Titanic Survival Prediction

Machine Learning Classification Report

Marek Homiak

February 2026

1 Introduction

The Titanic dataset is one of the most well-known benchmarks in machine learning. It contains information about passengers aboard the RMS Titanic, including demographic attributes, ticket class, fare, and whether the passenger survived. The goal of this project is to build a binary classification model that predicts passenger survival (`Survived = 1`) or death (`Survived = 0`) based on the available features.

The analysis covers the full machine learning pipeline: data inspection, exploratory data analysis, feature engineering, preprocessing, model training, and evaluation. Three models are trained and compared — Logistic Regression (baseline), Random Forest, and Gradient Boosting — with particular attention paid to avoiding data leakage and multicollinearity.

2 Data Overview

The dataset contains **891 entries** and **11 columns** after loading with `PassengerId` as the index. The features are:

- `Survived`: Target variable (0 = did not survive, 1 = survived)
- `Pclass`: Ticket class (1st, 2nd, 3rd)
- `Name`: Passenger name (string)
- `Sex`: Passenger gender
- `Age`: Age in years (float, contains missing values)
- `SibSp`: Number of siblings/spouses aboard
- `Parch`: Number of parents/children aboard
- `Ticket`: Ticket number (string)
- `Fare`: Passenger fare (float)
- `Cabin`: Cabin number (string, heavily missing)
- `Embarked`: Port of embarkation (C = Cherbourg, Q = Queenstown, S = Southampton)

Statistical summary of the numerical features reveals a mean age of approximately 29.7 years (with a median around 28), a mean fare of $32.20 (heavily right-skewed), and a survival rate of approximately 38.4%.

2.1 Missing Values

Initial inspection reveals that three columns contain missing values:

Column	**Missing Count**	**Missing %**
Age	177	19.87%
Cabin	687	77.10%
Embarked	2	0.22%

Given that the `Cabin` column is missing in over 77% of entries, it provides very limited analytical utility and was dropped. The `Ticket` column was also dropped as it is a high-cardinality identifier with no direct predictive value.

3 Exploratory Data Analysis

Various univariate and bivariate analyses were performed to understand distributions and relationships with the survival outcome.

3.1 Survival Distribution

Of the 891 passengers, **342 (38.38%) survived** and **549 (61.62%) did not survive**. This mild class imbalance is important to acknowledge when interpreting accuracy metrics, as a naive majority-class predictor would already achieve around 61% accuracy.

Breaking survival down by gender reveals a striking pattern: female passengers had a substantially higher survival rate than male passengers, consistent with the historical "women and children first" evacuation policy. Similarly, first-class passengers show much higher survival rates than those in second or third class.

3.2 Passenger Class Distribution

The majority of passengers travelled in third class, with the three classes distributed roughly as: Class 1 $\approx$ 24%, Class 2 $\approx$ 21%, Class 3 $\approx$ 55%. Higher ticket class is positively correlated with survival probability.

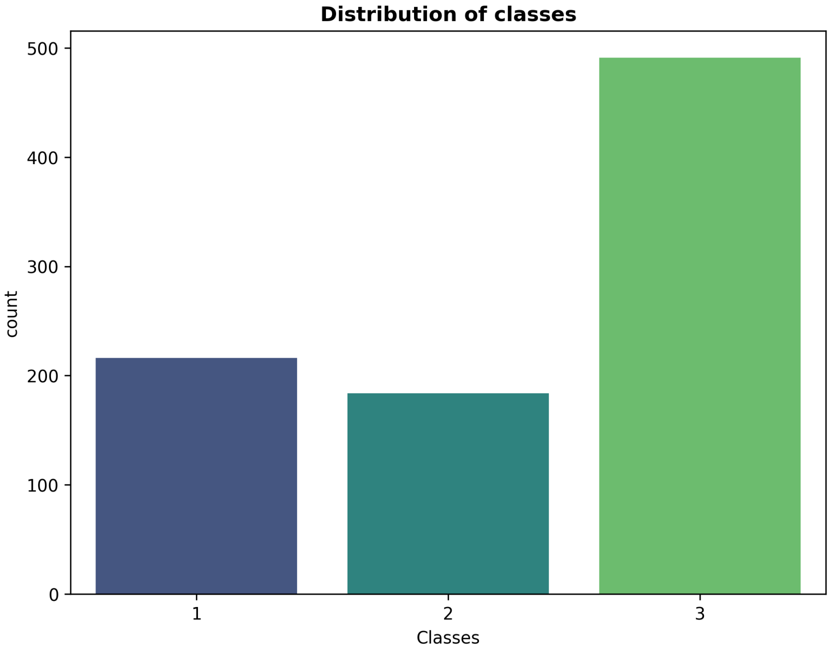


Figure 1: Distribution of classes.

3.3 Age and Fare Distribution

The age distribution is approximately bell-shaped with a slight right skew, and a median of around 28 years. Very young children (ages 0–5) show notably higher survival rates. The `Fare` feature exhibits extreme right skew, with a small number of very expensive first-class tickets inflating the mean. A log transformation ($\log(1 + \text{Fare})$) was applied during preprocessing to reduce this skewness.

Boxplot analysis shows that survivors tended to pay higher fares on average, reflecting the class-survival link. Age distributions between survivors and non-survivors are similar, though slightly younger passengers show a marginally higher survival probability.

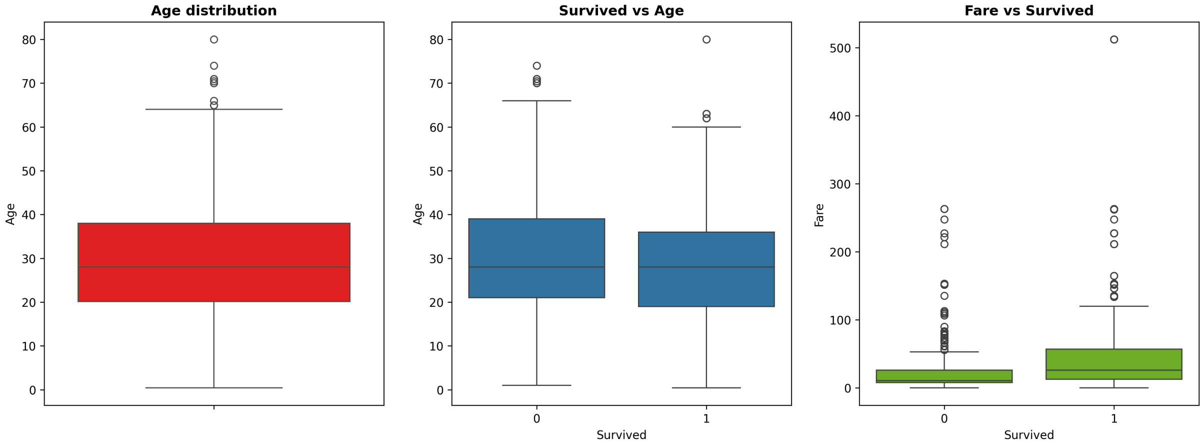


Figure 2: Distribution of Age, Fare and Survived vs Fare.

3.4 Title Extraction from Name

Passenger titles were extracted from the `Name` column using a regular expression and analysed for their survival rates:

Title (selected)	Count	Survival Rate (%)
Master	40	57.50
Miss	182	69.78
Mr	517	15.67
Mrs	125	79.20
Dr	7	42.86
Col / Major	4	50.00
Lady / Countess	2	100.00

The title is a strong proxy for both gender and social status, making it a valuable predictive feature. Titles were subsequently grouped into four categories for modelling.

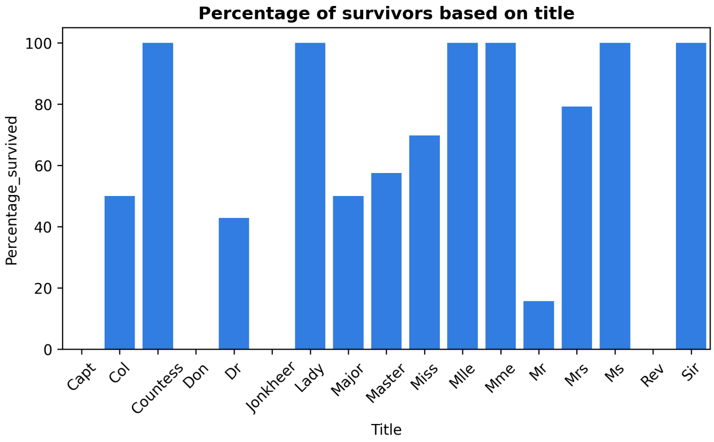


Figure 3: Titles survival.

3.5 Family Size

A `FamilySize` feature was engineered as SibSp + Parch + 1. Analysis of survival rate against family size shows that solo travellers had a lower-than-average survival probability, while medium-sized families (2–4 members) had the highest survival rates. Very large families ($\geq$ 6) had near-zero survival. The constituent components `SibSp` and `Parch` were retained in the final feature set while the `FamilySize` column itself was dropped.

4 Data Preprocessing

Preprocessing was performed carefully to prevent data leakage by fitting all transformers on the training set only and applying them to the test set.

4.1 Train–Test Split

The dataset was split 80/20 into training and test sets using stratified sampling to preserve the class distribution:

- Training set: 712 samples
- Test set: 179 samples

4.2 Missing Value Imputation

- `Age`: Imputed with the **mean** of the training set using `SimpleImputer(strategy='mean')`. Fitted on training data only to avoid data leakage.
- `Embarked`: Imputed with the **most frequent** value from the training set (only 2 missing values).

4.3 Feature Engineering: Title Grouping

Titles extracted from the `Name` column were grouped into four categories:

- `personal_t`: Mr, Mrs, Miss, Master, Mme, Ms, Mlle
- `professional_t`: Dr, Rev, Col, Major, Capt
- `nobel_t`: Sir, Lady, Don, Countess, Jonkheer
- `other_t`: all remaining titles

The raw `Name` and `LastName` columns were dropped after extraction.

4.4 Encoding Categorical Variables

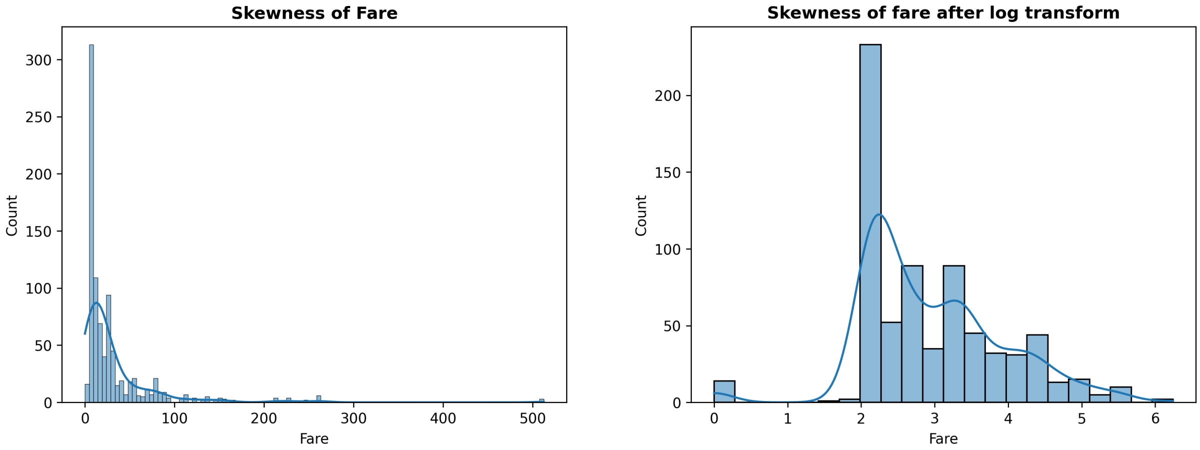
One-Hot Encoding was applied to `Sex`, `Embarked`, and `Title_group` using `OneHotEncoder(sparse_outpu`
producing 8 binary columns:

`Sex_female`, `Sex_male`, `Embarked_C`, `Embarked_Q`, `Embarked_S`, `Title_group_nobel_t`,
`Title_group_personal_t`, `Title_group_professional_t`

To avoid the **dummy variable trap**, one reference category per group was dropped: `Sex_male`, `Embarked_S`, and `Title_group_personal_t`.

4.5 Log Transformation of Fare

Due to extreme right skew, a $\log(1 + x)$ transformation was applied to `Fare` using `numpy.log1p`, substantially normalising the distribution.



(a) Before

(b) After

Figure 4: Fare before and after log transform

4.6 Standard Scaling

`StandardScaler` was applied to `Age` and `Fare` (post log-transform), fitted exclusively on training data.

4.7 Final Feature Set

After all preprocessing steps, the final training matrix contains **10 features**:

Feature	Type
`Pclass`	Ordinal integer
`Age`	Scaled float
`SibSp`	Integer
`Parch`	Integer
`Fare`	Log-transformed, scaled float
`Sex_female`	Binary (OHE)
`Embarked_C`	Binary (OHE)
`Embarked_Q`	Binary (OHE)
`Title_group_nobel_t`	Binary (OHE)
`Title_group_professional_t`	Binary (OHE)

5 Model Training and Evaluation

Three models were trained and evaluated on the held-out test set. Metrics reported are accuracy, ROC-AUC, F1-score, and the confusion matrix.

5.1 Logistic Regression (Baseline)

Logistic Regression with `max_iter=1000` and default L2 regularisation was used as the baseline. The fitted coefficients are:

Feature	Coefficient
`Pclass`	-0.842
`Age`	-0.517
`SibSp`	-0.350
`Parch`	-0.169
`Fare`	$+0.430$
`Sex_female`	$+2.531$
`Embarked_C`	$+0.269$
`Embarked_Q`	$+0.613$
`Title_group_nobel_t`	$+0.118$
`Title_group_professional_t`	$+0.259$

Being female is the strongest positive predictor of survival, while higher `Pclass` number (lower social class) and older age are the strongest negative predictors, consistent with historical accounts.

Test Set Results

Metric	Score
Accuracy	81.01%
ROC-AUC	83.89%
F1-score	73.85%

Confusion Matrix:

$$\begin{pmatrix} 97 & 13 \\ 21 & 48 \end{pmatrix}$$

(rows: actual, columns: predicted; 0 = did not survive, 1 = survived)

Cross-Validation across Regularisation Strengths

5-fold cross-validation was performed on the training set for $C \in \{0.1,\ 0.5,\ 1,\ 10,\ 15\}$:

C	Accuracy (CV)	ROC-AUC (CV)	F1 (CV)
0.1	79.64%	85.99%	71.53%
0.5	78.52%	86.15%	71.26%
1	78.52%	86.07%	71.26%
10	78.66%	85.76%	71.40%
15	78.52%	85.75%	71.26%

ROC-AUC peaks at $C = 0.5$, suggesting stronger regularisation generalises slightly better on this dataset despite a marginal accuracy drop.

5.2 Dummy Variable Trap Experiment

An identical Logistic Regression was trained retaining all one-hot encoded columns (without dropping reference categories) to empirically verify the dummy variable trap. Results:

Metric	Without trap	With trap
Accuracy	81.01%	79.89%
ROC-AUC	83.89%	83.72%
F1-score	73.85%	72.31%

Takeaways:

- L2 regularisation (sklearn default) can handle linearly dependent features, so the performance gap is small.
- The dummy variable trap is largely a theoretical concern when regularisation is applied, but dropping reference categories is still best practice and produces a marginally better model.
- For linear regression without regularisation the issue would be more significant.

5.3 Random Forest

A Random Forest with `n_estimators=300`, `max_depth=5`, and `random_state=42` was trained on the same feature set.

Metric	Score
Accuracy	81%
ROC-AUC	84%
F1-score	71%

Confusion Matrix:

$$\begin{pmatrix} 104 & 6 \\ 28 & 41 \end{pmatrix}$$

Compared to Logistic Regression, the Random Forest produces far fewer false positives (6 vs. 13) but more false negatives (28 vs. 21). It is more conservative — it is less likely to incorrectly predict survival, but misses more actual survivors as a result. Overall ROC-AUC matches Logistic Regression at 0.84.

5.4 Gradient Boosting

A Gradient Boosting classifier with `n_estimators=100`, `learning_rate=0.1`, `max_depth=3`, and `random_state=42` was trained on the same feature set.

Metric	Score
Accuracy	81%
ROC-AUC	81%
F1-score	72%

Confusion Matrix:

$$\begin{pmatrix} 101 & 9 \\ 25 & 44 \end{pmatrix}$$

Gradient Boosting achieves the same accuracy as the other two models but a lower ROC-AUC (0.81 vs. 0.84), suggesting the default hyperparameters are not optimal for this dataset. The confusion matrix sits between LR and RF in terms of the false positive / false negative trade-off. Hyperparameter tuning of `learning_rate` and `n_estimators` would likely improve performance.

6 Model Comparison

Model	**Accuracy**	**ROC-AUC**	**F1-score**
Logistic Regression	81.01%	83.89%	73.85%
Random Forest	81%	84%	71%
Gradient Boosting	81%	81%	72%

All three models converge on 81% accuracy, indicating they are making errors on the same difficult cases. Logistic Regression and Random Forest tie on ROC-AUC (0.84), making them the best-performing models with default settings. Gradient Boosting underperforms slightly on ROC-AUC with its current hyperparameters but offers the most room for improvement through tuning.

7 Conclusion

This analysis demonstrates a complete supervised machine learning workflow applied to the Titanic survival prediction task:

1. **Data Quality**: The dataset required targeted cleaning — dropping `Cabin` (77.1% missing), imputing `Age` and `Embarked`, and removing non-informative identifiers (`Ticket`, `Name`).

2. **Feature Engineering**: Extracting and grouping passenger titles from `Name` provided a strong categorical signal capturing both gender and social status in a single feature.

3. **Preprocessing Rigour**: All transformations — imputation, encoding, log-transform, scaling — were fitted exclusively on the training split to prevent data leakage.

4. **Model Performance**: All three models achieved 81% accuracy on the held-out test set. Logistic Regression and Random Forest tied on ROC-AUC at 0.84. The dominant survival predictors align with historical records: female gender, higher ticket class, and passenger title.

5. **Experimental Insights**: The dummy variable trap experiment confirmed that L2 regularisation mitigates multicollinearity in practice, though dropping reference categories remains best practice.

6. **Next Steps**: Gradient Boosting hyperparameter tuning, age imputation by title group, and a binned `FamilySize` feature are the most promising avenues for further improvement.