Algoritmos para Big Data
Lab class handout #4

Machine learning problem – binary classification **2024/2025**

Learning objectives

This lab class is about binary classification in a discrete space. We will setup a ML processing pipeline to achieve the goals, and the data to be considered relates to the domain of banking industry. Specifically, it is a case of fraud detection in credit cards transactions.

Note: It is acceptable that the work to be undertaken may require some time beyond the class. If that is the case, work should continue outside of class. And if needed, assistance will be provided during the office hours established or to be agreed with the lecturer.

Supporting information

- Course slides
- Machine Learning Library (MLlib) Guide
- Apache Spark ML pipeline

Problem to solve

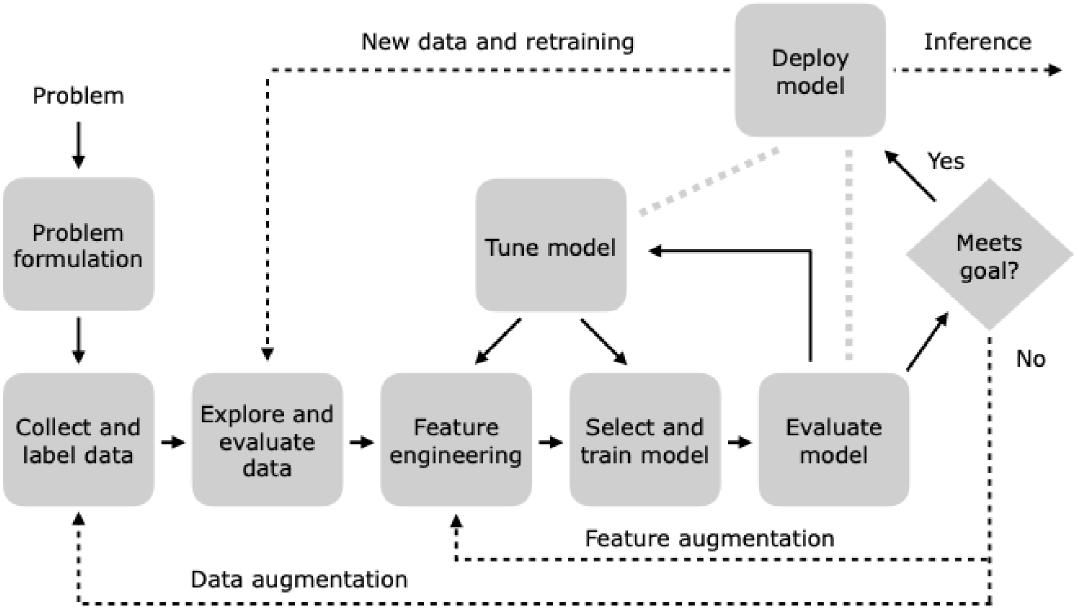
This problem is about credit card fraud detection, aiming to figure out whether a particular credit card transaction is fraudulent or not. Our case-study is based on a Kaggle dataset that holds synthetic data about credit card transactions.

The dataset to be used is in an archive file located at:
`https://bigdata.iscte-iul.eu/datasets/credit-cards-transactions.zip`

Alongside this handout describing the tasks to accomplish, it will be provided a notebook that it is expected to be completed with the implementation of task B.

ML workflow

In order to solve the problem described above, we will setup a ML workflow/pipeline. Recall that a typical ML workflow is designed to work as depicted below:



We will follow the workflow above. On the other hand, we have to consider the practical use of ML pipelines available in Apache Spark. As stated in the documentation: "ML Pipelines provide a uniform set of high-level APIs built on top of DataFrames that help users create and tune practical machine learning pipelines."

Notice that it is possible to combine multiple algorithms into a single pipeline. Besides DataFrames, the implementation involves the following concepts:

1. Transformer: an algorithm which can transform one DataFrame into another DataFrame. For example, an ML model is a Transformer, which transforms a DataFrame with features into a DataFrame with predictions.
2. Estimator: an algorithm which can be fit on a DataFrame to produce a Transformer. For example, a learning algorithm is an Estimator, which is training on a DataFrame and produces a model.
3. Pipeline: the way to chain multiple Transformers and Estimators together in order to specify a ML workflow.
4. Parameter: all Transformers and Estimators share a common API for specifying parameters.

Tasks to accomplish

A. Data ingestion, data preparation and understanding

In this initial task, execute the following operations:

1. Download the data file already mentioned and save it in a proper directory, within the working directory of the Pyspark instalation.

2. Create a notebook specifically for the implementation of this task. In that respect, it is worth looking at previous notebooks that were provided.

3. After creating a *SparkSession* to work with, read the data into a DataFrame and then check it. Make sure that the data is prepared/cleaned to be worked with. For example, check (i) if there are duplicated rows or not, (ii) the existence of NULLs and (iii) if datatypes are properly set or not.

   Hereafter, we will consider the name of the DataFrame containing the data of concern as `df_transactions`.

4. With `df_transactions`, carry out some exploratory data analysis based on descriptive analytics and visualizations. For example:
   - Use the statistical method *describe*() to figure out outliers.
   - Use the method *distinct*() to find unique values in columns of interest.
   - Compute correlations among numerical columns (with no NULLs), and plot them using *plotly.express*.
   - Compute various aggregations on some columns of interest, and plot them using *plotly.express*. Cases to be considered can be:
     - Number of transactions by (i) year and (ii) month.
     - Counting regarding the channel used in transactions (chip usage).
     - Counting of fraudulent transactions versus all transactions, and checking the kind of fraudulent transactions.
     - Counting regarding the channel used in fraudulent transactions (chip usage).
     - Counting and maximum amount regarding fraudulent transactions, by hour.

   If from the analysis above changes on `df_transactions` are warranted, including getting rid of some columns, or creating new columns derived from existing ones, do so.

5. Regardless of changes being made or not, create another dataframe after `df_transactions`, to be named `df_transactions_small`, which will be a 30% sampling (Feel free to use a different sampling fraction, but substantially lower than the maximum).

   Save these two dataframes as parquet files. This concludes the first notebook.

B. ML classifier model

This task will be implemented in the notebook provided. It is assumed that the data regarding `df_transactions_small` mentioned in the previous task has been stored as a parquet file. This is the data that will be used to create the classifier model.

The remaining text will be available by the time of the second lecture assigned to this lab class handout.