WEEK 8 LABS: Titanic - Machine Learning from Disaster

DUE DATE: Wednesday, 1st January 2025.

Objective

The objective of this lab is to apply machine learning techniques to the Titanic dataset to predict passenger survival. This will help students strengthen their understanding of the machine learning pipeline, from data preprocessing to model evaluation, while developing critical thinking and problem-solving skills.

Introduction

The sinking of the Titanic is one of the most infamous shipwrecks in history.

On April 15, 1912, during her maiden voyage, the widely considered “unsinkable” RMS Titanic sank after colliding with an iceberg. Unfortunately, there weren’t enough lifeboats for everyone onboard, resulting in the death of 1502 out of 2224 passengers and crew.

While there was some element of luck involved in surviving, it seems some groups of people were more likely to survive than others.

In this Lab, we ask you to build a predictive model that answers the question: “what sorts of people were more likely to survive?” using passenger data (ie name, age, gender, socio-economic class, etc.).

Data Description

The data has been split into two groups:

- training set (train.csv)
- test set (test.csv)

The training set should be used to build your machine learning models. For the training set, we provide the outcome (also known as the “ground truth”) for each passenger. Your model will be based on “features” like passengers’ gender and class. You can also use feature engineering to create new features. Is it this training set that the student can split further into another training set and validation set for tasks like cross-validation to improve their model.

The test set should be used to see how well your model performs on unseen data. For the test set, we do not provide the ground truth for each passenger. It is your job to predict these outcomes using your best model.

For each passenger in the test set, use the best model you trained to predict whether they survived the sinking of the Titanic.

We also include *sample_submission.csv*, a set of predictions that assume all and only female passengers survive, as an example of what the final submission file should look like.

Features Overview

The table gives an overview of the features in the dataset.

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1=1st, 2=2nd, 3=3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings/ spouses aboard the Titanic	
parch	# pf parents / children aboard the Titanic	
ticket	Ticket number	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Tasks

Task 1: Data Exploration and Visualization

Steps:

1. Load the Titanic dataset.
2. Analyze key statistics for each feature (mean, median, standard deviation, etc.).
3. Visualize relationships between features (e.g., survival rates by gender, class, and age group).

Deliverable: To be include in a Jupyter Notebook showing all the result for the tasks above with explaining for why you choose certain techniques.

Task 2: Data Cleaning and Preprocessing

Steps:

1. Handle missing values (e.g., for age and embarked columns).
2. Encode categorical variables (e.g., convert gender to numeric values).
3. Normalize/scale numerical features.
4. Split the dataset into training and validation sets.

Deliverable: To be include in a Jupyter Notebook showing all the result for the tasks above with explaining for why you choose certain techniques.

Task 3: Feature Engineering

Steps:

1. Generate new features.
2. Perform feature selection using correlation analysis or importance scores or any other analysis to select the most relevant variables.

Deliverable: A description of new features created and justification for their inclusion or exclusion should be included in the final report. The Jupyter Notebook should also include the result for your feature engineering.

Task 4: Model Selection and Training

Steps:

1. Train at least three different models (e.g., Logistic Regression, Random Forest, Support Vector Machines).
2. Use cross-validation to evaluate model performance.
3. Compare models using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.

Deliverable: Provide a table in the report comparing the performance of different models on the validation set. Consider at least three metrics.

Task 5: Model Optimization

Steps:

1. Perform hyperparameter tuning using *GridSearchCV* or *RandomizedSearchCV* on all the models you selected.
2. Evaluate the optimized models on the validation dataset.

Deliverable: To be include in a Jupyter Notebook showing all the result for the tasks above. A table to be added to the report comparing the performance of the different optimized. Consider at least three metrics.

Task 6: Testing and Submission

Steps:

1. Use your best model to make a prediction on the test dataset (test.csv) and submit the result as a csv file like the *"sample_submission.csv"*.
2. Name your submission file as "{first_name}_submission.csv".
   E.g.:*"Solomon_submission.csv"*

Deliverable: A submission file in a csv format.

Additional Guidelines

- Use tools like Python (Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn) or any preferred machine learning framework.
- Submit all deliverables by the stated deadline.

Submission Requirements and Format

Each student should submit:

1. A single executable Jupyter notebook* that addresses all the tasks and computations
2. One PDF document with all sections that need to be included in the Report.
3. A csv file containing the predictions on the testing data (e.g.: Solomon_submission.csv)

GOOD LUCK!!!!