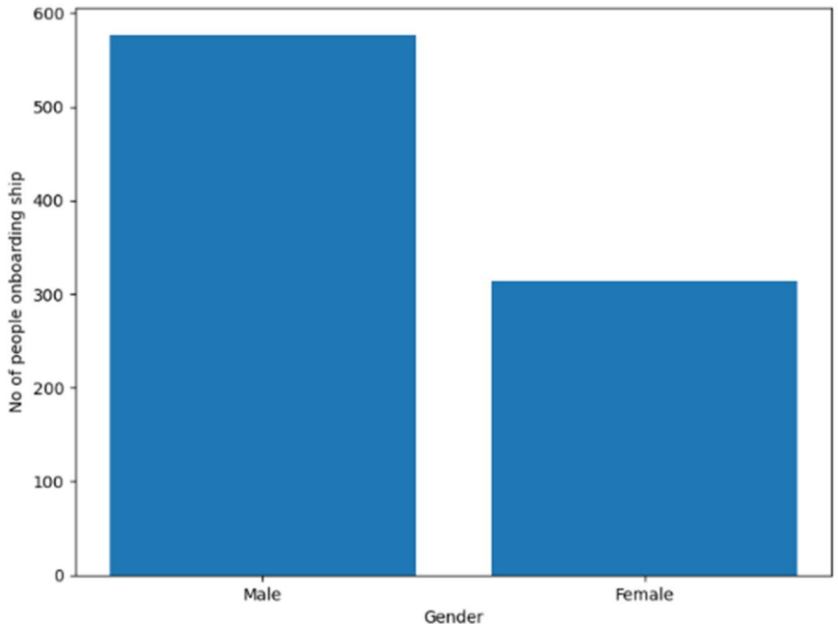
Filling in missing values in the fields Fare, Embarked, and Age

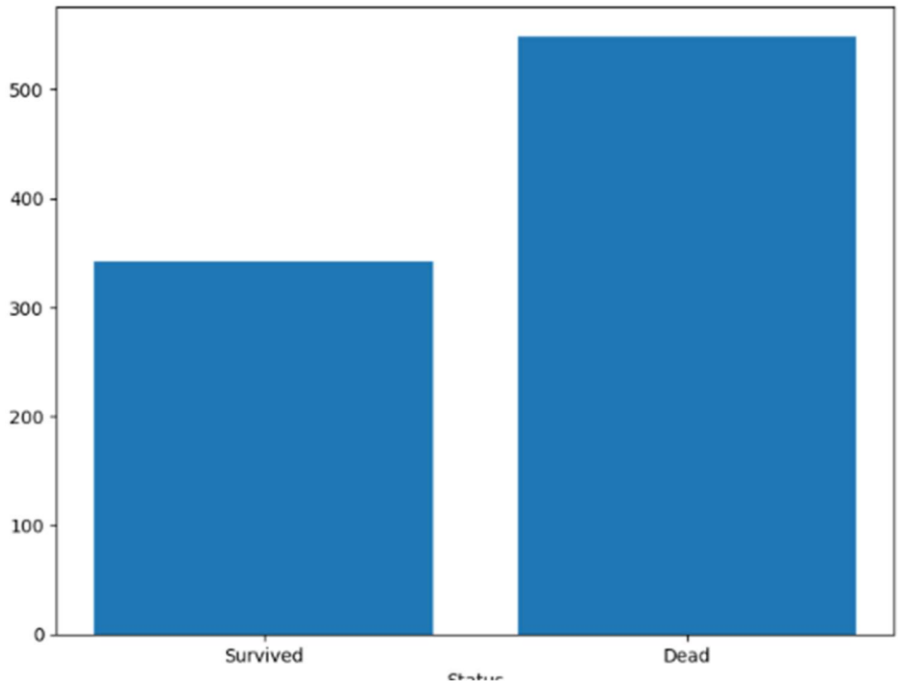
Since the number of missing values was small, we used median of all Fare values to fill in missing Fare fields, and the letter 'S' (most frequent value) for the field Embarked.

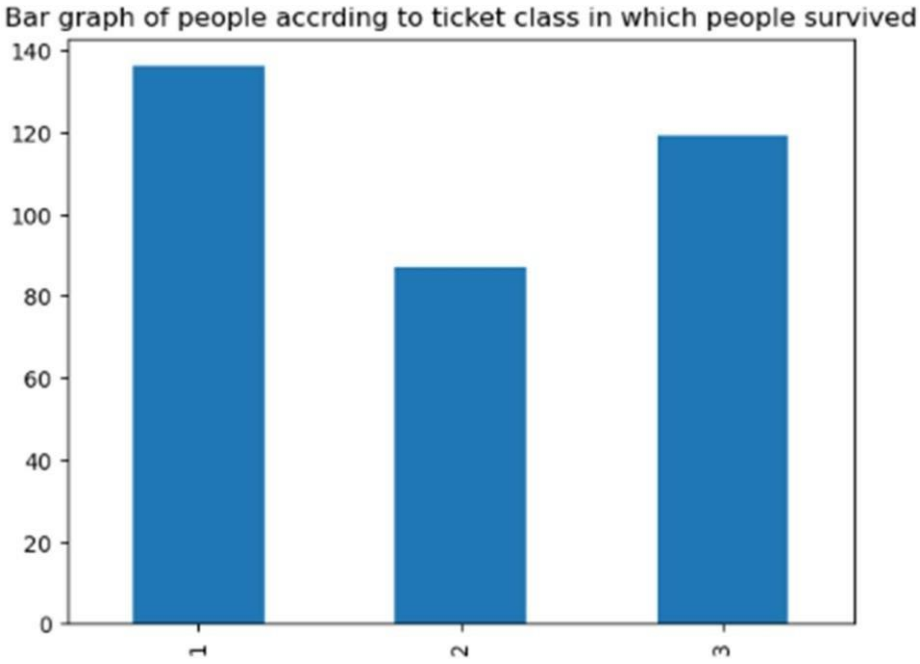
In the training and test data, there was significant amount of missing Ages. To fill in those, we used Linear Regression algorithm to predict Ages based on all other fields except Passenger_ID and Survived.

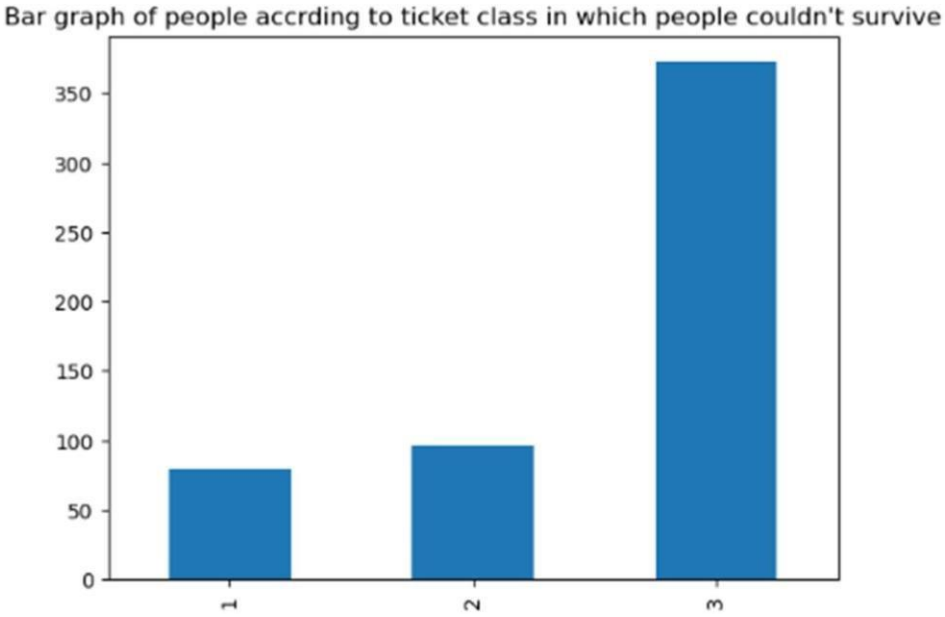
Importance of fields

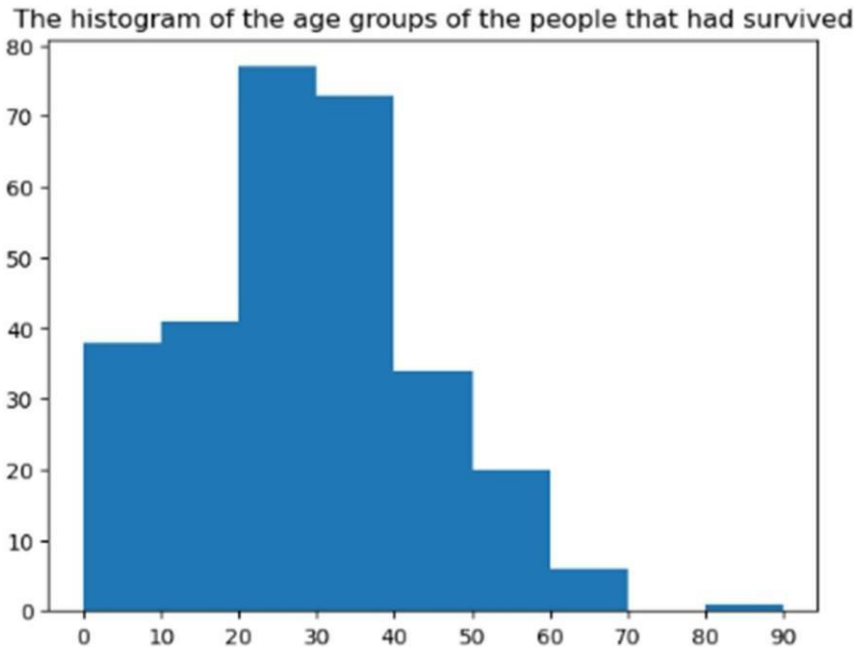
Decision Trees algorithm in the library SciKit-Learn allows to evaluate importance of each field used for prediction. Below is the chart displaying importance of each field.

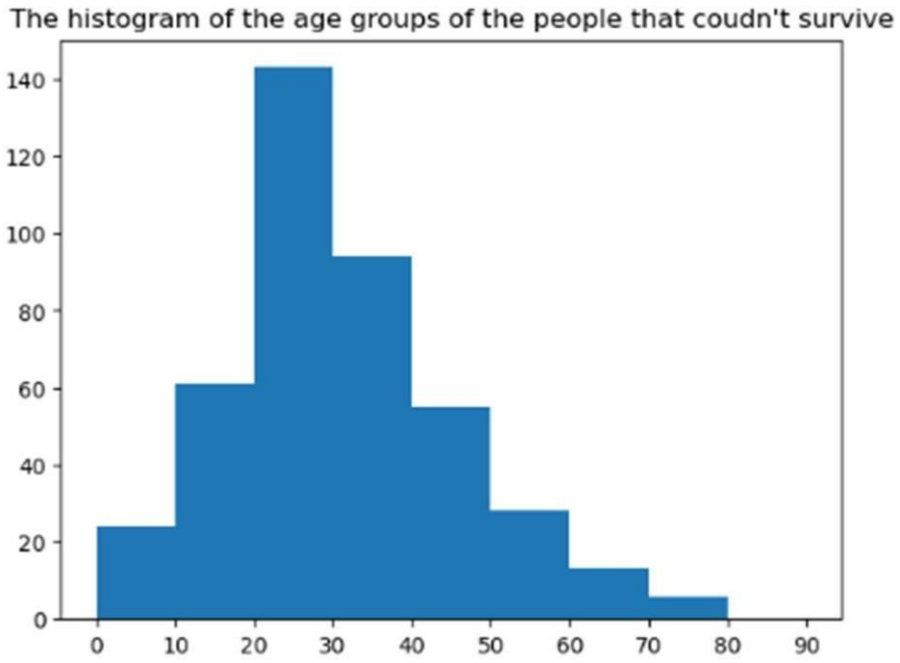


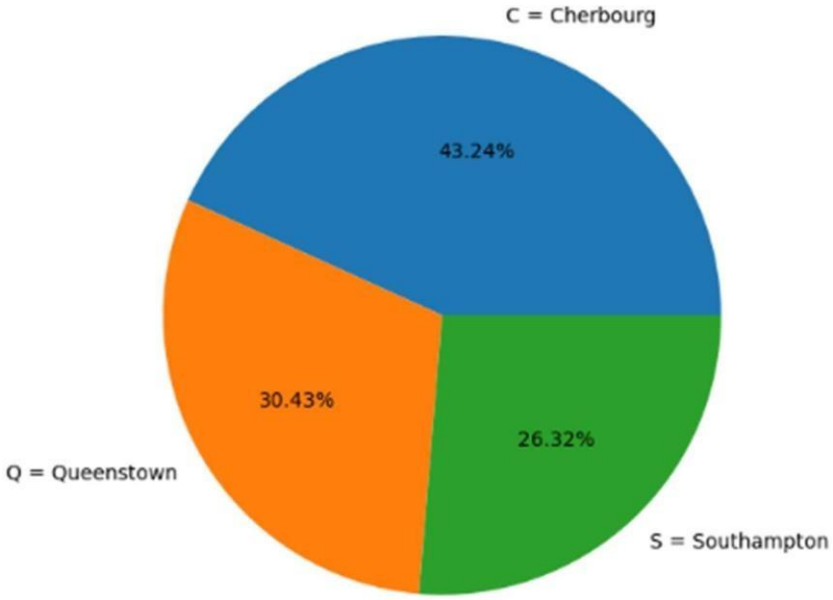

500
400
300
200
100
0
Survived
Dead

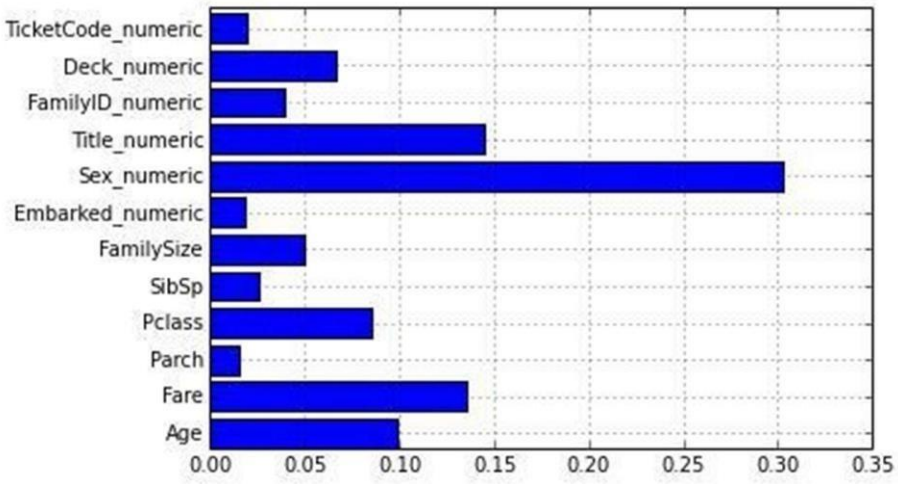

Bar graph of people accrding to ticket class in which people survived
140
120
100
80
60
40
20
0
1
2
3


Bar graph of people accrding to ticket class in which people couldn't survive
350
300
250
200
150
100
50
0
1
2
3


The histogram of the age groups of the people that had survived
80
70
60
50
40
30
20
10
0
0
10
20
30
40
50
60
70
80
90


The histogram of the age groups of the people that coudn't survive
140
120
100
80
60
40
20
0
0
10
20
30
40
50
60
70
80
90

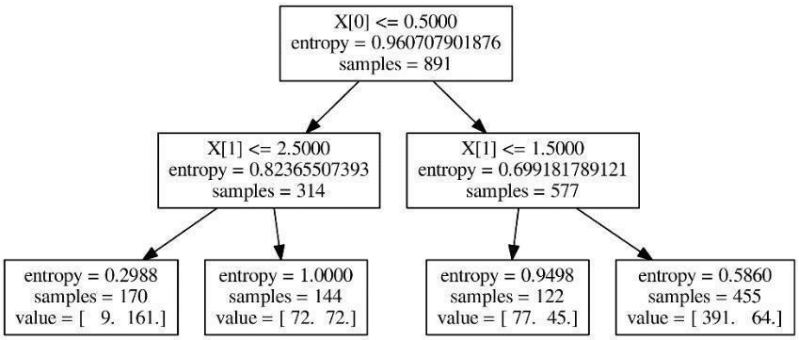

C = Cherbourg
43.24%
30.43%
Q = Queenstown
26.32%
S = Southampton



We can see that the field Sex is the most important one for prediction, followed by Title, Fare, Age, Class, Deck, Family_Size, etc.

Decision Trees

Our prediction system is based on growing Decision Trees to predict the survival status. A typical Decision Tree is pictured



The basic algorithm for growing Decision Tree:

1. Start at the root node as parent node
2. Split the parent node based on field X[i] to minimize the sum of child nodes uncertainty (maximize information gain)
3. Assign training samples to new child nodes
4. Stop if leave nodes are pure or early stopping criteria is satisfied, otherwise repeat step 1 and 2 for each new child node

Stopping Rules:

1. The leaf nodes are pure
2. A maximal node depth is reached
3. Splitting a node does not lead to an information gain

In order to measure uncertainty and information gain, we used the formula

( ) = ( ) − ( ) − h ( )

where

: Information Gain
: Impurity (Uncertainty Measure)
h : number of samples in the parent, the left child, and the right child nodes
h : training subset of the parent, the left child, and the right child nodes

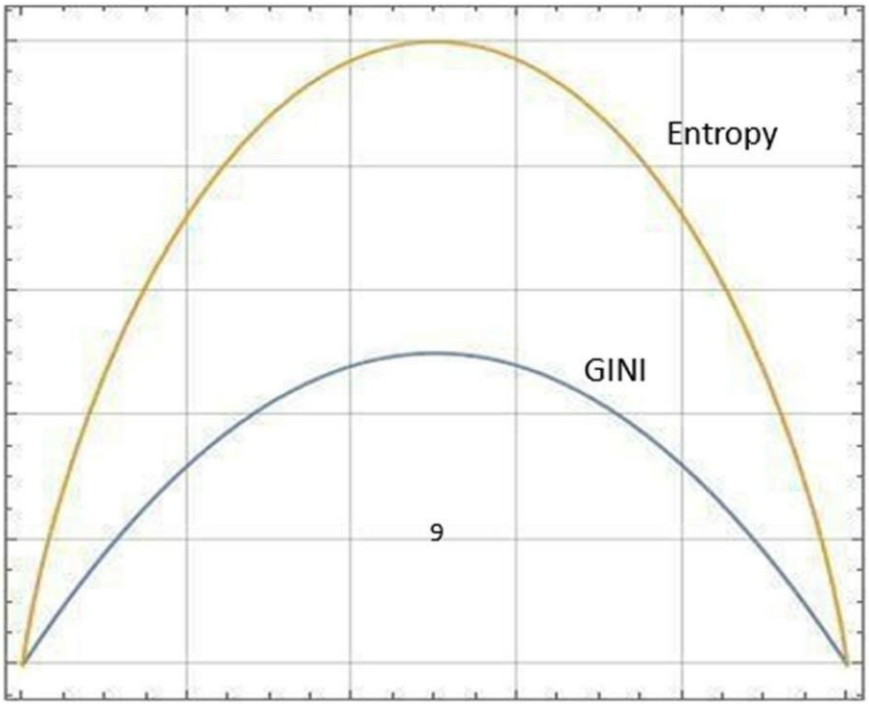
For Uncertainty Measure, we used Entropy defined by

(1, 2) = − 1 log2 1 − 2 log2 2 and

GINI index defined by

(1, 2)=2 1 2

The graphs of both measures are given below



1

We can see on the graph that when probability of an event is 0 or 1, then the uncertainty measure equals to 0, while if probability of an event is close to ½, then the uncertainty measure is maximum.

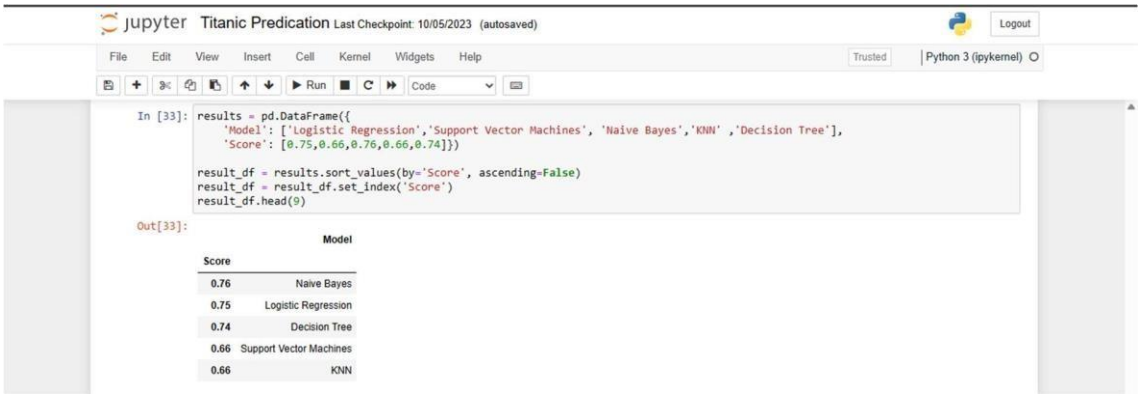
Random Forest and ExtraTrees

One common issue with all machine learning algorithms is Overfitting. For Decision Tree, it means growing too large tree (with strong bias, small variation) so it loses its ability to generalize the data and to predict the output. In order to deal with overfitting, we can grow several decision trees and take the average of their predictions. The library SciKit-Learn provides to such algorithm Random Forest and ExtraTrees.

In Random Forest, we grow N decision trees based on randomly selected subset of the data and randomly selected M fields, where .

In ExtraTrees, in addition to randomness of subsets of the data and of field, splits of nodes are chosen randomly.

Result



Score	Model
0.76	Naive Bayes
0.75	Logistic Regression
0.74	Decision Tree
0.66	Support Vector Machines
0.66	KNN

Conclusion

As a result of our work, we gained valuable experience of building prediction systems and achieved our best score on Kaggle: 80.383% of correct predictions (in Kaggle leaderboard, it corresponds to positions 477 - 881 out of 3911 participants).

- We performed featured engineering techniques
  - Changed alphabetic values to numeric
  - Calculated family size
  - Extracted title from name and deck label from ticket number
  - Used linear regression algorithm to fill in missing ages

- We used several prediction algorithms in python
  - Decision tree
  - Random forests
  - Extra trees

- We achieved our best score 80.383% correct predictions