UNIVERSITAT POLITÈCNICA DE CATALUNYA

CIÈNCIA I ENGINYERIA DE DADES

REPORT - PREDICTIVE ANALYTICS USING BIG DATA TECNOLOGIES

PRESENTED BY

Armand de Asís
Martí Farré

SUBJECT

BASES DE DADES AVANÇADES

January 3, 2023

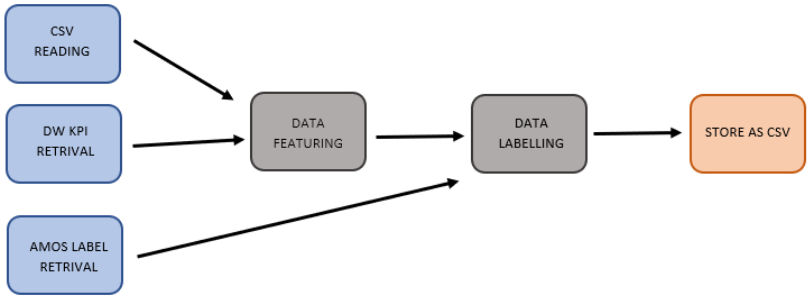
On of the main focuses of our project has been having a code as efficient as possible. To begin with, we have found that the most efficient partition when parallelizing is the number of cores our local machine has, so we ask the user to input it. The user can then change it if it wants to. With this partition and several other optimization techiniques we will later explain, we have improved the efficiency by 400%. For example, in the first pipeline we have improved our execution time from 200 s to 50 s.

Data Management Pipeline: The main things worth noting in this pipeline is that we have used coalesce before doing the group by to merge all the partitions used to read the csv. The group by is done after reading all the csv to have an accurate average calculation. After grouping by, we have sorted the data by aircraftid and timeid to increase the performance of future joins. When we read the data of the DW, we convert the jdbc floats to integers (we can't have 2.5 flight cycles) and we also sort by the same values. We join the csv and DW data with an inner join because we don't want airplanes with no sensors data. This join is optimized thanks to both sorts. To label the data we select from the maintenanceevent table the kind column, and as you mentioned, we consider as a maintenance both scheduled and unscheduled maintenance. To predict the 7 days prior to the maintenance events, we take the start date from the aircraft that required a maintenance and we iterate the 7 days before the event and we mark them as Maintenance. We also delete duplicates, because we don't care which kind is. After that we do a right join with the features. Then we save the features as a csv.

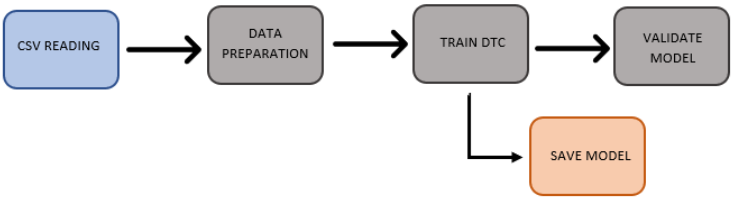
Data Analysis Pipeline: We read the csv we prepared in the previous pipeline as a fixed structure so we don't have errors, and we vectorize it as a feature vector with a label that is 1 if we have a maintenance, and 0 if we don't need a maintenance. We split the data set between training and validation with a prefixed partition of 70-30, but this can be modified. We train a Decision Tree Classifier and we save the model for the next pipeline. We validate the model with binary classification evaluator and we compute its accuracy and recall. The recall is defined as $\frac{TP}{TP+FN}$, so it's how many aircraft our model correctly predicted that they would need maintenance over the total amount of aircrafts. The results vary depending on how we train the model and the split of the data we use, but it ranges from 70% to 95% for accuracy and from 40% to 90% for the recall. With the 70-30 partition we got 91% accuracy and 78% recall.

Run Time Classifier Pipeline: In this pipeline we ask the user for an aircraftid and a timeid to predict if it will require maintance in the next 7 days. We obtain the sensors average and the KPI's value as we did on the Data Management Pipeline, but filtering by aircraftid and timeid. Here we don't need to sort because we only have one instance. Once we have the row with the data, we convert it to a feature vector but with no label, because we don't know if the aircraft will need maintenance in the following days. We finally display to the user if the aircraft will need maintenance in the following days.

Data Management Pipeline:



Data Analysis Pipeline:



Run Time Classifier Pipeline: