Surviving the Titanic: Using Machine Learning Classification for Prediction

Abstract:

On April 15, 1912,the Titanic, the worlds largest passenger liner Titanic sank into the North Atlantic Ocean about 400 miles south of Newfoundland, Canada killing more than 1500 people. Using machine learning methods and algorithms, this project will be using the Titanic Survivors dataset from Kaggle.com to predict the survivability of passengers based on features such as age, sex, embarkment etc.
Although the methods and techniques used in the project may be used for risk assessment for insurance pricing for cruise ships or eventually space tourism, the main question is: Would I survive the Titanic?

Goal:

The purpose of this project was to create a predictive classification model to identify who would have survived the sinking of the Titanic. This analysis seeks to establish a meaningfully predictive classification model for predicting survivors. I am hoping to build a model and obtain acceptable accuracy, precision, recall and F1 scores.

Design:

To create a classification model to be used for prediction requires identifying which features would be most useful.
The passenger data was a mix of quantitative and qualitative features. I did some data transformations and preprosessed categorical data to numerical data using sklern's LabelEncoder.

Once preliminary models were built, functions were used to fined the best scoring hyperparameters. Cross-validation of the model followed to check the validity of the model.

Once a model was selected, the model was run against validataion data and once again scored. With an acceptable score, the model was then used on test data and ultimately used to predict whether or not a person, given specific features would survive the Titanic.

The Data

The Titanic Survivor dateset was downloaded from Kaggle.com and had these features:

- PassengerId: A unique index for passenger rows. It starts from 1 for first row and increments by 1 for every new row.
- Survived: refers to whether passenger survived. 1 = survived 0 = not survived.
- Pclass: Ticket class. 1 = First class ticket. 2 = Second class ticket. 3 = Third class
- Name: Passenger's name. Name also contain title. "Mr" for man. "Mrs" for woman. "Miss" for girl. "Master" for boy.

- Sex: Passenger's gender. Male or Female.
- Age: Passenger's age.
- SibSp: Number of siblings or spouses travelling with each passenger.
- Parch: Number of parents of children travelling with each passenger.
- Ticket: Ticket number - Fare: price paid by passenger
- Cabin: Cabin number of the passenger.
- Embarked: refers to where the passenger boarded: Cherbourg (C) --> Southhanpton (S) --> Queenstown (Q)

EDA

- PassengerId, Name, Ticket and Cabin were dropped
- Age was simplified into 4 groupings then numerically encoded
- Embarked was numerically encoded
- Missing values for Age were imputed using mean of Age
- Missing values for Embarked were imputed using mode of Embarked

Algorithms:

- kNN
    - Used the default k value of 5 for base model with k-fold=5
        - Base score: 86%
        - Accuracy: 0.8015517545406453
        - Precision: 0.7378900503290746
        - Recall: 0.6963562753036437
        - F1: 0.7131776195067334
    - Looped through k=3 to 15 to find optimal hyperparameter value
    - Final scores using optimal k=3
        - Final Score: 88%
        - Accuracy after 5 Folds: 0.7657142857142858
        - Precision after 5 Folds: 0.7143853222800591
        - Recall after 5 Folds: 0.7741666666666667
        - F1 after 5 Folds: 0.7382787597797739
- AdaBoost
    - Used the default k value of 5 for base model with n_estimators =50, learning_rate=1 and max_depth=1
        - Base score: 83%
        - Accuracy after 5 Folds: 0.7939340504320226
        - Precision after 5 Folds: 0.7247869674185463
        - Recall after 5 Folds: 0.6855600539811066
        - F1 after 5 Folds: 0.7030123290057977
    - Final result with tuned hyperparameters
        - Final Score: 94%
        - Accuracy after 5 Folds: 0.770952380952381
        - Precision after 5 Folds: 0.7187971342383107
        - Recall after 5 Folds: 0.7616666666666666
        - F1 after 5 Folds: 0.7374441418796257

- RandomForest
- Naïve Bayes
- AdaBoost
- XGB
- SVC()

Tools:

- Pandas, Numpy, counter and random
- Itertools
- Scikit, sklearn
- Seaborn, Matplotlib, Altair for plotting