Titanic: Spark and Machine Learning from disaster

GIOVANNI MANFREDI SEBASTIANO MENEGHIN
gioman | meneghin @kth.se

21st October 2023

1 Problem

Last summer the tragic implosion of the Titan submarine operated by the company OceanGate, revamped the attention over disaster happening also to billionaires. Even if very rare, these events can happen and predicting them would be beneficial to many human lives.

The goal of this project is to process and query data regarding the historical Titanic wreckage [1] and implement a predictive analysis to classify which of the passenger on board will survive the tragedy, based on various parameters (name, age, gender, social status, etc.). The structure in Spark for pre-processing and querying for the model training and the evaluation of the methods based on their accuracy are the main outputs of this project.

2 Tools

The project makes use of:

- Databricks as cloud data platform for hosting the Spark Cluster.
- Scala as a programming language.
- Spark and Spark SQL to pre-process and query the data.
- Spark ML for the prediction model's implementation.
- java.util library to access regex's operations and methods.

3 Data

The data used in this project are gathered from the Titanic - Machine Learning from Disaster [1] online competition. Among the various file present there, only the file *train.csv* is used in the project.

4 Methodology and algorithm

The first step of the project is finding and accessing the data from the Kaggle website. After the download of the data from Kaggle's competiton, they are uploaded and transformed into a table on Databricks' platform using Spark.

The second step is to analyse the data by looking for patterns, feature correlations and missing values. Then the heterogeneity of the features is considered. During this analysis, the data is visualised using printed SQL queries that help understand the logic and motivations behind the choices taken in this phase.

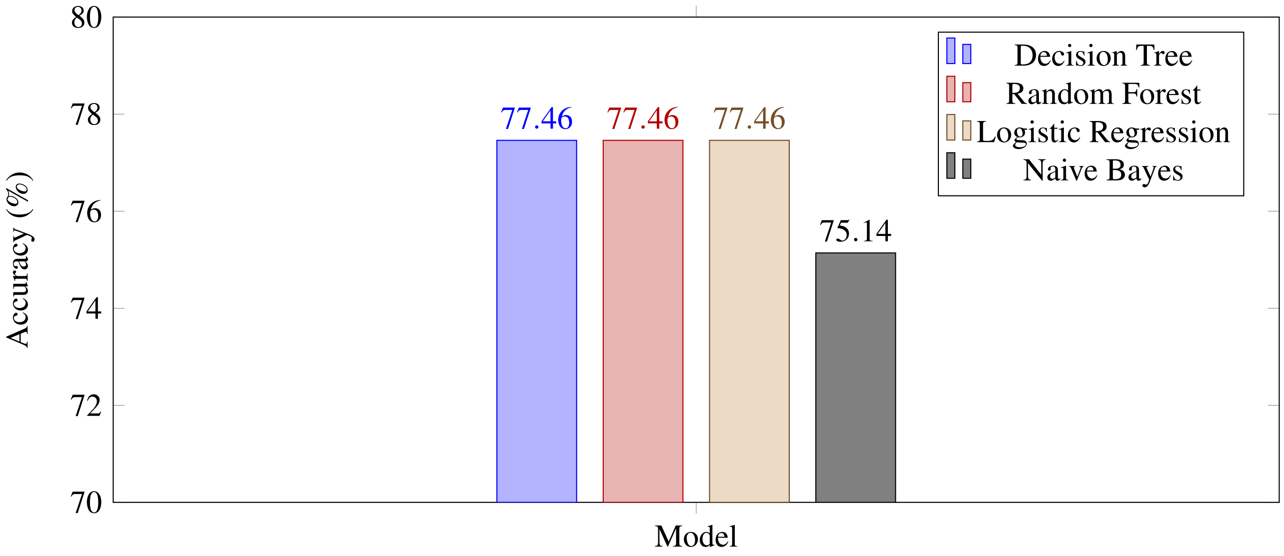
Once the least relevant features are identified, they are dropped, to reduce the complexity of the model, focusing only on key features. Next, the missing values are filled with median values of the respective feature. Then, still in the pre-processing phase, the key features are converted from string to numerical and from continuous ranges to categories to ease the model training.

The dataSet is then split randomly into traning and test set following the 80:20 Pareto Principle [2]. Next the data are transformed in vectors thanks to methods offered by the library *apache.spark.ml* and *apache.spark.mllib* and prepared for the model training. After the models are trained on the training set, the predictions are made and the accuracy of the four models is computed.

5 Results

The classification models used on this project were: Decision Tree, Random Forest, Linear Regression and Naive Bayes. They were trained on eight features ['PassengerId', 'Pclass', 'Title', 'Sex', 'Age', 'Fare', 'Embarked', 'BeingAlone'] to predict the label 'Survived'. Their accuracy was derived considering how many samples were predicted correctly, just buy confronting the predictions' vector with the removed results from test set.

At the first training on the training set, the accuracy of the methods resulted as follow:



As expected, Decision Tree and Random Forest have performed similarly. Logistic Regression has also performed in the same way, even if the predictions among the three were different. Different ways to split the data set and different trainings might effect the performance of Logistic Regression against Decision Tree and Random Forest. The less performing model has been Naive Bayes. The model performances were in line with results obtained by other participants to the challenge, even if they were running their models on slightly different and broader data sets.

6 How to run the project

The project was developed on the Databricks cloud data platform. The free license we used is the Databricks Community Edition. Once you have signed up and logged in in the platform, the first thing you need to do is entering the *Data Science and Engineering* section, going into the menu *Compute* and create and start your own server cluster.

Once the server cluster is running, you have to go in the menu *Workspace*, create a new folder where to run your project and import in it the notebook *main.scala* present on our repository, through the option *Import > File*. When you have opened it, open the tab *Connect* and load your notebook on the cluster by selecting the cluster server you have created.

Now, into the menu *Data* you have to import the file *dataset.csv* present in our repository, through the option *Create Table > Upload File*. Once you have selected the file *dataset.csv*, you need to press on *Create Table with UI* then select the cluster you have created above. Now, in the *table preview* window, you need to select *first row is header* and *infer schema* and **to rename the file as *dataset***. Now, pressing on *Create Table* you will have the access to the necessary data to run the notebook.

Now you can return on the notebook *main.scala* and run the all project, by pressing on the *Run All* button.

References

[1] W. Curkierski, "Titanic - Machine Learning from Disaster — kaggle.com," https://www.kaggle.com/c/titanic, 2012.

[2] R. Sanders, "The parete principle: its use and abuse," *Journal of Services Marketing*, vol. 1, no. 2, pp. 37–40, feb 1987. doi: 10.1108/eb024706. [Online]. Available: https://doi.org/10.1108/eb024706