ISSN 2959-6157

Titanic survival prediction based on machine learning.

Siyu Wu

Abstract:

Using the Titanic guest data offered by Kaggle, we executed data cleansing, feature engineering, and training of various designs, including logistic regression, Random Woodland, XGBoost, and LightGBM versions. We likewise brought out hyperparameter tuning and version combination, and ultimately enhanced the forecast efficiency of the version by heavy average approach. The evaluation results of the model showed that the integrated design surpassed the single design on the test collection, with greater accuracy and ROC AUC ratings.

Keywords: machine learning, Titanic guest data, predict data, model analysis

1. Introduction

The Titanic catastrophe, among the most well-known maritime catastrophes in history, occurred on April 15, 1912. The high-end liner sank after hitting an iceberg on its first trip, eliminating greater than 1,500 people. This occasion not just brought in widespread attention at the time, yet also ended up being the subject of various researches and conversations in the following centuries. An in-depth research of this catastrophe can help to recognize the factors of survival in an emergency and therefore notify future precaution.

Kaggle's Titanic Classification Contest is a classic device finding out job created to predict guests' opportunities of enduring the disaster based on their personal information (such as age, sex, fare, and so on). Individuals construct and optimize anticipating designs to accomplish the most effective anticipating performance on the test collection. The competition not only supplies a hands-on possibility for newbies to artificial intelligence, yet likewise a system for skilled information researchers to demonstrate and verify their technology.

The goal of this study is to evaluate and predict data from Titanic guests using multiple equipment learning algorithms, including logistic regression, Random Forest, XGBoost, and LightGBM. Via information cleaning, function engineering, model adjusting, and combination methods, we intend to boost the accuracy of forecasts and explore crucial factors that affect survival. This report will certainly information the information preprocessing procedure, version choice and training, design evaluation and optimization, and last prediction results and analysis.

2. Introduction of theoretical methods and ideas

2.1 Missing value processing

Missing value processing is a very important process in machine learning. The general model cannot accept missing data as input, and the missing data will also cause the model to lose some effective information [1]. Therefore, it is very necessary to deal with the missing value, and the common processing methods include deleting method, mean filling, median filling, multiple interpolation, etc.

The deletion method deletes all samples that contain missing data. It is more applicable when the amount of data is large and the missing data is not much. If the amount of data is not much, such processing will lose a lot of effective information.

Mean and median fill, that is, use the mean or median data of the total or partial data in the sample to fill in the missing values. This method is very simple, but introduces a lot of bias when the data distribution is skewed.

Iterative interpolation is a more advanced interpolation method that iterates over the data multiple times to predict missing values, while iterating constantly to minimize errors. Multiple interpolation uses regression models to predict missing values based on other features and is suitable for most missing mechanisms, especially when there is a clear correlation between different data [2].

In this exploration, we choose to use the iterative Imputer method to populate the gaps. Given that the characteristics of different people's positions, prices, and ages are not completely random, such correlated data tends to have better results using this method. At the same time, the deviation can be further reduced through multiple iterations. The proper handling of missing values is crucial to the final performance of the model, so it needs to be done carefully.

2.2 Feature Coding

In machine learning, most algorithms can only handle numerical data, not categorical variables directly. Therefore, it is crucial to convert categorical variables into numerical form so that the model can understand and utilize this information. Feature coding is the process of converting classified data into numerical data, making it suitable for machine learning models.

One-Hot Encoding:

Unique thermal coding is a common method that converts each classification value into a binary vector where only one position is 1 and the rest is 0. For example, for the "port of embarkation" feature, the unique thermal code generates a separate binary feature for each port (C, Q, S). This method is simple and intuitive, but when there are many values of categorical variables, it will lead to a sharp increase in the feature dimension, which will increase the calculation cost.

Ordinal Encoding:

Sequential encoding maps each value of a categorical variable to a unique integer. For example, gender could be coded as "male =1, female =0". This method works well for categorical variables that have a natural order, but its use on categorical variables that have no order relationship can introduce faulty assumptions, resulting in degraded model performance.

In this project, we chose to use unique thermal coding to deal with the two categorical variables "gender" and "port of embarkation". The advantage of single-heat coding is that it does not introduce hypothetical relationships between classes, thus avoiding the misdirection that sequential coding can bring. Although unique thermal coding increases the feature dimension, this increase is acceptable given the data size of this project. In addition, unique thermal coding can ensure that each classification value is treated independently and equally in the model, which helps to improve the accuracy and stability of the model [3].

2.3 Feature Engineering

Function engineering is the procedure of transforming raw information into a suitable machine finding out design. The aim is to extract valuable information from raw data to enable versions to find out and make predictions much better. Effective function design can dramatically boost design efficiency, lower training time, and boost model interpretability.Developing brand-new functions is a vital action in attribute design to enhance the anticipating power of a model by introducing new variables. In the Titanic Survival Prediction project, we can create new attributes based on existing ones, such as family dimension (FamilySize = SibSp + Parch + 1). This new function can help the version much better understand the guest's social history, therefore boosting the accuracy of predictions.

Feature choice is another vital action in feature engineering, which intends to decrease the dimensionality of the information collection and get rid of repetitive or irrelevant attributes, therefore improving the efficiency and accuracy of the design. The commonly used attribute choice techniques include filtering system, installing and wrapping. As an example, we can select one of the most beneficial attributes by determining a function's value rating and remove attributes that do not contribute to the model's forecasts. In this task, we eliminated PassengerId, Name, Cabin number, and Ticket number, as these functions are not straight relevant to the passenger's survival likelihood [4]

3. Data exploration and preprocessing

3.1 Data Description

The data sets used for this project were drawn from the Titanic survival prediction contest provided by Kaggle and were divided into training sets and test sets. The training set contains 891 records, while the test set contains 418 records. Each record represents a passenger's information, and features include:

Classification: Sex (Sex), embarkation port (Embarked), cabin class (Pclass).

Numerical characteristics: Age (Age), Fare (Fare), number of siblings/spouses (SibSp), number of parents/children (Parch).

Target variable: Survived, provided only in the training set.

The following diagram shows the characteristics and types of the training set. As you can see, the dataset contains multiple classification and numerical features that will be used for subsequent model training and prediction.

3.2 Data merging and cleaning

In order to deal with the data of the training set and the test set uniformly, we first combine the two data sets. The combined dataset contains a total of 1309 records. After combining the data, we performed the following data cleansing:

Remove irrelevant features:

We removed features not directly relevant to survival predictions, such as PassengerId, Name, Cabin number and Ticket number, to simplify the model. This reduces the complexity of the data and avoids the impact of noise on model training.

Dealing with missing values:

Embarked port: The missing value is embarked on the port ‘C’ with the highest frequency. This processing ensures data integrity and prevents the model from being unable to process due to missing values.

Age and Fare: Interpolates using an Iterative Imputer. The method improves the accuracy of interpolation by predicting the missing value through multiple iterations. Iterative Imputer uses other features to predict missing values to restore the integrity of the data more accurately.

Feature coding:

Sex: Encode ‘male’ as 1 and ‘female’ as 0. This transformation can turn classification features into numerical features and is suitable for most machine learning algorithms.

Embarked: The ‘C’, ‘Q’, and ‘S’ are converted to three binary features (Embarked_C, Embarked_Q, Embarked_S) using One-Hot Encoding. The unique heat coding can avoid the wrong assumptions introduced by sequential coding and make the model deal with classification features more accurately.

4. Model selection and training

4.1 Model selection

For the Titanic survival prediction project, we chose four models :logistic regression, Random Forest, XGBoost, and LightGBM.

Logistic regression is a linear model commonly used in binary classification. Its advantages are simple, quick, and easy to explain and understand. The survival classification of the Titanic is clearly a binary classification, suitable for logistic regression models. Logistic regression can quickly establish a baseline model and provide an interpretation of the feature survival probability [5]. However, logistic regression is only a simple linear model, which may not recognize some complex nonlinear relationships. Other models are therefore needed to be further supplemented.

A random forest is an integrated model based on decision trees by training a large number of decision tree models using random samples and integrating them according to the vote of each decision tree as the final result. Random forest has good anti-overfitting ability because of its random sample characteristics

At the same time, the decision tree class model is very suitable for processing categorical variables [6].

Both XGBoost and LightGBM are gradient elevators, but there are some differences in the specific details. Gradient elevator is to integrate a large number of weak learners to constantly fit close to the target value, and each model learns from the residual of the previous model, which can make the model have a strong generalization ability. XGBoost tends to have a lot of processing power; LightGBM adopts the decision tree algorithm based on histogram, which greatly improves the running speed of the model and can be iterated and optimized quickly [7].

4.2 Model Evaluation

Throughout the model analysis process, we made use of cross-validation to evaluate model efficiency. Specifically, we executed a 5-fold cross-validation for each version, determining its ordinary precision and common discrepancy. Cross-validation is an usual design evaluation approach, which can effectively avoid version overfitting and supply even more secure performance assessment results.

To further enhance the design specifications, we conduct a grid search. Through grid search, we figure out the finest criterion Settings for each and every version and utilize these criteria to re-evaluate design efficiency. The analysis results program that the enhanced XGBoost and LightGBM models execute well in both accuracy and stability.

It can be seen from the assessment results that the arbitrary forest, XGBoost and LightGBM versions all reveal high precision (around 0.83), while the logistic regression design does slightly worse. This reveals that set discovering techniques (such as arbitrary woodlands, XGBoost, and LightGBM) are much more reliable than straight models (such as logistic regression) when managing complex information and function communications. These results better validate our choice in model choice.

5. Model optimization and integration

5.1 Version evaluation index

To evaluate the efficiency of the incorporated version, we utilize the ROC AUC (Receiver Operating Characteristic Area Under Contour) as the key examination metric. ROC AUC is a frequently utilized efficiency analysis index of category versions, which can comprehensively mirror the classification efficiency of designs under various thresholds.

Assessment index

Accuracy: Measures the proportion of appropriately categorized samples in a prediction outcome.

Precision: Procedures the percentage of an example predicted to be favorable that is positive.

Recall: Measures the proportion of an example that is appropriately forecasted to be positive.

F1 Score: A harmonic standard of accuracy and recall, utilized to gauge the overall performance of the design.

ROC AUC: Comprehensively evaluates the category efficiency of the model under different limits. The greater the AUC value, the better the version efficiency.

5.2 Combination Approach

Ensemble Discovering is a method that improves the efficiency of a model by integrating the predictions of numerous base learners. Compared to a solitary version, ensemble discovering can substantially improve the generalization capability and effectiveness of the model. Common ensemble discovering approaches consist of Bagging (e.g., arbitrary forest), Boosting (e.g., XGBoost and LightGBM), and Mixing [8]

In this project, we utilized a hybrid technique of heavy average to integrate the predictions of logistic regression, arbitrary forest, XGBoost, and LightGBM. This method incorporates the benefits of each model by assigning various weights, so as to get more accurate forecast outcomes. By combining the forecast outcomes of numerous models, we get more steady and precise forecast performance. The ROC AUC value of the integrated version is 0.883, showing that the model has high precision and reliability in the survival forecast job.

6. Conclusion

Through the evaluation and forecast of Titanic passenger data, we efficiently used numerous maker finding out models, including logistic regression, Random Forest, XGBoost, and LightGBM. Via feature design, missing value processing and criterion tuning, we construct an efficient prediction version. The final incorporated model dramatically boosted the forecast performance by weighted standard technique, and the ROC AUC worth got to 0.883, ranking 294/15479 in the Kaggle competition, ranking in the leading 2%.

References

[1] Emmanuel, T., Maupong, T., Mpoeleng, D., et al. “A Survey on Missing Data in Machine Learning.” *Journal of Big Data*, vol. 8, no. 140, 2021, https://doi.org/10.1186/s40537-021-00516-9.

[2] Palanivinayagam, A., and R. Damaševičius. “Effective Handling of Missing Values in Datasets for Classification Using Machine Learning Methods.” *Information*, vol. 14, no. 2, 2023, p. 92, https://doi.org/10.3390/info14020092.

[3] Pudjihartono, N., Fadason, T., Kempa-Liehr, A. W., and J. M. O’Sullivan. “A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction.” *Frontiers in Bioinformatics*, vol. 2, 2022, p. 927312, https://doi.org/10.3389/fbinf.2022.927312.

[4] “Discover Feature Engineering: How to Engineer Features and How to Get Good at It.” *Machine Learning Mastery*, https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/. Accessed 26 May 2024.

[5] Couronne, R., Probst, P., and A. L. Boulesteix. “Random Forest versus Logistic Regression: A Large-Scale Benchmark Experiment.” *BMC Bioinformatics*, vol. 19, no. 270, 2018, https://doi.org/10.1186/s12859-018-2264-5.

[6] Shao, Z., Ahmad, M. N., and A. Javed. “Comparison of Random Forest and XGBoost Classifiers Using Integrated Optical and SAR Features for Mapping Urban Impervious Surface.” *Remote Sensing*, vol. 16, no. 4, 2024, p. 665, https://doi.org/10.3390/rs16040665.

[7] Hong, W., Zhou, X., Jin, S., Lu, Y., Pan, J., Lin, Q., Yang, S., et al. “A Comparison of XGBoost, Random Forest, and Nomograph for the Prediction of Disease Severity in Patients With COVID-19 Pneumonia: Implications of Cytokine and Immune Cell Profile.” *Frontiers in Cellular and Infection Microbiology*, vol. 12, 2022, p. 819267, https://doi.org/10.3389/fcimb.2022.819267.

[8] Zhang, Y., Liu, J., and W. Shen. “A Review of Ensemble Learning Algorithms Used in Remote Sensing Applications.” *Applied Sciences*, vol. 12, no. 17, 2022, p. 8654, https://doi.org/10.3390/app12178654.