MLOps Assignment 1

2276046 Minseo Kim

1. Problem Definition

The Titanic dataset, while widely used in AI education and competitive machine learning, holds untapped potential for real-world applications. This project seeks to fill a significant gap by creating a predictive model to evaluate mortality risk in maritime disasters, directly applicable to insurance underwriting. The model measures individual fatality probabilities based on passenger characteristics, facilitating the development of data-driven insurance products for maritime incidents.

The code is located here: https://github.com/440g/kaggle-struggle/tree/main/0.titanic.

2. Dataset Description

Given the limited timeframe for this project, I used MLJAR, an AutoML package that I learned about in class, to preprocess the data and select a model. Since the Titanic dataset is already well-characterized, I used AutoML to validate it as a baseline model.

AutoML results showed that the ensemble model performed the best for the logloss metric. On the other hand, the neural network model was significantly less accurate than the baseline model. The high performance achieved by Xgboost, Random Forest, and ensemble models suggests that this may be due to the nature of the dataset.[1]

In other words, the dataset's characteristics—its small size, structured structure, many nonlinear interactions, and categorical variables—meant that it performed well with tree-based models and poorly with neural network configurations.

2.1 Data Source

Utilized the canonical Titanic dataset from Kaggle comprises passenger records with demographic, travel-class, and survival information.[2]

[1] https://github.com/440g/kaggle-struggle/tree/main/0.titanic/AutoML_1
[2] https://www.kaggle.com/competitions/titanic/leaderboard

2.2 Strategic Preprocessing Pipeline

Missing Value Handling:

- Age: Imputed using salutation-derived social status indicators (Mr./Mrs./Master, etc.) extracted from the Name field
- Cabin: Transformed into a binary 'Has_Cabin' feature, treating missingness as potentially meaningful (indicating lower-priority accommodations)

Feature Engineering:

- Applied a logarithmic transformation to the Fare to address the right-skewed distribution and extreme outliers
- Derived 'IsAlone' from family size indicators (SibSp/Parch)
- Converted categorical variables (Sex, Pclass, Embarked) to model-ready formats

Feature Selection:

- Eliminated low-predictive features (Ticket, PassengerId)

Exploratory analysis revealed key predictive features: Fare, Age, Title (name-derived), Sex, and Pclass.

3. Model Description, Evaluation Methods, and Metrics

The CatBoostClassifier was selected through rigorous evaluation:

Selection Criteria:

1. Performance Validation: Tree-based models (XGBoost, Random Forest) demonstrated superior performance in AutoML benchmarks
2. Data Compatibility: Native handling of categorical features (7/10 features categorical) without need for one-hot encoding
3. Robustness: Built-in regularization and ordered boosting mitigate overfitting risks with limited samples (n=891)
4. Operational Efficiency: Achieved 1,000-iteration training in 7.4 seconds with default parameters

Additionally,

- Class weight inversion to prioritize death prediction (class 0) accuracy
- Stratified 10-fold cross-validation maintaining class balance (62% deceased, 38% survived)

4. Results & Interpretation

4.1 Aggregate Metrics

Accuracy	Precision	Recall	F1-Score	ROC AUC
0.8492822966507	0.7986577181208	0.7828947368421	0.7906976744186	0.8350563909774

Overall, we can see that the classification performance is good, with a good precision/recall balance.

4.2 Class-Specific Performance

However, since the goal I was trying to solve is to better classify dead people, it is necessary to check the classification for each class. The results and confusion matrix by class are shown below.

	Precision	Recall	F1-Score
Dead(0)	0.88	0.89	0.88
Survived(1)	0.80	0.78	0.79

	Predicted 0	Predicted 1
TN(0)	236	30
TF(1)	33	119

4.3 Interpretation

The model demonstrates strong discriminative capability, particularly for fatality prediction (89% recall). The balanced precision-recall tradeoff (F1=0.88) confirms operational viability for insurance risk assessment. While survival prediction shows marginally lower performance, this aligns with the business objective of prioritizing accurate mortality risk quantification.

5. Development Environment

- Platform: macOS 15.4.1
- Dependencies: Full environment specification in requirements.txt.[3]

6. Scenarios for Practical Use

This solution successfully transforms historical disaster data into an actionable risk prediction tool. The 89% death recall rate provides insurers with statistically robust mortality estimates, enabling:

- Data-driven premium calculation for maritime policies
- Survivor benefit package design
- Disaster preparedness analytics

Future enhancements could incorporate additional maritime disaster datasets and socioeconomic contextual features to improve generalizability beyond the Titanic's specific demographic profile.

[3] https://github.com/440g/kaggle-struggle/blob/main/0.titanic/requirements.txt