WRITTEN ANALYSIS: TENSOR FLOW MODEL

A screenshot of a computer program

Description automatically generated

I decided to work on Google colab, in order to build our machine learning I started by installing pyspark and findspark.

A screen shot of a computer

Description automatically generated

In order to read the CSVs created previously I had to authorize Google Drive to link the CSVs to this Google colabotary notebook.

A screenshot of a computer program

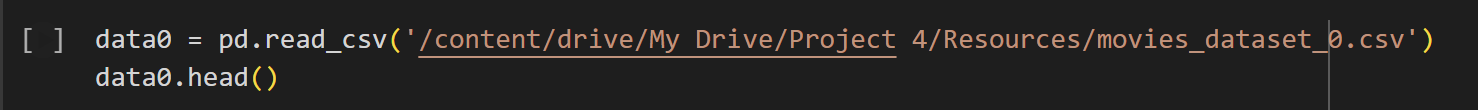
Description automatically generated

Imported dependencies and packages.

A screenshot of a computer program

Description automatically generated

Showing the content of my Google drive in order to create the path to read the CSVs.



Reading all of the CSVs.

A screenshot of a computer program

Description automatically generated

Initializing Spark Session

A screenshot of a computer

Description automatically generated

The following parameters are used to customize the reading process:

* **str(data0)**: The path to the CSV file to be read. The **str()** function is used to convert the **data0** variable to a string, which is required by the **spark.read.csv()** method.
* **sep=","**: Specifies that the CSV file uses commas (**,**) as the separator between values.
* **header=True**: Indicates that the first row of the CSV file contains column headers, which will be used as column names in the resulting DataFrame.
* **inferSchema=True**: Tells Spark to automatically infer the data types of each column based on the data in the CSV file.
* **quote='"'**: Specifies that quoted strings in the CSV file are enclosed in double quotes (**"**) and should be handled accordingly.
* **escape='"'**: Specifies that the double quote (**"**) is also used as the escape character for embedded quotes within quoted strings.
* **multiLine=True**: Enables the handling of multiline fields in case of long text, which means that Spark will correctly parse fields that span multiple lines.

In summary, this code reads a CSV file into a Spark DataFrame, using commas as separators, with automatic schema inference, and handling quoted strings, embedded quotes, and multiline fields.

A screen shot of a computer program

Description automatically generated

We then decided to union the CSVs in order to start building the model. We chose unioning because it provides a more scalable, flexible, and efficient way to combine data from multiple files, especially when dealing with different schemas or large datasets.

A black screen with white text

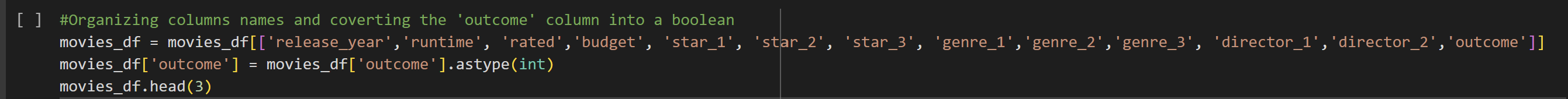
Description automatically generated

Converted the CSV into a Pandas dataframe for easier use.

A screenshot of a computer program

Description automatically generated

We then cleaned the dataframe by checking the columns, dropped the unnecessary columns and checking the data types.



We continues with organizing the column names and converting the ‘outcome’ column into a Boolean.