A Mini Project Report

on

“Who Survived the Titanic shipwreck Prediction using Machine Learning”

by

Samir Hasan Shaikh (.)

Kalyani Arjun Sansare (.)

Lina Pravin Birari (.)

Under the guidance of

Prof. Puspendu Biswas

Department of Computer

Engineering

S.M.E.S. Sanghavi College

of Engineering,Nashik

SAVITRIBAI PHULE PUNE UNIVERSITY

2022_2023

Department of Computer Engineering

S.M.E.S. Sanghavi College of Engineering

Date:

CERTIFICATE

This is to certify that, Samir Hasan Shaikh (.) Kalyani Arjun Sansare (.) Lina Pravin Birari (.) of class **T.E Computer**; have successfully completed their mini project work on **“Who Survived the Titanic shipwreck Prediction using Machine Learning”** at **Institute of Technology,Management & research, Nashik** in the partial fulfillment of the Graduate Degree course in **B.E** at the department of **Computer Engineering** in the academic Year 2022-2023 Semester – I as prescribed by the Savitribai Phule PuneUniversity.

{ Prof. Puspendu Biswas }	**{ Prof. Puspendu Biswas }**
Project Guide	**Head of Department**

Acknowledgements

With deep sense of gratitude we would like to thanks all the people who have lit our path with their kind guidance. We are very grateful to these intellectuals who did their best to help during our project work.

It is our proud privilege to express deep sense of gratitude to **Prof. A.D. Lokhande,** Principal Sanghvi college of engineering, Nashik for his comments and kind permission to complete this project. We remain indebted to Prof. Puspendu Biswas, H.O.D. of Computer Engineering Department for his timely suggestion and valuable guidance.

The special gratitude goes to Prof. Puspendu Biswas excellent and precious guidance in completion of this work .We thanks to all the colleagues for their appreciable help for our working project. With various industry owners or lab technicians to help, it has been our endeavor to throughout our work to cover the entire project work.

We also thankful to our parents who providing their wishful support for our project completion successfully .And lastly we thanks to our all friends and the people who are directly or indirectly related to our project work.

Samir Hasan Shaikh (.)

Kalyani Arjun Sansare (.)

Lina Pravin Birari (.)

CONTENTS

Abstract

The sinking of the RMS Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. This sensational tragedy shocked the international community and led to better safety regulations for ships.

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

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:

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 SizeIt seems advantageous to calculate family size as follows Family_Size = Parents_Children Siblings_Spouses + 1

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

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

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

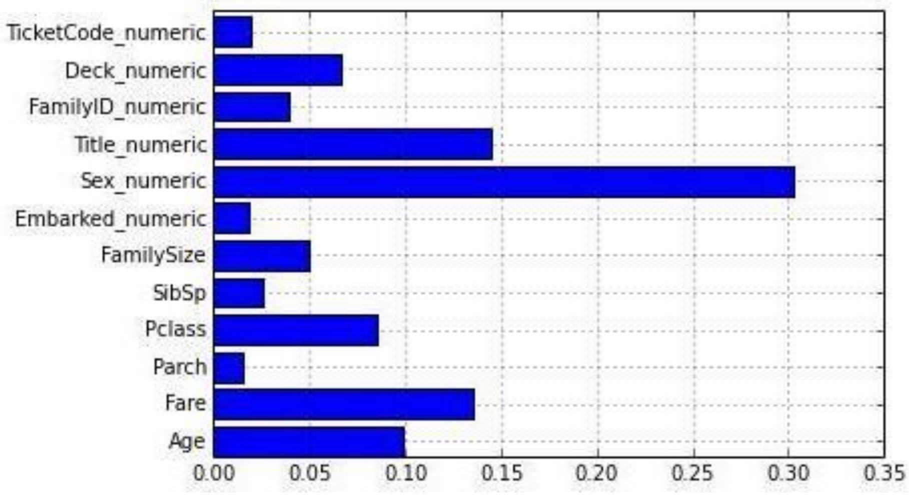
Since the number of missing values was small, we used median of all Fare values to fill in missing Fare fields, and the letter 'S' (most frequent value) for the field Embarked.

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.

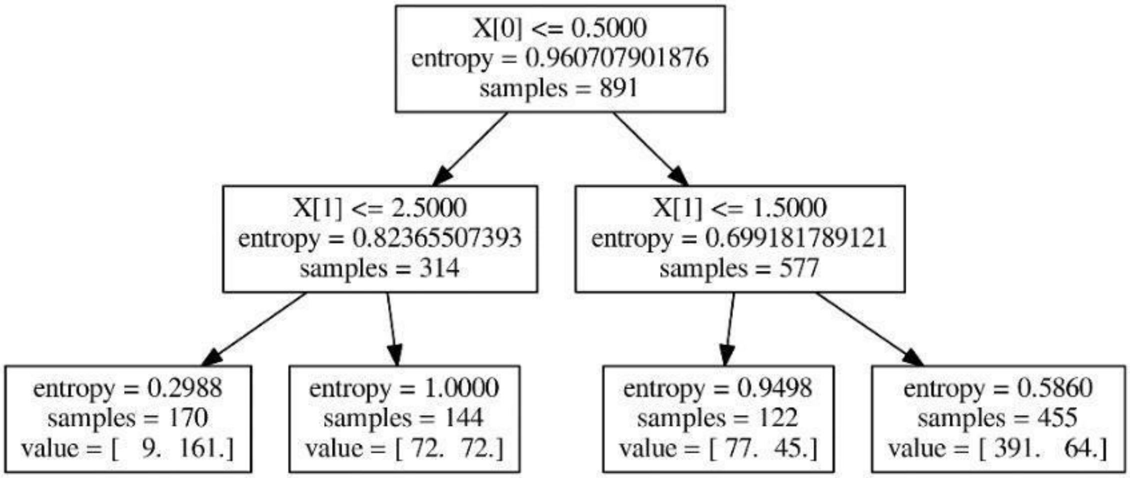
Index	Ticket Code	Number of occurrences
0	No Code	961
1	A	42
2	C	77
3	F	13
4	L	1
5	P	98
6	S	98
7	W	19



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 below



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

Extra Trees and **Random Forest**

Therefore, **trees ensemble methods are better than simple decision trees**, but is an ensemble better than the other? Which one should I use? This post is going to try to answer these questions, studying the differences between the Extra Trees and Random Forest and comparing both of them in terms of results. The two ensembles have a lot in common. Both of them are **composed of a large number of decision trees**, where the final decision is obtained taking into account the prediction of every tree. Specifically, by majority vote in classification problems, and by the arithmetic mean in regression problems. Furthermore, both algorithms have the same growing tree procedure (with one exception explained below). Moreover, when selecting the partition of each node, both of them randomly choose a subset of features.

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

References

[1] Geurts, Pierre, Damien Ernst, and Louis Wehenkel. “Extremely randomized trees.” Machine learning 63.1 (2006): 3-42.

Adebayo, Julius, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. “Sanity checks for saliency maps.” arXiv preprint arXiv:1810.03292 (2018).

Alain, Guillaume, and Yoshua Bengio. “Understanding intermediate layers using linear classifier probes.” arXiv preprint arXiv:1610.01644 (2016).

Alber, Maximilian, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof T. Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Sven Dähne, and Pieter-Jan Kindermans. “iNNvestigate neural networks!.” J. Mach. Learn. Res. 20, no. 93 (2019): 1-8.

Alberto, Túlio C, Johannes V Lochter, and Tiago A Almeida. “Tubespam: comment spam filtering on YouTube.” In Machine Learning and Applications (Icmla), Ieee 14th International Conference on, 138–43. IEEE. (2015).

Alvarez-Melis, David, and Tommi S. Jaakkola. “On the robustness of interpretability methods.” arXiv preprint arXiv:1806.08049 (2018).