Team members:
Rob Sisto, Annemarie Andaleon, Jacob Plaza, Melchor Ronquillo

COMP358 Project 1

Dataset description: "Student Alcohol Consumption"

Our dataset is from kaggle and uploaded by UCI Machine Learning. The data was obtained from a survey of secondary school students in Portugal. This survey primarily focuses on students and their alcohol consumption. Alcohol consumption is measured by 2 variables, WALC (Weekend Alcohol Consumption) and DALC (Workday Alcohol Consumption). Both are on a scale from 1 being very low to 5 being very high. To see what contributes to or what is affected by alcohol consumption, there are many other variables included within the dataset specific to each student. There is demographic information such as age, sex, and address, family information such as parents relationship, occupations, education, etc, and many other external information such as health condition, amount of time spent studying, traveling to school, going out, internet at home, and many others.

Link to Dataset:
https://www.kaggle.com/datasets/uciml/student-alcohol-consumption

*****Work for addressing each of data analytics pipeline phases:**
Identification of Data:

Process? Kaggle → Upload / export for Index column → create table in hbase → populate table with data (mapping) → create view -> query

Our group wanted to explore the factors that contribute to alcoholism in students, as well as what factors are affected by a student's alcoholism.

To conduct this investigation, we used Kaggle to source our data from UCI Machine Learning. The dataset did not need cleansing, although we went through and identified some possible points of interest in our exploration. Some of these were study time, travel time, time spent browsing the internet, and how often students go out with friends.

In the extraction step, we found that our data set did not have row keys, which created errors when trying to use it with Hbase and Phoenix. To fix this, we added in an index by quickly passing it through R, which then resolved the issue and made it fit for analysis.

For the aggregation and subsequent steps of the pipeline, we then created a table in Hbase using two column families: "student alcohol, and "order". We then mapped the data onto the table in Hbase, Switching into Phoenix, we constructed a view, therefore making our data possible to query.

Evaluate: The motive, scope, and parameters of the analysis are identified.
Identify the data: Various data sources required for analysis are identified.
Data filtering: Includes data cleansing, data normalization, noise removal, missing data, and so on.
Data extraction: Assures the collected data is fit for analysis, e.g.,the data is compatible with the tool used for analysis.
Data aggregation: Assures the collected data is fit for analysis, e.g.,the data is compatible with the tool used for analysis.
Data analysis and visualization: Analytical tools are used to perform the designed analysis. Then analysis is graphically communicated using tools like tableau.
analysis outcome: The analysis result is available for various decision making.

Evaluate: Our group wanted to determine the factors that contribute to alcoholism in students, as well as what factors are affected by a student's alcoholism.

Scripts used for building HBase / Phoenix tables and views:

HBase Table

- hbase shell
- create 'student_alcohol', 'order'
- exit
- su root
- hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.separator=, -Dimporttsv.columns="HBASE_ROW_KEY, order:school, order:sex, order:age, order:address, order:famsize, order:pstatus, order:medu, order:fedu, order:mjob, order:fjob, order:reason, order:guardian, order:traveltime, order:studytime, order:failures, order:schoolsup, order:famsup, order:paid, order:activities, order:nursery, order:higher, order:internet, order:romantic, order:famrel, order:freetime, order:goout, order:dalc, order:walc, order:health, order:absences, order:g1, order:g2, order:g3" student_alcohol /tmp/student_alcohol.csv
- su hbase
- hbase shell
- scan 'student_alcohol'

Phoenix View:

- create view "student_alcohol" (
- "row" VARCHAR primary key,
- "order"."school" VARCHAR,
- "order"."sex" VARCHAR,
- "order"."age" VARCHAR,
- "order"."address" VARCHAR,
- "order"."famsize" VARCHAR,
- "order"."pstatus" VARCHAR,

- "order"."medu" VARCHAR,
- "order"."fedu" VARCHAR,
- "order"."fjob" VARCHAR,
- "order"."reason" VARCHAR,
- "order"."guardian" VARCHAR,
- "order"."traveltime" VARCHAR,
- "order"."studytime" VARCHAR,
- "order"."failures" VARCHAR,
- "order"."famsup" VARCHAR,
- "order"."paid" VARCHAR,
- "order"."activities" VARCHAR,
- "order"."nursery" VARCHAR,
- "order"."higher" VARCHAR,
- "order"."internet" VARCHAR,
- "order"."romantic" VARCHAR,
- "order"."famrel" VARCHAR,
- "order"."freetime" VARCHAR,
- "order"."goout" VARCHAR,
- "order"."dalc" VARCHAR,
- "order"."walc" VARCHAR,
- "order"."health" VARCHAR,
- "order"."absences" VARCHAR,
- "order"."g1" VARCHAR,
- "order"."g2" VARCHAR,
- "order"."g3" VARCHAR);

The queries used for achieving the analysis objectives.

Melchor

1) List the number of students per age group that have a workday alcohol consumption (**Walc)** of greater than or equal to 4 at each school.
    a) Query: SELECT count (1) as Number_of_Students, age, school FROM stu_alcohol WHERE dalc >= ‘4’ GROUP BY age, school ORDER BY count(1) desc;

```
[[quickstart.cloudera:21000] > select count(1) as Number_of_Students, age, school
[                            > from stu_alcohol
[                            > where dalc >= '4'
[                            > group by age, school
[                            > order by count(1) desc;
Query: select count(1) as Number_of_Students, age, school
from stu_alcohol
where dalc >= '4'
group by age, school
order by count(1) desc
+--------------------+-----+--------+
| number_of_students | age | school |
+--------------------+-----+--------+
| 5                  | 17  | GP     |
| 4                  | 16  | GP     |
| 3                  | 18  | MS     |
| 2                  | 15  | GP     |
| 2                  | 18  | GP     |
| 1                  | 22  | GP     |
| 1                  | 20  | MS     |
+--------------------+-----+--------+
Fetched 7 row(s) in 1.35s
```

2) List all the students with their age, sex, address, workday alcohol consumption, weekend alcohol consumption, school, parent status and family relationship who have a workday (**Dalc)** and weekend (**Walc)** alcohol consumption of greater than / equal to 3 (Average level) and are younger than 18 (legal age).
    a) Query: SELECT age, ex, address, dalc, walc,, school, pstatus, famrel FROM stu_alcohol WHERE age <= ‘17’ and dalc >= ‘3’ and walc >= ‘3’ ORDER BY age;

```
[[quickstart.cloudera:21000] > select age, ex, address, dalc, walc, school, pstatus, famrel
[                          > from stu_alcohol
[                          > where age <= '17' and dalc >= '3' and walc >= '3'
[                          > order by age;
Query: select age, ex, address, dalc, walc, school, pstatus, famrel
from stu_alcohol
where age <= '17' and dalc >= '3' and walc >= '3'
order by age
+-----+----+---------+------+------+--------+---------+--------+
| age | ex | address | dalc | walc | school | pstatus | famrel |
+-----+----+---------+------+------+--------+---------+--------+
| 15  | M  | U       | 3    | 4    | GP     | T       | 5      |
| 15  | M  | R       | 3    | 5    | GP     | T       | 1      |
| 15  | M  | U       | 3    | 4    | GP     | A       | 5      |
| 15  | F  | U       | 4    | 4    | GP     | A       | 5      |
| 15  | M  | U       | 5    | 5    | GP     | A       | 1      |
| 16  | M  | U       | 5    | 5    | GP     | T       | 4      |
| 16  | M  | R       | 3    | 5    | GP     | T       | 3      |
| 16  | M  | R       | 3    | 4    | GP     | T       | 4      |
| 16  | F  | U       | 5    | 5    | GP     | T       | 5      |
| 16  | M  | R       | 3    | 4    | GP     | T       | 4      |
| 16  | M  | U       | 4    | 4    | GP     | T       | 4      |
| 16  | M  | U       | 3    | 5    | GP     | A       | 4      |
| 16  | M  | U       | 3    | 5    | GP     | T       | 4      |
| 16  | F  | U       | 3    | 3    | GP     | T       | 3      |
| 16  | M  | U       | 5    | 5    | GP     | T       | 4      |
| 17  | M  | U       | 3    | 4    | MS     | T       | 2      |
| 17  | M  | R       | 5    | 5    | GP     | T       | 4      |
| 17  | M  | U       | 3    | 5    | GP     | T       | 4      |
| 17  | M  | U       | 3    | 4    | GP     | A       | 4      |
| 17  | M  | U       | 3    | 4    | GP     | T       | 5      |
| 17  | M  | U       | 5    | 5    | GP     | T       | 4      |
| 17  | M  | U       | 4    | 5    | GP     | T       | 5      |
| 17  | F  | U       | 3    | 4    | GP     | T       | 4      |
| 17  | M  | U       | 4    | 5    | GP     | T       | 4      |
| 17  | M  | U       | 4    | 4    | GP     | T       | 4      |
| 17  | M  | R       | 3    | 3    | GP     | T       | 2      |
+-----+----+---------+------+------+--------+---------+--------+
Fetched 26 row(s) in 1.27s
[quickstart.cloudera:21000] >
```

Rob

1) List of all the students who want to take higher education, their mothers, their father's education, and their sex (**Medu, Fedu, Sex**)
    a) `select "medu", "fedu", "sex" from "student_alcohol" where "higher"='"yes"';`

```
0: jdbc:phoenix:> select "medu", "fedu", "sex" from "student_alcohol" where "higher"='"yes"' limit 10;
+-------+-------+------+
| medu  | fedu  | sex  |
+-------+-------+------+
| 4     | 4     | "F"  |
| 3     | 4     | "M"  |
| 4     | 3     | "F"  |
| 4     | 4     | "M"  |
| 4     | 4     | "M"  |
| 4     | 4     | "M"  |
| 3     | 2     | "F"  |
| 3     | 4     | "M"  |
| 3     | 3     | "F"  |
| 2     | 2     | "F"  |
+-------+-------+------+
10 rows selected (0.386 seconds)
```

2) List of students with their addresses, travel time greater than 30 min and school (**Address, Travel Time, School)** - traveltime(numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)
   a) `select "address","school" from "student_alcohol" where "traveltime">'2';`

```
0: jdbc:phoenix:> select "address", "school" from "student_alcohol" where "traveltime">'2' limit 10;
+----------+---------+
| address  | school  |
+----------+---------+
| "R"      | "GP"    |
| "R"      | "GP"    |
| "U"      | "GP"    |
| "U"      | "GP"    |
| "R"      | "GP"    |
| "R"      | "GP"    |
| "U"      | "GP"    |
| "R"      | "GP"    |
| "R"      | "GP"    |
| "R"      | "GP"    |
+----------+---------+
10 rows selected (0.065 seconds)
```

Annemarie

1) List all the family relationship qualities (**famrel**) for students whose parents are separated (**Pstatus**)

a) `select "famrel" from "student_alcohol" where "pstatus" <= '2';`

```
0: jdbc:phoenix:> select "famrel" from "student_alcohol" where "pstatus"<='2' limit 10;
+-----------+
|  famrel   |
+-----------+
| "famrel"  |
| 4         |
| 5         |
| 5         |
| 4         |
| 4         |
| 5         |
| 4         |
| 5         |
| 4         |
+-----------+
10 rows selected (0.035 seconds)
```

2) List all the students and the number of failures they've had based on if they've had low health (**traveltime**)
   a) `select "failures", "health" from "student_alcohol" where "health"<'3';`

```
0: jdbc:phoenix:> select "failures","health" from "student_alcohol" where "health"<'3' limit 10;
+-------------+-----------+
|  failures   |  health   |
+-------------+-----------+
| "failures"  | "health"  |
| 0           | 2         |
| 0           | 1         |
| 0           | 1         |
| 0           | 2         |
| 1           | 2         |
| 0           | 2         |
| 0           | 1         |
| 0           | 1         |
| 2           | 2         |
+-------------+-----------+
10 rows selected (0.058 seconds)
```

Jacob

1. List all students who have a long travel time and who very rarely go out with friends
   a. `Select "traveltime", "goout" from "student_alcohol" where "traveltime" >= '3' and "goout" = '1';`

```
0: jdbc:phoenix:> select "traveltime", "goout" from "student_alcohol" where "traveltime" >= '3
' and "goout" = '1';
+-------------+--------+
| traveltime  | goout  |
+-------------+--------+
| 3           | 1      |
| 3           | 1      |
| 3           | 1      |
| 3           | 1      |
| 3           | 1      |
+-------------+--------+
5 rows selected (0.083 seconds)
```

2. List all students who have large amounts of study time (4 to 10 hours) and who have high weekend alcohol consumption
   a. `Select "studytime", "walc" from "student_alcohol" where "studytime" >= '3' and "walc" > '4';`

```
0: jdbc:phoenix:> select "studytime", "walc" from "student_alcohol" where "studytime" >= ' 3'
and "walc" > '4' limit 10;
+------------+-------+
| studytime  | walc  |
+------------+-------+
| 1          | 5     |
| 4          | 5     |
| 1          | 5     |
| 1          | 5     |
| 1          | 5     |
| 1          | 5     |
| 1          | 5     |
| 2          | 5     |
| 2          | 5     |
| 2          | 5     |
+------------+-------+
10 rows selected (0.093 seconds)
```

Instructions to reproduce results (Virtual Box w/ Phoenix):

#to copy files to hortonworks
#note: you have to specify the file where the data will be written
scp -P 2222 student_alcohol.csv root@127.0.0.1:/student_alcohol.csv

#to copy files from hortonworks
scp -P 2222 root@127.0.0.1:/data.csv data.csv

#to copy a directory use the -r command
scp -P 2222 -r root@127.0.0.1:/src src

#steps to insert the csv into hdfs (make sure the student_alchol.csv is already in your vm)

- hadoop fs -put /student_alcohol.csv /tmp
- su hbase
-

- hbase shell
- create 'student_alcohol', 'order'
- exit
- su root
- hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.separator=, -Dimporttsv.columns="HBASE_ROW_KEY, order:school, order:sex, order:age, order:address, order:famsize, order:pstatus, order:medu, order:fedu, order:mjob, order:fjob, order:reason, order:guardian, order:traveltime, order:studytime, order:failures, order:schoolsup, order:famsup, order:paid, order:activities, order:nursery, order:higher, order:internet, order:romantic, order:famrel, order:freetime, order:goout, order:dalc, order:walc, order:health, order:absences, order:g1, order:g2, order:g3" student_alcohol /tmp/student_alcohol.csv
- su hbase
- hbase shell
- scan 'student_alcohol'
- exit
- exit
- cd /usr/hdp/current/phoenix-client/bin
- ./sqlline.py
-
- create view "student_alcohol" (
- "row" VARCHAR primary key,
- "order"."school" VARCHAR,
- "order"."sex" VARCHAR,
- "order"."age" VARCHAR,
- "order"."address" VARCHAR,
- "order"."famsize" VARCHAR,
- "order"."pstatus" VARCHAR,
- "order"."medu" VARCHAR,
- "order"."fedu" VARCHAR,
- "order"."fjob" VARCHAR,
- "order"."reason" VARCHAR,
- "order"."guardian" VARCHAR,
- "order"."traveltime" VARCHAR,
- "order"."studytime" VARCHAR,
- "order"."failures" VARCHAR,
- "order"."famsup" VARCHAR,
- "order"."paid" VARCHAR,
- "order"."activities" VARCHAR,
- "order"."nursery" VARCHAR,
- "order"."higher" VARCHAR,
- "order"."internet" VARCHAR,
- "order"."romantic" VARCHAR,
- "order"."famrel" VARCHAR,

- "order"."freetime" VARCHAR,
- "order"."goout" VARCHAR,
- "order"."dalc" VARCHAR,
- "order"."walc" VARCHAR,
- "order"."health" VARCHAR,
- "order"."absences" VARCHAR,
- "order"."g1" VARCHAR,
- "order"."g2" VARCHAR,
- "order"."g3" VARCHAR);

- Select * from “student_alcohol” limit 10;

Instructions to reproduce outcomes (Docker w/ impala):

- Import csv into python and export csv with an index column
    - Important when mapping data to table
    - If no index column, then index will become the schools as it is the first column
- Upload csv to shared folder
    - Path: /Users/melchorronquillo/Desktop/COMP358/student_alcohol2.csv
- Start docker and attach to running container
    - Docker ps
    - Docker attach _ _ _ _ (first 4 of container ID)
- Switch to cloudera
    - Su cloudera
- Check if file is in shared folder
    - ls /src
    - cat /src/student_alcohol2.csv | head

- Move file to hdfs
    - hdfs dfs -put /src/student_alcohol2.csv /tmp
    - To check if in hdfs
        - hdfs dfs -ls /tmp
        - Hdfs dfs -cat /tmp/student_alcohol2.csv | head
- Open hbase and create table with respective column family
    - Hbase shell
    - Create ‘student_alcohol’, ‘info’
    - list
    - !describe ‘student_alcohol’
    - Exit
- Map data into table to populate
    - hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.separator=, -Dimporttsv.columns="HBASE_ROW_KEY,info:school,info:sex,info:age,info:address,info:famsize,info:pstatus,info:medu,info:fedu,info:mjob,info:fjob,info:reason,info:guardian,info:traveltime,info:studytime,info:failures,info:schoolsup,info:famsup,info:paid,info:activities,info:nursery,info:higher,info:internet,info:romantic,info:famrel,info:freetime,info:goout,info:dalc,info:walc,info:health,info:absences,info:g1,info:g2,info:g3" student_alcohol /tmp/student-mat.csv
- Open hbase to check table
    - hbase shell
    - list
    - scan ‘student_alcohol’
    - Exit
- Open hive to create an external table that is mapped to hbase table
    - Hive shell
    - CREATE EXTERNAL TABLE stu_alcohol(id string,school string,ex string,age string,address string,famsize string,pstatus string,medu string,fedu string,mjob string,fjob string,reason string,guardian string,traveltime string,studytime string,failures string,schoolsup string,famsup string,paid string,activities string,nursery string,higher string,internet string,romantic string,famrel string,freetime string,goout string,dalc string,walc string,health string,absences string,g1 string,g2 string,g3 string) STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH SERDEPROPERTIES ("hbase.columns.mapping"=":key,info:school,info:sex,info:age,info:address,info:famsize,info:pstatus,info:medu,info:fedu,info:mjob,info:fjob,info:reason,info:guardian,info:traveltime,info:studytime,info:failures,info:schoolsup,info:famsup,info:paid,info:activities,info:nursery,info:higher,info:internet,info:romantic,info:famrel,info:freetime,info:goout,info:dalc,info:walc,info:health,info:absences,info:g1,info:g2,info:g3") TBLPROPERTIES ("hbase.table.name" = "student_alcohol");
    - Quit
- Open impala to access external table and query
    - impala-shell

- Invalidate metadata stu_alcohol;
- show tables;
- Query 1
  - List the number of students per age group that have a workday alcohol consumption (**Walc)** of greater than or equal to 4 at each school.
    - SELECT count (1) as Number_of_Students, age, school FROM stu_alcohol WHERE dalc >= ‘4’ GROUP BY age, school ORDER BY count(1) desc;
- Query 2
  - List all the students with their age, sex, address, workday alcohol consumption, weekend alcohol consumption, school, parent status and family relationship who have a workday (**Dalc)** and weekend (**Walc)** alcohol consumption of greater than / equal to 3 (Average level) and are younger than 18 (legal age).
    - Query: SELECT age, ex, address, dalc, walc,, school, pstatus, famrel FROM stu_alcohol WHERE age <= ‘17’ and dalc >= ‘3’ and walc >= ‘3’ ORDER BY age;
- Quit;

- **(20 Points)** A project report that highlights the following:
  - The team members and their tasks
  - The dataset brief description (include the source)
  - The work for addressing each of the data analytics pipeline phases.
- **(20 Points)** The scripts used for building the HBase/Phoenix tables and views.
- **(20 Points)** The queries used for achieving the analysis objectives.
- **(20 Points)** The analysis outcome (Any visualization of results: tables or figures).
- **(20 Points)** The dataset, and brief instructions to reproduce the results.