Titanic passengers and crew survival classification

Egor Zuev – 1020418
Email: egor.zuev@studio.unibo.it

***Abstract*—This project utilizes the Titanic dataset from the R language package 'stablelearner', which contains information about passengers and crew members, including their age, sex, class, fare, and survival status. By leveraging different Machine Learning techniques such as K nearest neighbours, logistic regression, decision tree, random forest, gradient boosting, and MLP, the project aims to build accurate models that can classify passengers into survivors and non-survivors.**

1. Introduction

The Titanic tragedy that happened on April 15, 1912, was one of the most devastating maritime disasters in history, where many lives were lost due to various factors such as inadequate safety measures, limited lifeboats, and the severity of the collision with the iceberg. [1]

This report focuses on utilizing various machine learning techniques to classify the Titanic crew members and passengers into those who survived the crash and those who did not. The report begins by discussing data preparation and understanding, followed by an analysis of the utilized algorithms. Finally, the report concludes with a comparison of the algorithms' performance.

2. Data understanding

In this section, we will explore the Titanic data set, focusing on comprehensive data understanding and preprocessing steps to prepare it for machine learning analysis. Moreover, analyzing the data structure provides insights on how to effectively work with the dataset, uncovering patterns and facilitating further analysis.

2.1. Demography of passengers

In this paragraph, the paper looks into the main characteristics of the entire data set. By examining various groups based on gender, age, and onboard status (crew or passenger), a more comprehensive understanding can be gained of the data and the parameters with the strongest influence on survival. The total number of the people onboard, at least the ones we have the knowledge about, was 2207, out of which 1496 (or 68%) died. In this table, 'M' refers to male passengers, and 'F' to female ones. '18-', '18-60', and '60+' stand for the age groups. 'P' and 'C' are Passengers and Crew members respectively. '%' refers to the survival rate in the group.

TABLE 1: People onboard demography

Survived	M	F	18-	18-60	60+	P	C
Yes	352	359	86	617	8	498	213
No	1366	130	112	1351	31	793	703
%	20.49	73.42	43.43	31.35	20.51	38.57	23.25
Total	1718	489	198	1968	39	1291	916

As evidenced by the table, women and children were more likely to survive and the crew members had a lower survival rate in comparison to the passengers. This is probably due to the fact that they were given priority when using lifeboats. The number of people over 60 years old is relatively low and they had a low survival rate. Also it is clear that women, children and old people are relatively small groups if compared to men and people in the age from 18 to 60 years respectively.

2.2. Dropping Columns

To remove irrelevant information, 'name', 'ticketno', 'country' columns were removed from the data set. Obviously, a person's name and ticket number can not influence their survival chance. While the country of origin of the person may be an important factor because of easier communication with people who speak the same language, keeping this column makes the data set sparse which negatively affects the algorithms' performance.

2.3. Describing Numeric Data

Next, a descriptive analysis of the numeric variables in the train data set was performed. Since it is created randomly as a 70% part of the whole data set, it can be assumed that the numbers are the same.

This analysis involved computing basic statistical measures such as mean, standard deviation, minimum, maximum, and quartiles for variables such as age, fare, "sibsp" (number of siblings, spouses aboard), and "parch" (number of parents/children aboard). It can be seen that the count for the age column is higher than the count for other columns since columns `sibsp` and `parch` are not present for crew members and `fare` is `NA` for them. It is also evident that passengers were mostly "young": half of them were 29 years or younger. In addition to that, the major part of the tickets sold had a lower fare; most of the passengers traveled with zero or one family member (spouse, children, or parents).

TABLE 2: Summary Statistics

	Age	Fare	SibSp	Parch
Count	1542.000	901.000	910.000	910.000
Mean	30.684	33.143	0.485	0.388
Std	12.165	52.277	1.048	0.901
Min	0.417	4.000	0.000	0.000
25%	22.000	7.181	0.000	0.000
50%	29.000	14.090	0.000	0.000
75%	38.000	31.051	1.000	0.000
Max	72.000	512.061	8.000	9.000

2.4. Simplifying Survival Field

To simplify the data processing, the `Survived` field, which originally contained 'yes' and 'no' values, was converted to Boolean values. 'Yes' was replaced with `True`, indicating survival, while 'No' was replaced with `False`. This transformation was performed in order to facilitate code writing.

2.5. Encoding Gender

As known, pairplot plots only numerical data, so to include the gender into the visualization we encoded the `gender` variable. We replaced 'male' with 1 and 'female' with 0, assigning numerical values to represent the gender categories.

2.6. Pairplot Visualization

Finally, a pairplot was created using the preprocessed data set. A pairplot is a graphical representation that showcases the relationships between variables by plotting scatter plots for numeric variables and histograms for categorical variables. This visualization allows us to observe patterns, correlations, and potential insights regarding survival based on different features in the data set. It is clearly evidenced by the graphs that women tended to survive significantly more often then men. The age – age graph demonstrates that children tended to survive more often because at approximately 15 years old the distribution bell goes up more sharply than before 15 years. The major part of both survivals and non-survivals were less then 50 years old which corresponds to the Titanic's demography.

Fare - age graph demonstrates that from 100 pounds fare people tended to survive significantly more often regardless of age. However, these people were statistically few.

In sibsp - sibsp and parch - parch graphs there are peaks of survivals that correspond to higher survival rates. From this, it can be concluded that people with their family members onboard survived more frequently than the ones without them.

2.7. Data prepocessing for the ML models

In the beginning standard operations are performed, such as

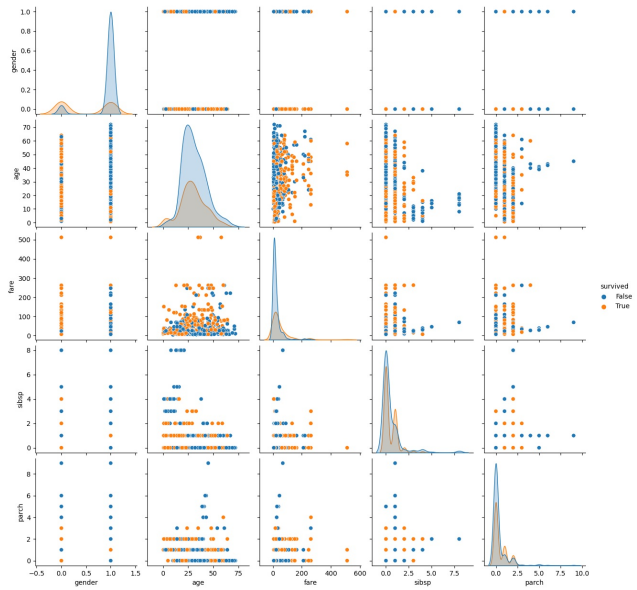

Figure 1: Pairplot Visualization

- creating 'train target' with the survival classification and dropping this data from 'train' dataset
- filling NA values with a mean value in both 'train' and 'test' datasets
- the categorical features are processed using One-HotEncoder, numerical values are transformed with StandardScaler

2.8. PCA Visualisation

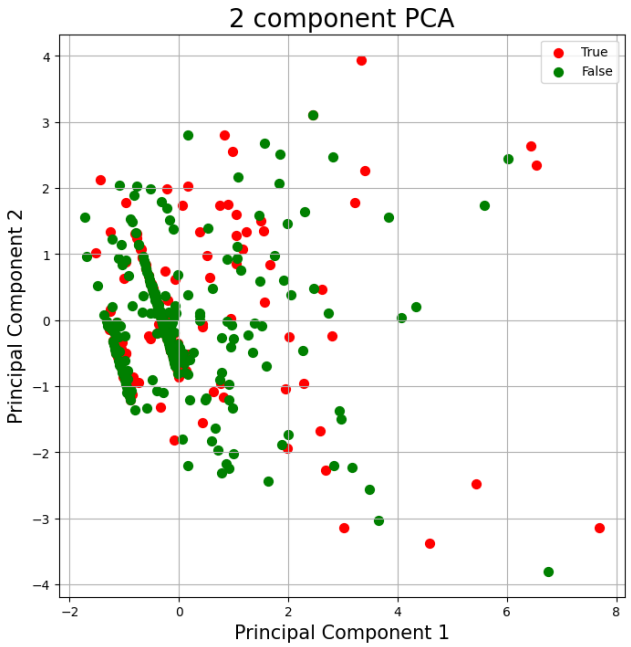

Figure 2: PCA Visualization

Principal Component Analysis (PCA) is performed to reduce the dimensionality of high-dimensional data while preserving its essential structure. It identifies the directions (principal components) along which the data varies the most, allowing for a compact representation and visualization of the data.

As we can see, there are two green lines of non-survivals on the PCA graph that cover two similar red lines. Since there were more non-survivals there are more green dots on the graph. We can speculate that these two lines are probably connected to gender and age distribution.

3. Models utilized

3.1. How the models are evaluated

After completing the data preprocessing, the following steps are taken to evaluate each model:

- Creating a grid of parameters based on the most commonly used ones. This grid allows for exploring different combinations of model hyperparameters during the evaluation process.
- The model is trained on the preprocessed "train data" iterating through the specified hyperparameters.
- The model with the best hyperparameters is passed to the `evaluate` function

3.2. Evaluation function

The `evaluate` function is taking the model as a function parameter, then the model is applied to test data. The best grid search parameters are printed, as well as the TPR, TNR, FPR, FNR, Accuracy, Precision, Recall, F1 Score parameters

3.3. KNN

K-Nearest Neighbors (KNN) is an algorithm used for both classification and regression tasks. It assigns a new data point to the majority class of its k nearest neighbors in the feature space, making predictions based on the local neighborhood's characteristics. [2]

TABLE 3: KNN Parameter Grid and Best Values

Parameter	Grid	Best Value
n_neighbors	[1, 30]	5
weights	uniform, distance	uniform
metric	euclidean, manhattan, chebyshev	euclidean
algorithm	ball_tree, kd_tree, brute	ball_tree

3.4. Logistic regression

Logistic Regression is an algorithm used for binary classification tasks. It models the relationship between the input features and the probability of belonging to a particular class, employing a logistic function to transform the linear regression output into a probability score. [3]

TABLE 4: Logistic Regression Parameter Grid and Best Values

Parameter	Grid	Best Value
C	np.logspace(-4, 4, 20)	1.623
solver	newton-cg, lbfgs, liblinear, sag, saga	saga
class_weight	None, balanced	None
penalty	l1, l2, elasticnet, none	elasticnet
l1_ratio	np.arange(0.1, 0.9, 0.05)	0.30

3.5. Desicion tree

A decision tree is an ML algorithm that uses a hierarchical structure of nodes to make decisions based on input features. It recursively partitions the data based on feature thresholds to create a tree-like structure. Each leaf node represents a class label or outcome, making decision trees useful for both classification and regression tasks. [4]

TABLE 5: Decision Tree Parameter Grid and Best Values

Parameter	Grid	Best Value
criterion	gini, entropy	entropy
splitter	random, best	random
min_samples_split	np.arange(2, 15)	11
class_weight	balanced, None	None
ccp_alpha	np.arange(0, 0.9, 0.1)	0.0

3.6. Random forest

Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. It creates a forest of trees by randomly selecting subsets of features and data samples. Each tree independently predicts the outcome, and the final prediction is determined through voting or averaging. [5]

TABLE 6: Random Forest Parameter Grid and Best Values

Parameter	Grid	Best Value
criterion	gini, entropy	entropy
class_weight	balanced,balanced_subsample,None	balanced_subs.
n_estimators	np.arange(10, 150, 10)	70
bootstrap	True, False	True
max_samples	np.arange(0.1, 1.0, 0.1)	0.2

3.7. Gradient boosting

Gradient Boosting is a machine learning technique that combines weak prediction models, typically decision trees, in an iterative manner to create a strong predictive model. It works by sequentially fitting new models to the residuals of the previous models, with each subsequent model focusing on reducing the errors made by the previous models. [6]

3.8. MLP Classifier

The MLP classifier (Multilayer Perceptron classifier), is a neural network model used for classification tasks in

TABLE 7: Gradient Boosting Parameter Grid and Best Values

Parameter	Grid	Best Value
loss	deviance, exponential	deviance
n_estimators	np.arange(10, 150, 10)	130
subsample	np.arange(0.1, 1.0, 0.1)	0.8
criterion	friedman_mse, squared_error, mse, mae	squared_error

machine learning. It consists of interconnected layers of nodes that perform weighted sums and activation functions to learn the patterns in data. [7]

TABLE 8: MLP Classifier Parameter Grid and Best Values

Parameter	Grid	Best Value
hidden_layer_sizes	(50,), (100,), (50,50,), (100,100,)	(100,)
activation	relu, tanh, logistic	tanh
solver	adam, sgd	adam
alpha	0.0001, 0.001, 0.01	0.01
learning_rate	constant, adaptive	constant

4. Conclusion

In conclusion, the project aimed to utilize different machine learning models to classify Titanic passengers into survivals and non-survivals. We can see the final results in the tables below. The first one compares the accuracy of obtained models on the train data set and the test data set.

As predicted, models showed better results on the train data set, than on a test data set. More than that, there is no overfitting and results do not differ dramatically. It means that the data was divided into the train and test data sets correctly and the models were also configured in the right way.

TABLE 9: Accuracy of ML Models

Model	Train Data Set	Test Data Set
KNN	0.807	0.775
Logistic Regression	0.808	0.769
Decision Tree	0.797	0.759
Random Forest	0.812	0.762
Gradient Boosting	0.818	0.793
MLP Classifier	0.819	0.769

Now let's discuss the performance metrics table. As we can see, KNN and Gradient Boosting have the highest True Positive Rate of all models, which means that they are the best in correctly identifying survivals. However, TPR values lie between 0.5 and 0.6, so the models are only slightly (less than 10%) better than random survival identification.

However, models tend to better classify non-survivals, especially Gradient Boosting and Logistic regression which have scores of 0.901 and 0.872 respectively. We can explain this by the fact that there were more non-survivals (68%) and since the data set is relatively small this plays a crucial role in the models' behavior.

Since the current data set has mostly negative values it is important to consider Precision metric since it measures how effectively the model can identify positive cases and the metric does not has any negatives in the formula unlike True Positive Rate. We can see that Gradient Boosting has significantly higher Precision when compared to other models.

TABLE 10: Performance metrics for different ML models on a test dataset

Model	Metrics						
	TPR	TNR	FPR	FNR	Accr.	Prec.	F1
KNN	0.581	0.868	0.131	0.418	0.775	0.679	0.626
Log. Reg.	0.553	0.872	0.127	0.446	0.769	0.676	0.608
Dec. Tree	0.530	0.868	0.131	0.469	0.758	0.658	0.587
Rand. Forest	0.539	0.868	0.131	0.460	0.761	0.662	0.594
Grad. Boost.	0.567	0.901	0.098	0.432	0.793	0.734	0.640
MLP	0.562	0.868	0.131	0.437	0.769	0.672	0.612

The MLP Classifier has overall high performance, for instance, it has the second highest accuracy rate as well as relatively high F1 score. However, the MLP Classifier is usually used for larger data sets.

In general, KNN has almost the highest performance out of all algorithms (except for Gradient Boosting). We can say that it happens because KNN works well also on smaller data sets and captures non-linear dependencies with not well-defined boundaries, which this classification task is about.

Overall, we can say that Gradient Boosting has the best performance out of all algorithms because it has significantly higher accuracy and precision rather then other algorithms.

References

[1] D. A. Butler, *Unsinkable: The Full Story of RMS Titanic*. Mechanicsburg, PA: Stackpole Books, 1998.

[2] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[3] D. W. Hosmer and S. Lemeshow, *Applied Logistic Regression*, 2nd ed. Wiley, 2000.

[4] M. Studer, G. Ritschard, A. Gabadinho, and N. S. Müller, *Discrepancy Analysis of State Sequences*, *Sociological Methods & Research*, vol. 40, no. 3, pp. 471–510, 2011.

[5] T. K. Ho, *Random Decision Forests*, in *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Montreal, QC, Canada, Aug. 14-16, 1995, pp. 278–282.

[6] S. M. Piryonesi and T. E. El-Diraby, *Data Analytics in Asset Management: Cost-Effective Prediction of the Pavement Condition Index*, *Journal of Infrastructure Systems*, vol. 26, no. 1, Mar. 2020.

[7] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York, NY, 2009.