Laboratory 4

Variant 4

Class Group 105

Group 24

By Dimitrios Gkolias and Vasileios Ntalas

Introduction

The task of this lab project is to solve the binary classification problem which is to predict if the person survived the Titanic crash using the Titanic dataset from the Kaggle website. For this task we chose to train a Logistic Regression model and a Random Forest model. Having prepared appropriately the data and fine tuned the parameters of each model we will test the accuracy of these models for the specific problem.

Implementation

Data Preparation:

For the data preparation of our models we proceeded with these actions:

- We removed redundant features: *embark_town, alive, who, class, and adult_male,* since we have similar features for each of these. Namely for *embark_town* we are sufficient with *embarked*, for *alive* we have *survived*, for *who* or *adult_male* we have *sex* and *age*, for *class* we have *pclass* (class is duplicate of pclass, just in text form), and finally *alone* is covered by *sibsp* and *parch*.
- We removed *deck* feature because it had over 50% missing values, and specifically 688 values out of 891 were missing.
- We filled the missing values for *age* feature, specifically for the 177 values missing we filled them with the median age. This is a standard way to handle missing numeric values. Also, for the 2 missing values of the *embarked* feature we filled them with the most frequent port (‘S’ for Southampton).
- We one-hot-encoded the features *sex* and *embarked,* because the models can’t match text categories, so we transform them to numerical values. Moreover, for the *embarked* feature we drop the first category, as because we only need 2 columns to represent 3 categories, for example if we know that a row is not neither of the two categories we can understand that it is the remaining third column. Actually having all 3 columns introduces multicollinearit**y**- one column is always a linear combination of the others and that can confuse our Logistic Regression model, even though we wouldn’t have a problem with our Random Forest model.
- We normalize the numeric features with the StandardScaler of sklearn library, because the values must be in the same scale, because otherwise the model could get biased.

Data split and choice of Models:

We then proceed to the data separation with a random seed to train and test sets, with test set being 20% of the whole dataset, while the train is the other 80%.

We chose the Logistic Regression model, because:

- ◆ It's the go-to baseline for binary classification.
- ◆ Outputs a probability (e.g., “this person has a 75% chance of surviving”).
- ◆ It’s simple, fast, and very interpretable (we can see which features help survival).
- ◆ Works best when the relationship between features and target is linear.

We chose it as our first model to establish a solid baseline and see how a simple model performs.

We also chose the Random Forest model, because:

- 🌲 It is an ensemble of decision trees.
- 🌲 Can handle complex, non-linear relationships between features.
- 🌲 Very powerful and usually performs better than simpler models on real world data.
- 🌲 It’s also robust to noise, missing data, and outliers.

We chose it to compare a non-linear, more flexible model against logistic regression.

Model Training:

We chose accuracy as our evaluation metric because the Titanic dataset has a fairly balanced distribution of the target classes, and accuracy provides a simple and intuitive way to compare model performance. For a classification task like this, it allows us to quickly assess how well our model predicts survival without requiring more complex evaluation metrics.

Moreover, for both of our models we performed 4-fold cross validation and the results we got are the following:

Logistic Regression

C	CV Accuracy
0.01	0.7233
0.1	0.8020
1.0	0.7935
10.0	0.7893

Best Logistic Regression Params: **C = 0.1**, and Best CV Accuracy: **0.8020**.

Random Forest

n_estimators	max_depth	min_samples_split	CV Accuracy
50	None	2	0.7865

50	None	5	0.8062
50	5	2	0.8146
50	5	5	0.8118
50	10	2	0.8090
50	10	5	0.8146
100	None	2	0.7893
100	None	5	0.8048
100	5	2	0.8118
100	5	5	0.8160
100	10	2	0.8146
100	10	5	**0.8188**
200	None	2	0.7921
200	None	5	0.8048
200	5	2	0.8146
200	5	5	0.8160
200	10	2	0.8160
200	10	5	**0.8188**

Best Random Forest Params: **n_estimators = 100, max_depth = 10, min_samples_split = 5**, and

Best CV Accuracy: **0.8188**.

*With 200 estimators we also get the same best accuracy, but we choose to keep 100 estimators as our model parameter.

Discussion

Their performance was assessed using 4-fold cross-validation accuracy, training accuracy, and test accuracy:

Logistic Regression:

- CV Accuracy: **0.8020 ± 0.0287**.
- Training Accuracy: **0.8075842696629213**.
- Test Accuracy: **0.8100558659217877**.

Random Forest:

- Random Forest CV Accuracy: **0.8188 ± 0.0262**.
- Training Accuracy: **0.9115168539325843**.
- Test Accuracy: **0.8659217877094972**.

Logistic Regression achieved a cross-validation accuracy of **0.8020 ± 0.0287**, a training accuracy of **0.808**, and a test accuracy of **0.8101**. These values are consistent across metrics, indicating that the model generalizes well and does not suffer from overfitting. As a linear model, Logistic Regression is simple and interpretable, and its performance demonstrates that even basic models can be effective on this dataset.

Random Forest, on the other hand, achieved a higher cross-validation accuracy of **0.8188 ± 0.0262**, with a training accuracy of **0.9115** and a test accuracy of **0.8659**. The significantly higher training accuracy suggests some level of overfitting, which is typical for tree-based ensemble models. However, the Random Forest still outperforms Logistic Regression on the test set, indicating that it captures more complex relationships in the data and provides better overall predictive performance.

Conclusion

In this lab, we approached the binary classification task of predicting Titanic passenger survival by applying and evaluating two machine learning models: Logistic Regression and Random Forest. Through thoughtful data preprocessing, including handling missing values, removing redundant features, and encoding categorical data, we ensured our dataset was clean and suitable for training.

Logistic Regression served as a strong and interpretable baseline model. It showed consistent performance across cross-validation, training, and test sets, indicating good generalization with no signs of overfitting. On the other hand, Random Forest, a more complex ensemble model, was able to capture non-linear relationships in the data and achieved higher accuracy on both the test set and during cross-validation.

Hyperparameter tuning further improved both models, with the Random Forest reaching a best CV accuracy of **0.8188** and test accuracy of **0.8659**, compared to Logistic Regression's **0.8020** CV and **0.8101** test accuracy. While Random Forest exhibited slight overfitting, its higher performance makes it the more effective model for this task.

Overall, this project demonstrates how model selection, preprocessing, and proper evaluation can significantly impact classification performance. For production scenarios where accuracy is critical, the Random Forest model is the preferred choice. However, for scenarios requiring simplicity and interpretability, Logistic Regression remains a reliable alternative.