DATA 572: Supervised Learning
Course Project Report

Group 7 Members: Hwimin Park, Sage Yang, Anita Zeng

January 30, 2026

1 Abstract

This project aims to implement supervised machine learning techniques to predict passenger survival in the Titanic disaster using an extended set of features. Three machine learning models are implemented: Logistic Regression, K Nearest Neighbours, and Random Forest. To obtain efficient and high-performance models with a reduced number of features, three machine learning models are implemented. The data preprocessing step includes handling missing values, converting categorical variables to numerical variables, and normalizing numerical variables. The dataset is split into 75% for training and 25% for testing. Stratified sampling is applied to split the data, and a random number generator seed is set to 42 for reproducibility. K-fold cross-validation is applied to GridSearchCV to tune hyperparameters for each model. To reduce features, feature selection is implemented by considering only the top 7 features by Random Forest importances. To evaluate the models, accuracy is implemented for each model. From the results, it can be determined that title group, fare per person, and age are significant features in predicting passenger survival. The tuned Logistic Regression achieved 82.5% accuracy, KNN 81.8%, and Random Forest 85.4%. From the experiment, it can be determined that hyperparameter tuning and feature reduction are important in building accurate classification models.

2 Introduction

Supervised learning is a key area of machine learning that is utilized for classification purposes. In classification, a machine learning algorithm is trained on a set of data and then utilized for predictions on unseen data. In real-world applications like medical diagnosis or fraud detection, classification algorithms are used for making decisions based on complex data. The survival prediction on the Titanic dataset is a classic classification problem where a binary decision is made based on various passenger-related features like age, class, etc. The Titanic dataset is a classic example of a complex classification problem where classes are imbalanced, there are missing values, and some features are redundant.

The main objective of this project is to develop small yet efficient models using three supervised approaches: Logistic Regression, K Nearest Neighbours, and Random Forests. These three approaches were chosen for their individual strengths, which can be used for solving the problem. Logistic Regression can handle linear relationships. KNN can handle instance-based learning in non-linear space. Random Forests can handle non-linear relationships as well. We focus on efficiency with a high level of accuracy, which can handle a reduced feature space without compromising the efficiency of the model, as per the requirement of developing a model that can handle unseen instances without overfitting. The augmented Titanic problem with new features such as family_size and fare_per_person can offer deeper insights into the problem and can improve the results of the model, which can be achieved with preprocessing and feature selection.

3 Methodology

To address the classification task, we followed a structured approach encompassing data preprocessing, splitting, model building, hyperparameter tuning, and evaluation. All procedures were implemented in Python using scikit-learn, ensuring reproducibility with a fixed random seed of 42.

3.1 Data Preprocessing

The data set consists of 891 data points with 26 features, where Survived is the target variable, taking values 0 for no, 1 for yes. Missing values were present in Age (88 missing values), Cabin (147 missing values), Embarked (3 missing values), and cabin room number (92 missing values). Age and Fare were filled with median values, but since Fare does not have missing values, we only filled Age with median values for robustness against outliers, Embarked with mode 'S', and cabin room number with 0, as it is a placeholder for unknown values. Similarly, missing values in Cabin were filled with 'Unknown' but were dropped later on due to sparsity.

Categorical data: Sex was encoded as binary (male = 0, female = 1). Similarly, Embarked was encoded as categorical ('C', 'Q', 'S'); title group was encoded as categorical ('Mr', 'Mrs', 'Miss', 'Master', 'Other'); cabin deck was encoded as categorical (A, B, C, D, E, F, G, Unknown); is alone was encoded as categorical (0, 1). Similarly, Pclass was encoded as categorical (1, 2, 3). Numerical data: Age, SibSp, Parch, Fare, name length, family size, ticket group size, fare per person, age fare ratio, cabin room number, cabin score, name word count were standardized using StandardScaler, as they are scale-invariant, which is useful for distance-based algorithms like KNN. Irrelevant features: PassengerId is a unique id, Name, Ticket, as they contain redundant information in the form of name length, Ticket, title, as it is already present in title group, booking reference, service id, as they are hashed ids, have low correlation with target, Cabin, as it is already present in cabin deck.

A ColumnTransformer was used to combine the steps of imputation and data transformations, but it was only performed on the training data set to avoid data leakage.

3.2 Data Splitting and Resampling

The data was then split into training data (75% or 668 samples) and test data (25% or 223 samples) using stratified sampling, where the Survived column is taken into account to ensure the data is well-balanced, considering the small imbalance in the data. In order to perform comprehensive hyperparameter tuning, cross-validation was performed on the training data set, using the GridSearchCV function, where the data is split into five folds to prevent overfitting

3.3 Model Building and Training

We built three models:

- **Logistic Regression**: A linear model estimating survival probabilities via the sigmoid function: $P(y = 1|X) = \frac{1}{1+e^{-(\beta_0+\beta^T X)}}$.
- **K Nearest Neighbours (KNN)**: A non-parametric instance-based model that classifies a sample based on the majority vote of its k nearest neighbors in feature space, using Euclidean distance.
- **Random Forests**: An ensemble of decision trees using bagging and feature randomness to reduce variance. Each tree partitions data based on Gini impurity, with aggregation via majority vote.

Models were trained on preprocessed training data. To promote efficiency, feature importance from a baseline Random Forest (top 7 importances selected) was used to reduce features post-initial training, retraining models on the subset.

3.4 Hyperparameter Tuning

Tuning was performed on the training set using GridSearchCV with 5-fold CV and accuracy scoring:

- KNN: n_neighbors [3, 5, 7, 9, 11].
- Random Forests: n_estimators in [50, 100, 200], max_depth in [None, 10, 20].

Best parameters were selected based on cross-validated accuracy and refit to the full training set.

3.5 Result Evaluation and Visualization

Performance evaluation for the test set was done using the following: accuracy (primary metric for classes with an almost equal number of samples) and confusion matrices. Visualizations used were confusion matrices' heatmaps and feature importances' bar plots. If applicable, plots for the loss/accuracy curve vs. epoch would also be used. However, since this is not a neural network-based model, we used the tables for the mentioned metrics.

4 Experiment

4.1 Experimental Design

The experiment aimed to compare the baseline models with their tuned versions using both full and reduced features. The train/test sets for the experiment were the same for both models to ensure an accurate comparison. The baseline models were trained to obtain initial insights into the experiment, which were then used to tune the models for optimization. Feature selection was used after the baseline Random Forest to determine the top features' contribution to the experiment. The tuned models were trained using the reduced set (7 features), which helped to check the efficiency improvements. The experiment's metrics were calculated using the test set to check the generalization ability of the models. The experiment also faced some challenges, such as the overfitting problem with tree-based models.

4.2 Results

Baseline performances on the full feature set:

Model	Accuracy
Logistic Regression	0.825
KNN	0.806
Random Forest	0.851

Tuned performances (full features):

Model	Accuracy	Best Parameters
Logistic Regression	0.825	
KNN	0.818	n_neighbors=7
Random Forest	0.854	max_depth=10, n_estimators=100

Feature importances from baseline Random Forest identified top 7: age, sex, fare_per_person, parch, ticket_group_size, pclass, sibsp.

On reduced features, the tuned Random Forest maintained 0.816 accuracy, with confusion matrix:

$$\begin{bmatrix} 120 & 19 \\ 22 & 60 \end{bmatrix}$$

(True Negatives: 120, False Positives: 199, False Negatives: 22, True Positives: 60).

4.3 Analysis and Discussion

Random Forest performed best out of all, with an accuracy of (0.854), which is expected due to the reduction in variance and non-linear interactions, especially when features are correlated, as in the case of Parch and ticket group size. Logistic Regression performed well with an accuracy of (0.825) due to linear separation in sex/class/fare space. KNN performed poorly with an accuracy of (0.818) due to sensitivity to scaling, noise, and the curse of dimensionality, despite standardization.

The features used were significant in this case. Sex is an important feature as it accounts for gender and marital status biases in survival (survival of women and children first), fare per person and Fare account for wealth status (high classes had higher survival rates), and age accounts for age biases in survival (prioritization based on age). The features used were reduced to 7 out of approximately 30 features without compromising performance, thus improving efficiency (faster processing time and lower complexity) without compromising performance, as the features removed (cabin score, age fare ratio) had low importance values

(<0.02).

The challenges faced in this case were multicollinearity in features (PClass and fare per person have correlation coefficient -0.8), which was overcome by using feature selection methods to select features with high importance values. The missing values in the age column were biased towards median, which was overcome by median imputation. The class imbalance in this case resulted in higher recall for non-survivors. Overfitting was prevented using various methods like CV and tuning, which resulted in test set accuracy close to CV set means.

5 Conclusion

This project successfully applied Logistic Regression, KNN, and Random Forests to predict Titanic survival, achieving up to 81.6% accuracy with minimal features. Key insights include the dominance of socio-economic and demographic factors in predictions, the benefits of tuning models, and feature selection for efficiency.

6 References

- Data Science Dojo. (n.d.). Titanic Dataset.
  Retrieved from https://github.com/datasciencedojo/datasets/blob/master/titanic.csv.

- Scikit-learn Documentation. (2023). User Guide. https://scikit-learn.org/stable/user_guide.html.