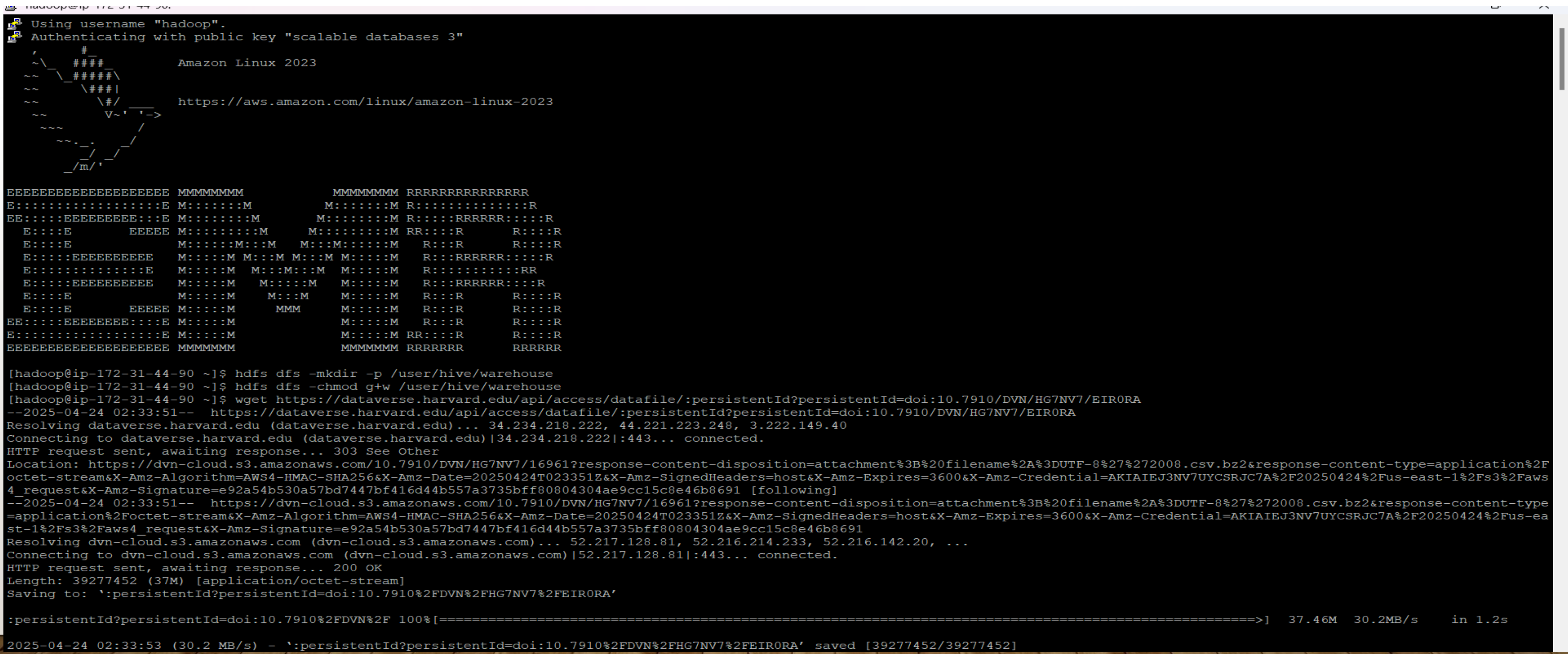
Scalable Project Phase-2

Presented by

Bulty Dolui
Jinhwi Lee
Omkar shinde
Talari Ria

Step 1:-

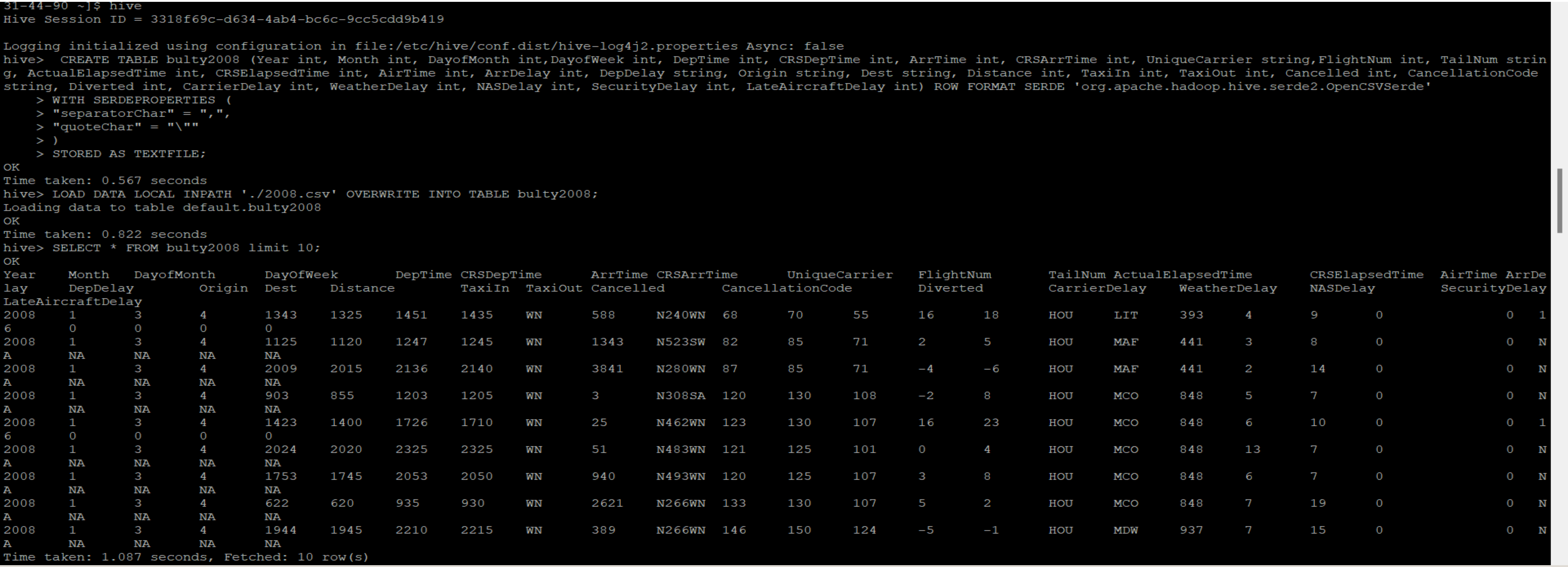
Loading the EMR cluster and importing the 2008 year csv



```
Using username "hadoop".
Authenticating with public key "scalable databases 3"
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-44-90 ~]$ hdfs dfs -mkdir -p /user/hive/warehouse
[hadoop@ip-172-31-44-90 ~]$ hdfs dfs -chmod g+w /user/hive/warehouse
[hadoop@ip-172-31-44-90 ~]$ wget https://dataverse.harvard.edu/api/access/datafile/:persistentId?persistentId=doi:10.7910/DVN/HG7NV7/EIR0RA
--2025-04-24 02:33:51--  https://dataverse.harvard.edu/api/access/datafile/:persistentId?persistentId=doi:10.7910/DVN/HG7NV7/EIR0RA
Resolving dataverse.harvard.edu (dataverse.harvard.edu)... 34.234.218.222, 44.221.223.248, 3.222.149.40
Connecting to dataverse.harvard.edu (dataverse.harvard.edu)|34.234.218.222|:443... connected.
HTTP request sent, awaiting response... 303 See Other
Location: https://dvn-cloud.s3.amazonaws.com/10.7910/DVN/HG7NV7/16961?response-content-disposition=attachment%3B%20filename%2A%3DUTF-8%27%272008.csv.bz2&response-content-type=application%2F
octet-stream&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20250424T023351Z&X-Amz-SignedHeaders=host&X-Amz-Expires=3600&X-Amz-Credential=AKIAIEJ3NV7UYCSRJC7A%2F20250424%2Fus-east-1%2Fs3%2Faws
4_request&X-Amz-Signature=e92a54b530a57bd7447bf416d44b557a3735bff80804304ae9cc15c8e46b8691 [following]
--2025-04-24 02:33:51--  https://dvn-cloud.s3.amazonaws.com/10.7910/DVN/HG7NV7/16961?response-content-disposition=attachment%3B%20filename%2A%3DUTF-8%27%272008.csv.bz2&response-content-type
=application%2Foctet-stream&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20250424T023351Z&X-Amz-SignedHeaders=host&X-Amz-Expires=3600&X-Amz-Credential=AKIAIEJ3NV7UYCSRJC7A%2F20250424%2Fus-ea
st-1%2Fs3%2Faws4_request&X-Amz-Signature=e92a54b530a57bd7447bf416d44b557a3735bff80804304ae9cc15c8e46b8691
Resolving dvn-cloud.s3.amazonaws.com (dvn-cloud.s3.amazonaws.com)... 52.217.128.81, 52.216.214.233, 52.216.142.20, ...
Connecting to dvn-cloud.s3.amazonaws.com (dvn-cloud.s3.amazonaws.com)|52.217.128.81|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 39277452 (37M) [application/octet-stream]
Saving to: ':persistentId?persistentId=doi:10.7910%2FDVN%2FHG7NV7%2FEIR0RA'

:persistentId?persistentId=doi:10.7910%2FDVN%2F 100%[===============================================================================================>]  37.46M  30.2MB/s    in 1.2s

2025-04-24 02:33:53 (30.2 MB/s) - ':persistentId?persistentId=doi:10.7910%2FDVN%2FHG7NV7%2FEIR0RA' saved [39277452/39277452]
```

Step 2: -

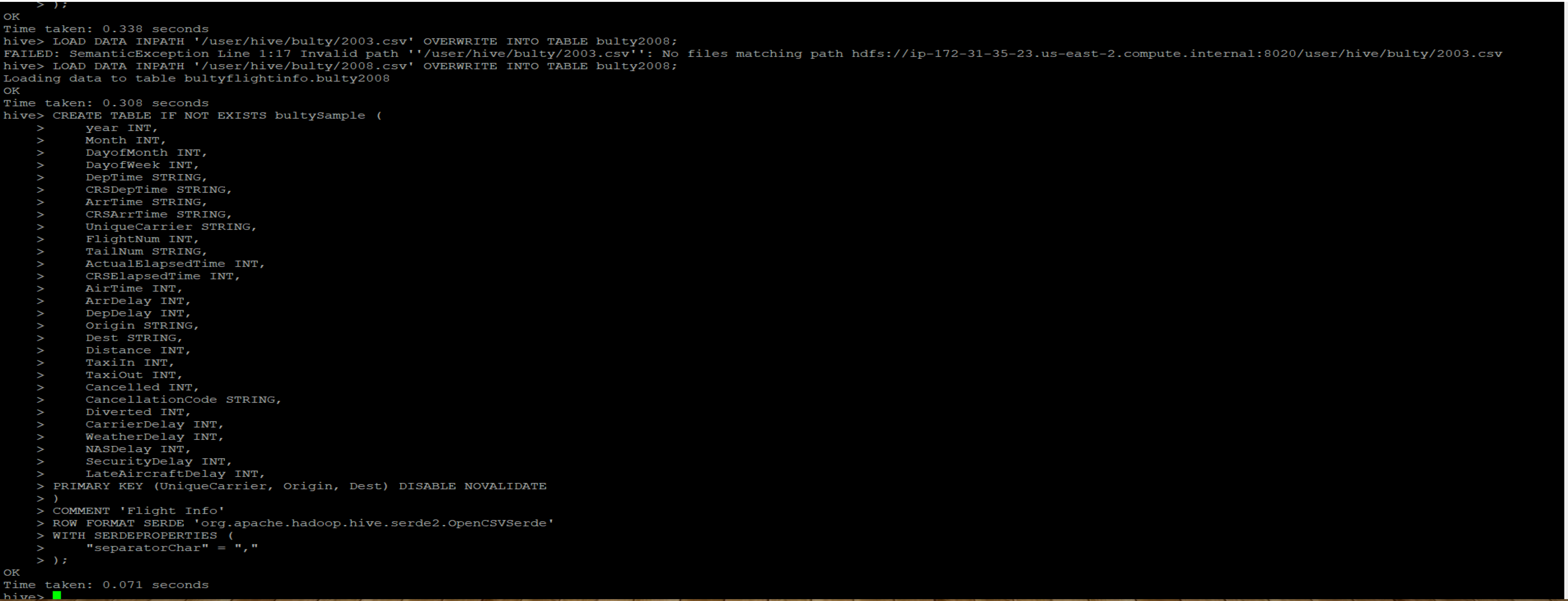
Creating a new table in hive to extract the full dataset from website and load it into the table



```
31-44-90 ~]$ hive
Hive Session ID = 3318f69c-d634-4ab4-bc6c-9cc5cdd9b419

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive>  CREATE TABLE bulty2008 (Year int, Month int, DayofMonth int,DayofWeek int, DepTime int, CRSDepTime int, ArrTime int, CRSArrTime int, UniqueCarrier string,FlightNum int, TailNum strin
g, ActualElapsedTime int, CRSElapsedTime int, AirTime int, ArrDelay int, DepDelay string, Origin string, Dest string, Distance int, TaxiIn int, TaxiOut int, Cancelled int, CancellationCode
string, Diverted int, CarrierDelay int, WeatherDelay int, NASDelay int, SecurityDelay int, LateAircraftDelay int) ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
    > WITH SERDEPROPERTIES (
    > "separatorChar" = ",",
    > "quoteChar" = "\""
    > )
    > STORED AS TEXTFILE;
OK
Time taken: 0.567 seconds
hive> LOAD DATA LOCAL INPATH './2008.csv' OVERWRITE INTO TABLE bulty2008;
Loading data to table default.bulty2008
OK
Time taken: 0.822 seconds
hive> SELECT * FROM bulty2008 limit 10;
OK
Year    Month   DayofMonth      DayOfWeek       DepTime CRSDepTime      ArrTime CRSArrTime      UniqueCarrier   FlightNum       TailNum ActualElapsedTime       CRSElapsedTime  AirTime ArrDe
lay     DepDelay        Origin  Dest    Distance        TaxiIn  TaxiOut Cancelled       CancellationCode        Diverted        CarrierDelay    WeatherDelay    NASDelay        SecurityDelay
LateAircraftDelay
2008    1       3       4       1343    1325    1451    1435    WN      588     N240WN  68      70      55      16      18      HOU     LIT     393     4       9       0               0    1
6       0       0       0       0
2008    1       3       4       1125    1120    1247    1245    WN      1343    N523SW  82      85      71      2       5       HOU     MAF     441     3       8       0               0    N
A       NA      NA      NA      NA
2008    1       3       4       2009    2015    2136    2140    WN      3841    N280WN  87      85      71      -4      -6      HOU     MAF     441     2       14      0               0    N
A       NA      NA      NA      NA
2008    1       3       4       903     855     1203    1205    WN      3       N308SA  120     130     108     -2      8       HOU     MCO     848     5       7       0               0    N
A       NA      NA      NA      NA
2008    1       3       4       1423    1400    1726    1710    WN      25      N462WN  123     130     107     16      23      HOU     MCO     848     6       10      0               0    1
6       0       0       0       0
2008    1       3       4       2024    2020    2325    2325    WN      51      N483WN  121     125     101     0       4       HOU     MCO     848     13      7       0               0    N
A       NA      NA      NA      NA
2008    1       3       4       1753    1745    2053    2050    WN      940     N493WN  120     125     107     3       8       HOU     MCO     848     6       7       0               0    N
A       NA      NA      NA      NA
2008    1       3       4       622     620     935     930     WN      2621    N266WN  133     130     107     5       2       HOU     MCO     848     7       19      0               0    N
A       NA      NA      NA      NA
2008    1       3       4       1944    1945    2210    2215    WN      389     N266WN  146     150     124     -5      -1      HOU     MDW     937     7       15      0               0    N
A       NA      NA      NA      NA
Time taken: 1.087 seconds, Fetched: 10 row(s)
```

Step 3: -

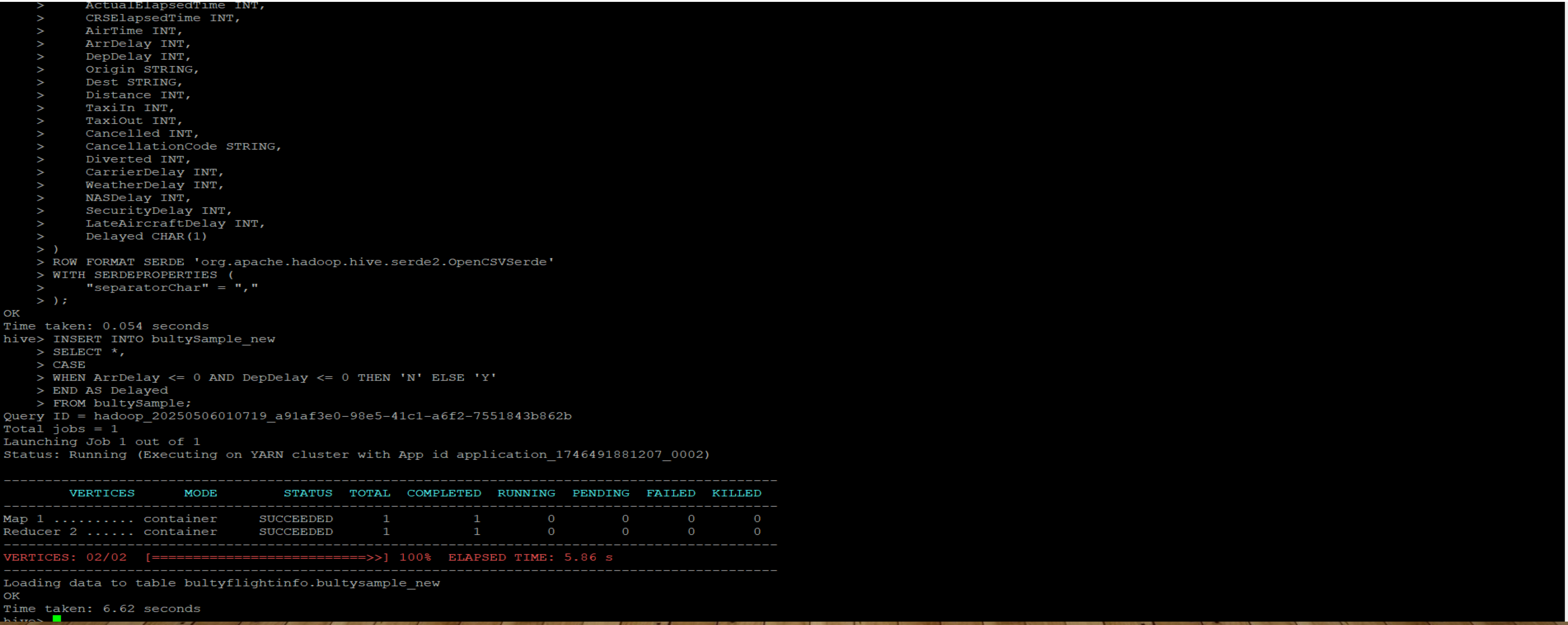
Creating a new table to hold random 30,000 records



```
    > );
OK
Time taken: 0.338 seconds
hive> LOAD DATA INPATH '/user/hive/bulty/2003.csv' OVERWRITE INTO TABLE bulty2008;
FAILED: SemanticException Line 1:17 Invalid path ''/user/hive/bulty/2003.csv'': No files matching path hdfs://ip-172-31-35-23.us-east-2.compute.internal:8020/user/hive/bulty/2003.csv
hive> LOAD DATA INPATH '/user/hive/bulty/2008.csv' OVERWRITE INTO TABLE bulty2008;
Loading data to table bultyflightinfo.bulty2008
OK
Time taken: 0.308 seconds
hive> CREATE TABLE IF NOT EXISTS bultySample (
    >     year INT,
    >     Month INT,
    >     DayofMonth INT,
    >     DayofWeek INT,
    >     DepTime STRING,
    >     CRSDepTime STRING,
    >     ArrTime STRING,
    >     CRSArrTime STRING,
    >     UniqueCarrier STRING,
    >     FlightNum INT,
    >     TailNum STRING,
    >     ActualElapsedTime INT,
    >     CRSElapsedTime INT,
    >     AirTime INT,
    >     ArrDelay INT,
    >     DepDelay INT,
    >     Origin STRING,
    >     Dest STRING,
    >     Distance INT,
    >     TaxiIn INT,
    >     TaxiOut INT,
    >     Cancelled INT,
    >     CancellationCode STRING,
    >     Diverted INT,
    >     CarrierDelay INT,
    >     WeatherDelay INT,
    >     NASDelay INT,
    >     SecurityDelay INT,
    >     LateAircraftDelay INT,
    > PRIMARY KEY (UniqueCarrier, Origin, Dest) DISABLE NOVALIDATE
    > )
    > COMMENT 'Flight Info'
    > ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
    > WITH SERDEPROPERTIES (
    >     "separatorChar" = ","
    > );
OK
Time taken: 0.071 seconds
hive>
```

Step 4: -

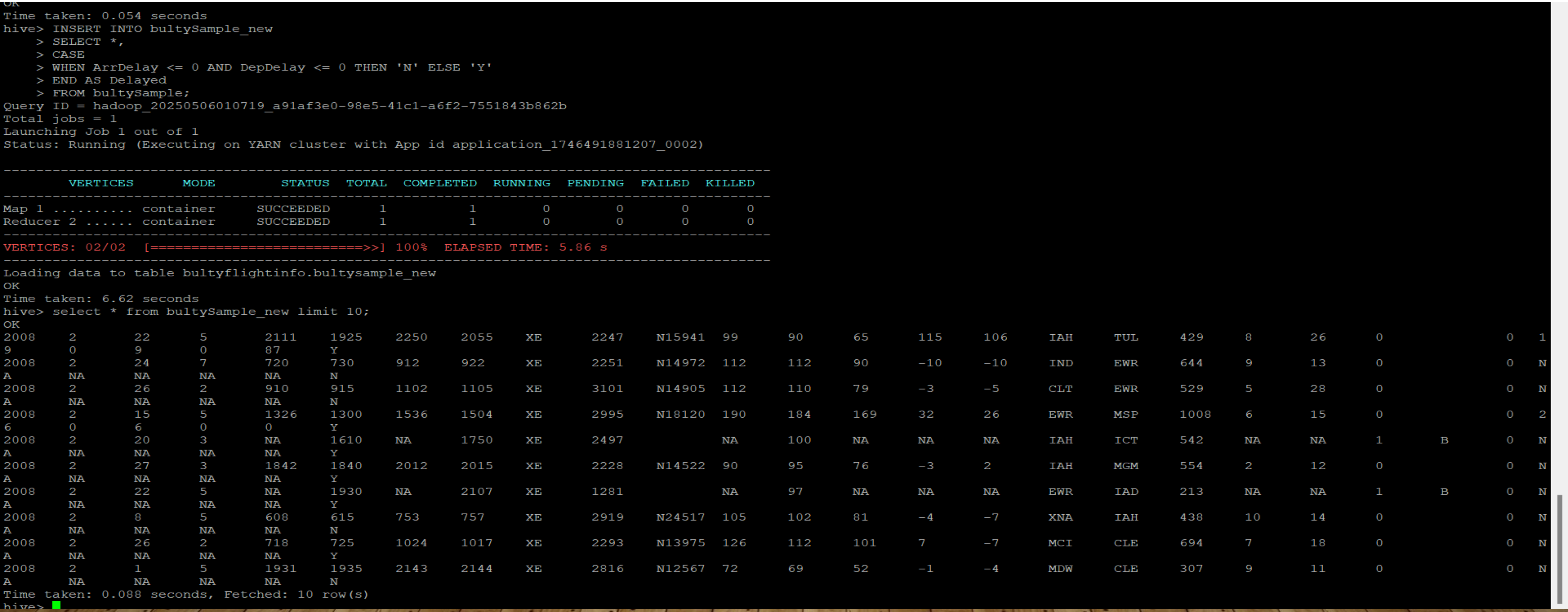
Applying the delay logic to new rows and putting them into a table and adding a column displaying the delay in boolean value



```
    >     ActualElapsedTime INT,
    >     CRSElapsedTime INT,
    >     AirTime INT,
    >     ArrDelay INT,
    >     DepDelay INT,
    >     Origin STRING,
    >     Dest STRING,
    >     Distance INT,
    >     TaxiIn INT,
    >     TaxiOut INT,
    >     Cancelled INT,
    >     CancellationCode STRING,
    >     Diverted INT,
    >     CarrierDelay INT,
    >     WeatherDelay INT,
    >     NASDelay INT,
    >     SecurityDelay INT,
    >     LateAircraftDelay INT,
    >     Delayed CHAR(1)
    > )
    > ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
    > WITH SERDEPROPERTIES (
    >     "separatorChar" = ","
    > );
OK
Time taken: 0.054 seconds
hive> INSERT INTO bultySample_new
    > SELECT *,
    > CASE
    > WHEN ArrDelay <= 0 AND DepDelay <= 0 THEN 'N' ELSE 'Y'
    > END AS Delayed
    > FROM bultySample;
Query ID = hadoop_20250506010719_a91af3e0-98e5-41c1-a6f2-7551843b862b
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1746491881207_0002)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 5.86 s
----------------------------------------------------------------------------------------------
Loading data to table bultyflightinfo.bultysample_new
OK
Time taken: 6.62 seconds
hive>
```

Step 5: -

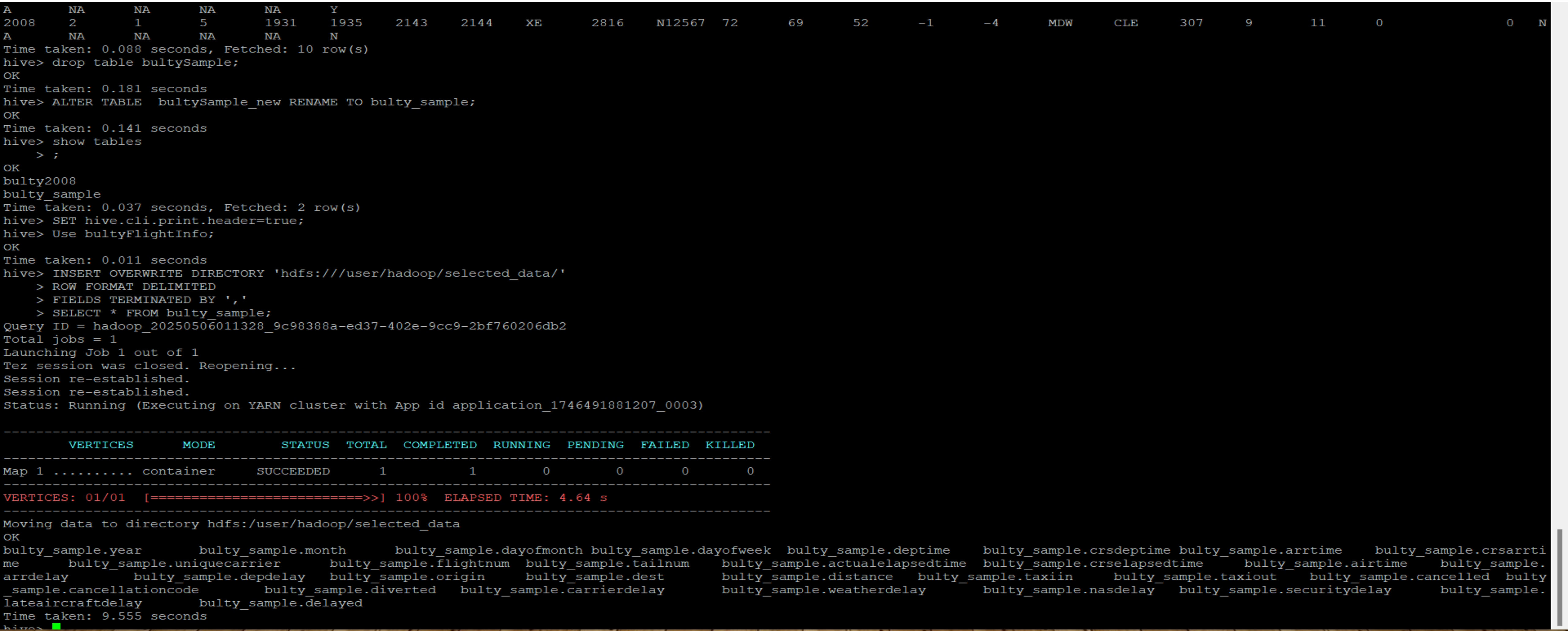
Displaying first 10rows of the new table with delay column



```
OK
Time taken: 0.054 seconds
hive> INSERT INTO bultySample_new
    > SELECT *,
    > CASE
    > WHEN ArrDelay <= 0 AND DepDelay <= 0 THEN 'N' ELSE 'Y'
    > END AS Delayed
    > FROM bultySample;
Query ID = hadoop_20250506010719_a91af3e0-98e5-41c1-a6f2-7551843b862b
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1746491881207_0002)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 5.86 s
----------------------------------------------------------------------------------------------
Loading data to table bultyflightinfo.bultysample_new
OK
Time taken: 6.62 seconds
hive> select * from bultySample_new limit 10;
OK
2008    2       22      5       2111    1925    2250    2055    XE      2247    N15941  99      90      65      115     106     IAH     TUL     429     8       26      0               0   1
9       0       9       0       87      Y
2008    2       24      7       720     730     912     922     XE      2251    N14972  112     112     90      -10     -10     IND     EWR     644     9       13      0               0   N
A       NA      NA      NA      NA      N
2008    2       26      2       910     915     1102    1105    XE      3101    N14905  112     110     79      -3      -5      CLT     EWR     529     5       28      0               0   N
A       NA      NA      NA      NA      N
2008    2       15      5       1326    1300    1536    1504    XE      2995    N18120  190     184     169     32      26      EWR     MSP     1008    6       15      0               0   2
6       0       6       0       0       Y
2008    2       20      3       NA      1610    NA      1750    XE      2497            NA      100     NA      NA      NA      IAH     ICT     542     NA      NA      1       B       0   N
A       NA      NA      NA      NA      Y
2008    2       27      3       1842    1840    2012    2015    XE      2228    N14522  90      95      76      -3      2       IAH     MGM     554     2       12      0               0   N
A       NA      NA      NA      NA      Y
2008    2       22      5       NA      1930    NA      2107    XE      1281            NA      97      NA      NA      NA      EWR     IAD     213     NA      NA      1       B       0   N
A       NA      NA      NA      NA      Y
2008    2       8       5       608     615     753     757     XE      2919    N24517  105     102     81      -4      -7      XNA     IAH     438     10      14      0               0   N
A       NA      NA      NA      NA      N
2008    2       26      2       718     725     1024    1017    XE      2293    N13975  126     112     101     7       -7      MCI     CLE     694     7       18      0               0   N
A       NA      NA      NA      NA      Y
2008    2       1       5       1931    1935    2143    2144    XE      2816    N12567  72      69      52      -1      -4      MDW     CLE     307     9       11      0               0   N
A       NA      NA      NA      NA      N
Time taken: 0.088 seconds, Fetched: 10 row(s)
hive>
```

Step 6: -

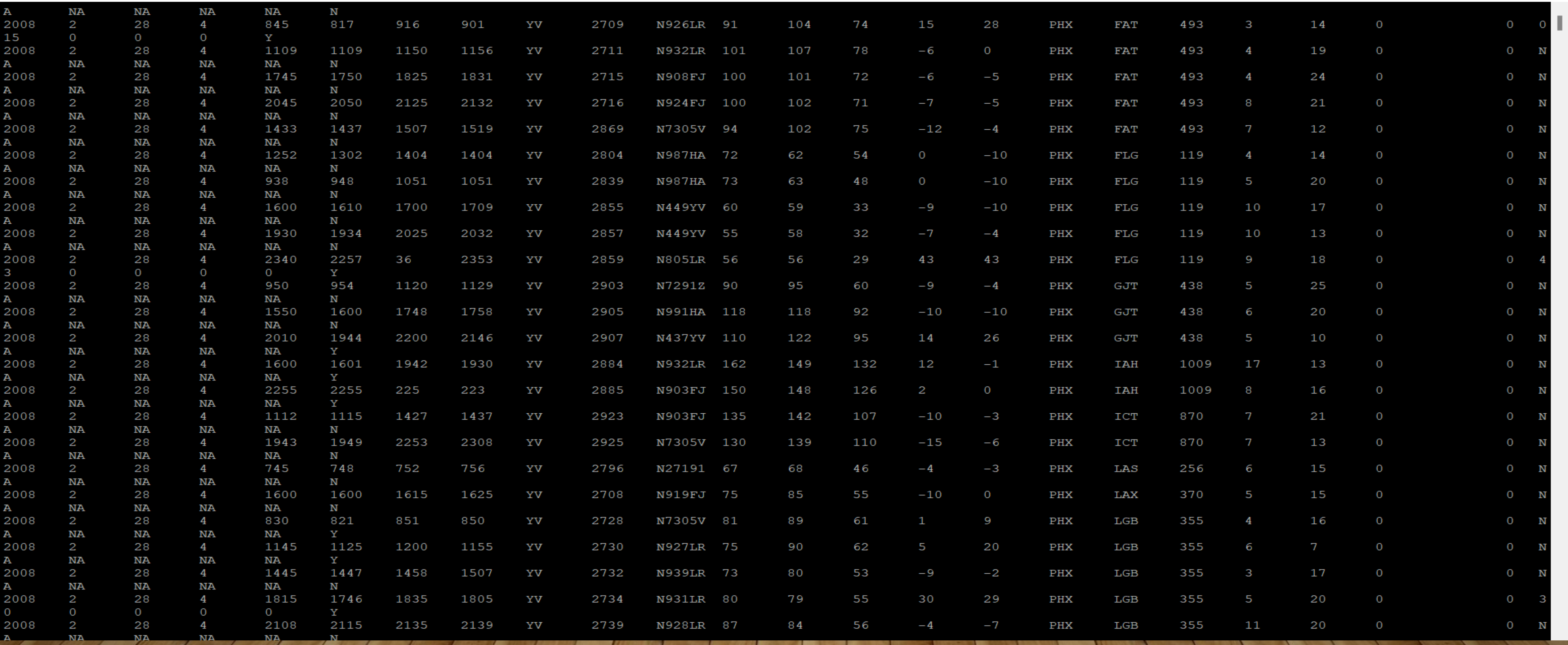
Adding a header to the table



```
A       NA      NA      NA      NA      Y
2008    2       1       5       1931    1935    2143    2144    XE      2816    N12567  72      69      52      -1      -4      MDW     CLE     307     9       11      0                       0   N
A       NA      NA      NA      NA      N
Time taken: 0.088 seconds, Fetched: 10 row(s)
hive> drop table bultySample;
OK
Time taken: 0.181 seconds
hive> ALTER TABLE  bultySample_new RENAME TO bulty_sample;
OK
Time taken: 0.141 seconds
hive> show tables
    > ;
OK
bulty2008
bulty_sample
Time taken: 0.037 seconds, Fetched: 2 row(s)
hive> SET hive.cli.print.header=true;
hive> Use bultyFlightInfo;
OK
Time taken: 0.011 seconds
hive> INSERT OVERWRITE DIRECTORY 'hdfs:///user/hadoop/selected_data/'
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > SELECT * FROM bulty_sample;
Query ID = hadoop_20250506011328_9c98388a-ed37-402e-9cc9-2bf760206db2
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1746491881207_0003)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 01/01  [==========================>>] 100%  ELAPSED TIME: 4.64 s
----------------------------------------------------------------------------------------------
Moving data to directory hdfs:/user/hadoop/selected_data
OK
bulty_sample.year       bulty_sample.month      bulty_sample.dayofmonth bulty_sample.dayofweek  bulty_sample.deptime    bulty_sample.crsdeptime bulty_sample.arrtime    bulty_sample.crsarrti
me      bulty_sample.uniquecarrier      bulty_sample.flightnum  bulty_sample.tailnum    bulty_sample.actualelapsedtime  bulty_sample.crselapsedtime     bulty_sample.airtime    bulty_sample.
arrdelay        bulty_sample.depdelay   bulty_sample.origin     bulty_sample.dest       bulty_sample.distance   bulty_sample.taxiin     bulty_sample.taxiout    bulty_sample.cancelled  bulty
_sample.cancellationcode        bulty_sample.diverted   bulty_sample.carrierdelay       bulty_sample.weatherdelay       bulty_sample.nasdelay   bulty_sample.securitydelay      bulty_sample.
lateaircraftdelay       bulty_sample.delayed
Time taken: 9.555 seconds
hive> 
```

Step 7: -

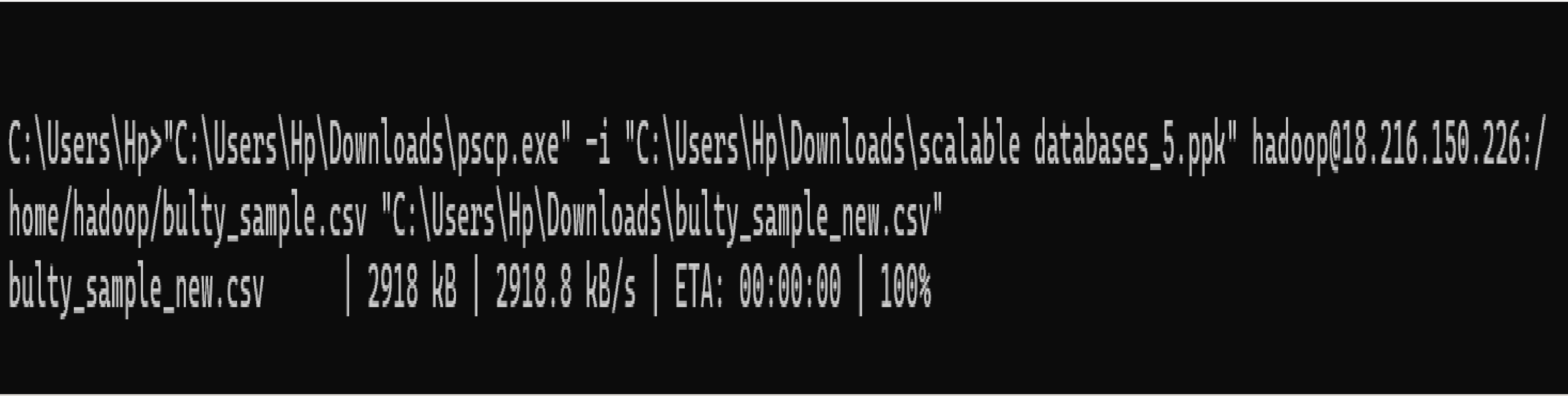
Displaying the new table containing the 30,000 randomly sampled records



```
A       NA      NA      NA      NA      N
2008    2       28      4       845     817     916     901     YV      2709    N926LR  91      104     74      15      28      PHX     FAT     493     3       14      0               0       0
15      0       0       0       Y
2008    2       28      4       1109    1109    1150    1156    YV      2711    N932LR  101     107     78      -6      0       PHX     FAT     493     4       19      0               0       N
A       NA      NA      NA      NA      N
2008    2       28      4       1745    1750    1825    1831    YV      2715    N908FJ  100     101     72      -6      -5      PHX     FAT     493     4       24      0               0       N
A       NA      NA      NA      NA      N
2008    2       28      4       2045    2050    2125    2132    YV      2716    N924FJ  100     102     71      -7      -5      PHX     FAT     493     8       21      0               0       N
A       NA      NA      NA      NA      N
2008    2       28      4       1433    1437    1507    1519    YV      2869    N7305V  94      102     75      -12     -4      PHX     FAT     493     7       12      0               0       N
A       NA      NA      NA      NA      N
2008    2       28      4       1252    1302    1404    1404    YV      2804    N987HA  72      62      54      0       -10     PHX     FLG     119     4       14      0               0       N
A       NA      NA      NA      NA      N
2008    2       28      4       938     948     1051    1051    YV      2839    N987HA  73      63      48      0       -10     PHX     FLG     119     5       20      0               0       N
A       NA      NA      NA      NA      N
2008    2       28      4       1600    1610    1700    1709    YV      2855    N449YV  60      59      33      -9      -10     PHX     FLG     119     10      17      0               0       N
A       NA      NA      NA      NA      N
2008    2       28      4       1930    1934    2025    2032    YV      2857    N449YV  55      58      32      -7      -4      PHX     FLG     119     10      13      0               0       N
A       NA      NA      NA      NA      N
2008    2       28      4       2340    2257    36      2353    YV      2859    N805LR  56      56      29      43      43      PHX     FLG     119     9       18      0               0       4
3       0       0       0       0       Y
2008    2       28      4       950     954     1120    1129    YV      2903    N7291Z  90      95      60      -9      -4      PHX     GJT     438     5       25      0               0       N
A       NA      NA      NA      NA      N
2008    2       28      4       1550    1600    1748    1758    YV      2905    N991HA  118     118     92      -10     -10     PHX     GJT     438     6       20      0               0       N
A       NA      NA      NA      NA      N
2008    2       28      4       2010    1944    2200    2146    YV      2907    N437YV  110     122     95      14      26      PHX     GJT     438     5       10      0               0       N
A       NA      NA      NA      NA      Y
2008    2       28      4       1600    1601    1942    1930    YV      2884    N932LR  162     149     132     12      -1      PHX     IAH     1009    17      13      0               0       N
A       NA      NA      NA      NA      Y
2008    2       28      4       2255    2255    225     223     YV      2885    N903FJ  150     148     126     2       0       PHX     IAH     1009    8       16      0               0       N
A       NA      NA      NA      NA      Y
2008    2       28      4       1112    1115    1427    1437    YV      2923    N903FJ  135     142     107     -10     -3      PHX     ICT     870     7       21      0               0       N
A       NA      NA      NA      NA      N
2008    2       28      4       1943    1949    2253    2308    YV      2925    N7305V  130     139     110     -15     -6      PHX     ICT     870     7       13      0               0       N
A       NA      NA      NA      NA      N
2008    2       28      4       745     748     752     756     YV      2796    N27191  67      68      46      -4      -3      PHX     LAS     256     6       15      0               0       N
A       NA      NA      NA      NA      N
2008    2       28      4       1600    1600    1615    1625    YV      2708    N919FJ  75      85      55      -10     0       PHX     LAX     370     5       15      0               0       N
A       NA      NA      NA      NA      N
2008    2       28      4       830     821     851     850     YV      2728    N7305V  81      89      61      1       9       PHX     LGB     355     4       16      0               0       N
A       NA      NA      NA      NA      Y
2008    2       28      4       1145    1125    1200    1155    YV      2730    N927LR  75      90      62      5       20      PHX     LGB     355     6       7       0               0       N
A       NA      NA      NA      NA      Y
2008    2       28      4       1445    1447    1458    1507    YV      2732    N939LR  73      80      53      -9      -2      PHX     LGB     355     3       17      0               0       N
A       NA      NA      NA      NA      N
2008    2       28      4       1815    1746    1835    1805    YV      2734    N931LR  80      79      55      30      29      PHX     LGB     355     5       20      0               0       3
0       0       0       0       0       Y
2008    2       28      4       2108    2115    2135    2139    YV      2739    N928LR  87      84      56      -4      -7      PHX     LGB     355     11      20      0               0       N
A       NA      NA      NA      NA      N
```

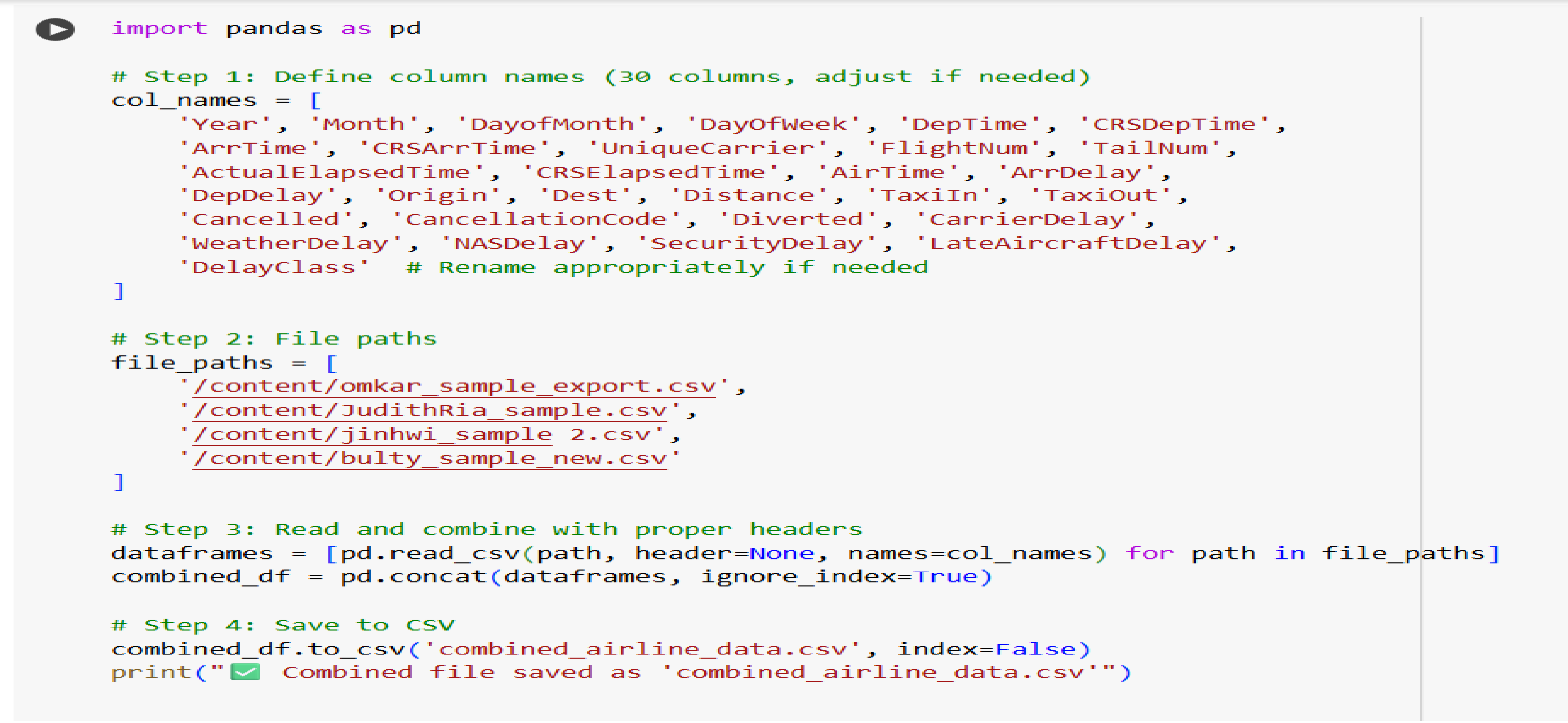
Step 8: -

Downloading the new table in newly created file

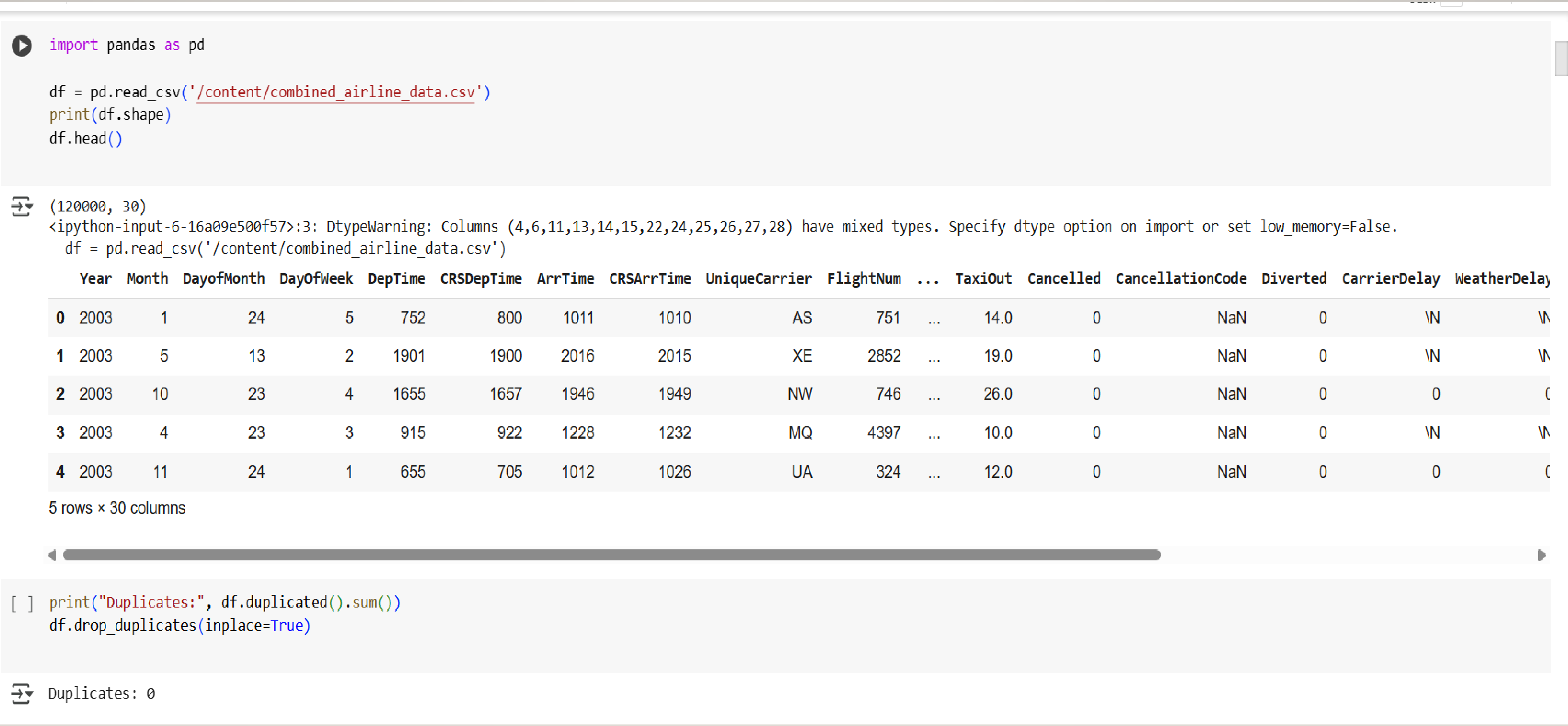


```
C:\Users\Hp>"C:\Users\Hp\Downloads\pscp.exe" -i "C:\Users\Hp\Downloads\scalable databases_5.ppk" hadoop@18.216.150.226:/
home/hadoop/bulty_sample.csv "C:\Users\Hp\Downloads\bulty_sample_new.csv"
bulty_sample_new.csv      | 2918 kB | 2918.8 kB/s | ETA: 00:00:00 | 100%
```

Data Pre-Processing



```python
import pandas as pd

# Step 1: Define column names (30 columns, adjust if needed)
col_names = [
    'Year', 'Month', 'DayofMonth', 'DayOfWeek', 'DepTime', 'CRSDepTime',
    'ArrTime', 'CRSArrTime', 'UniqueCarrier', 'FlightNum', 'TailNum',
    'ActualElapsedTime', 'CRSElapsedTime', 'AirTime', 'ArrDelay',
    'DepDelay', 'Origin', 'Dest', 'Distance', 'TaxiIn', 'TaxiOut',
    'Cancelled', 'CancellationCode', 'Diverted', 'CarrierDelay',
    'WeatherDelay', 'NASDelay', 'SecurityDelay', 'LateAircraftDelay',
    'DelayClass'  # Rename appropriately if needed
]

# Step 2: File paths
file_paths = [
    '/content/omkar_sample_export.csv',
    '/content/JudithRia_sample.csv',
    '/content/jinhwi_sample 2.csv',
    '/content/bulty_sample_new.csv'
]

# Step 3: Read and combine with proper headers
dataframes = [pd.read_csv(path, header=None, names=col_names) for path in file_paths]
combined_df = pd.concat(dataframes, ignore_index=True)

# Step 4: Save to CSV
combined_df.to_csv('combined_airline_data.csv', index=False)
print("✅ Combined file saved as 'combined_airline_data.csv'")
```



```
import pandas as pd

df = pd.read_csv('/content/combined_airline_data.csv')
print(df.shape)
df.head()
```

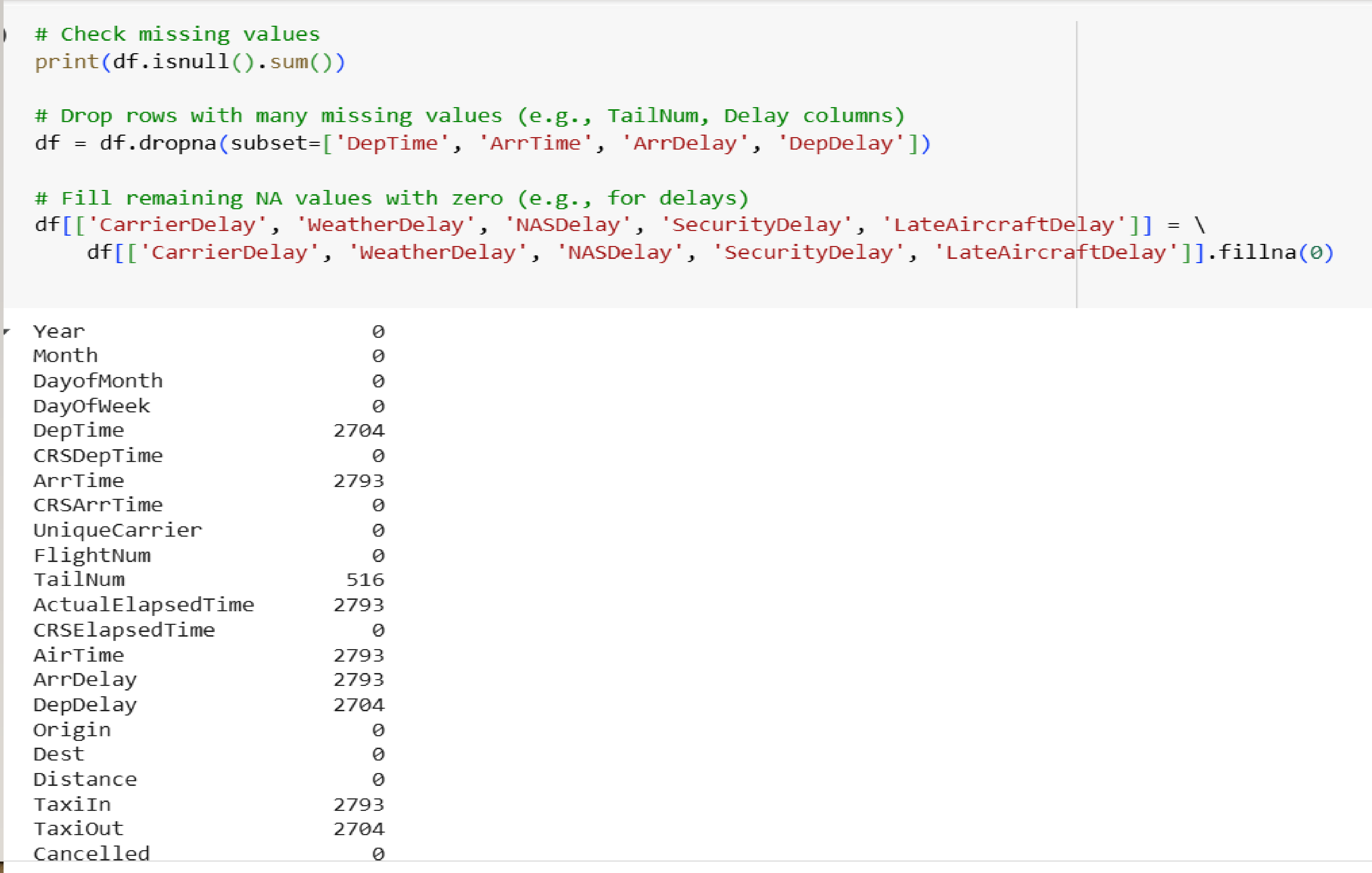
```
(120000, 30)
<ipython-input-6-16a09e500f57>:3: DtypeWarning: Columns (4,6,11,13,14,15,22,24,25,26,27,28) have mixed types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv('/content/combined_airline_data.csv')
```

	Year	Month	DayofMonth	DayOfWeek	DepTime	CRSDepTime	ArrTime	CRSArrTime	UniqueCarrier	FlightNum	...	TaxiOut	Cancelled	CancellationCode	Diverted	CarrierDelay	WeatherDelay
0	2003	1	24	5	752	800	1011	1010	AS	751	...	14.0	0	NaN	0	\N	\N
1	2003	5	13	2	1901	1900	2016	2015	XE	2852	...	19.0	0	NaN	0	\N	\N
2	2003	10	23	4	1655	1657	1946	1949	NW	746	...	26.0	0	NaN	0	0	0
3	2003	4	23	3	915	922	1228	1232	MQ	4397	...	10.0	0	NaN	0	\N	\N
4	2003	11	24	1	655	705	1012	1026	UA	324	...	12.0	0	NaN	0	0	0

5 rows × 30 columns

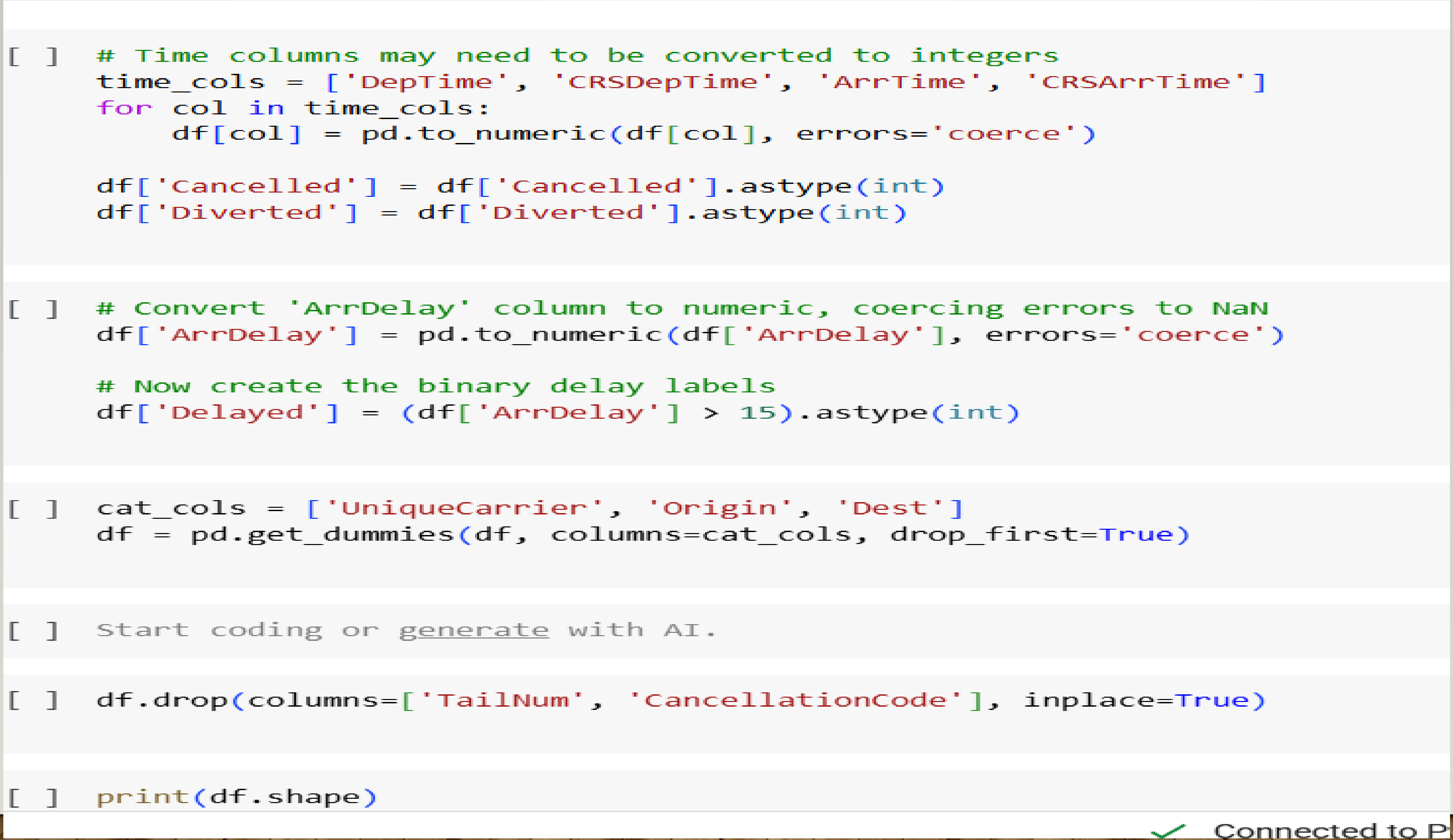
```
print("Duplicates:", df.duplicated().sum())
df.drop_duplicates(inplace=True)
```

```
Duplicates: 0
```



```python
# Check missing values
print(df.isnull().sum())

# Drop rows with many missing values (e.g., TailNum, Delay columns)
df = df.dropna(subset=['DepTime', 'ArrTime', 'ArrDelay', 'DepDelay'])

# Fill remaining NA values with zero (e.g., for delays)
df[['CarrierDelay', 'WeatherDelay', 'NASDelay', 'SecurityDelay', 'LateAircraftDelay']] = \
    df[['CarrierDelay', 'WeatherDelay', 'NASDelay', 'SecurityDelay', 'LateAircraftDelay']].fillna(0)
```

```
Year                    0
Month                   0
DayofMonth              0
DayOfWeek               0
DepTime              2704
CRSDepTime              0
ArrTime              2793
CRSArrTime              0
UniqueCarrier           0
FlightNum               0
TailNum               516
ActualElapsedTime    2793
CRSElapsedTime          0
AirTime              2793
ArrDelay             2793
DepDelay             2704
Origin                  0
Dest                    0
Distance                0
TaxiIn               2793
TaxiOut              2704
Cancelled               0
```



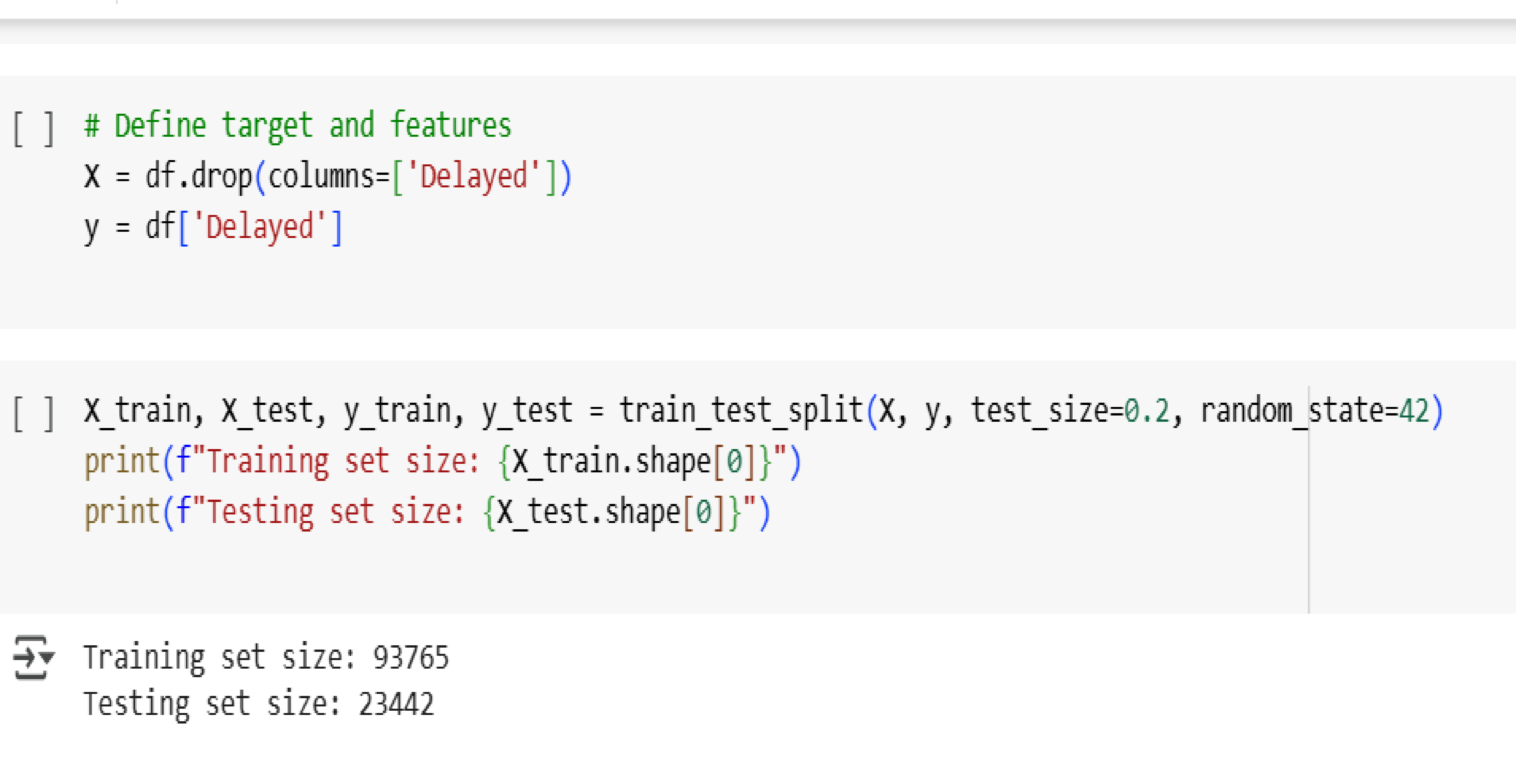
```
# Time columns may need to be converted to integers
time_cols = ['DepTime', 'CRSDepTime', 'ArrTime', 'CRSArrTime']
for col in time_cols:
    df[col] = pd.to_numeric(df[col], errors='coerce')

df['Cancelled'] = df['Cancelled'].astype(int)
df['Diverted'] = df['Diverted'].astype(int)
```

```
# Convert 'ArrDelay' column to numeric, coercing errors to NaN
df['ArrDelay'] = pd.to_numeric(df['ArrDelay'], errors='coerce')

# Now create the binary delay labels
df['Delayed'] = (df['ArrDelay'] > 15).astype(int)
```

```
cat_cols = ['UniqueCarrier', 'Origin', 'Dest']
df = pd.get_dummies(df, columns=cat_cols, drop_first=True)
```

```
df.drop(columns=['TailNum', 'CancellationCode'], inplace=True)
```

```
print(df.shape)
```

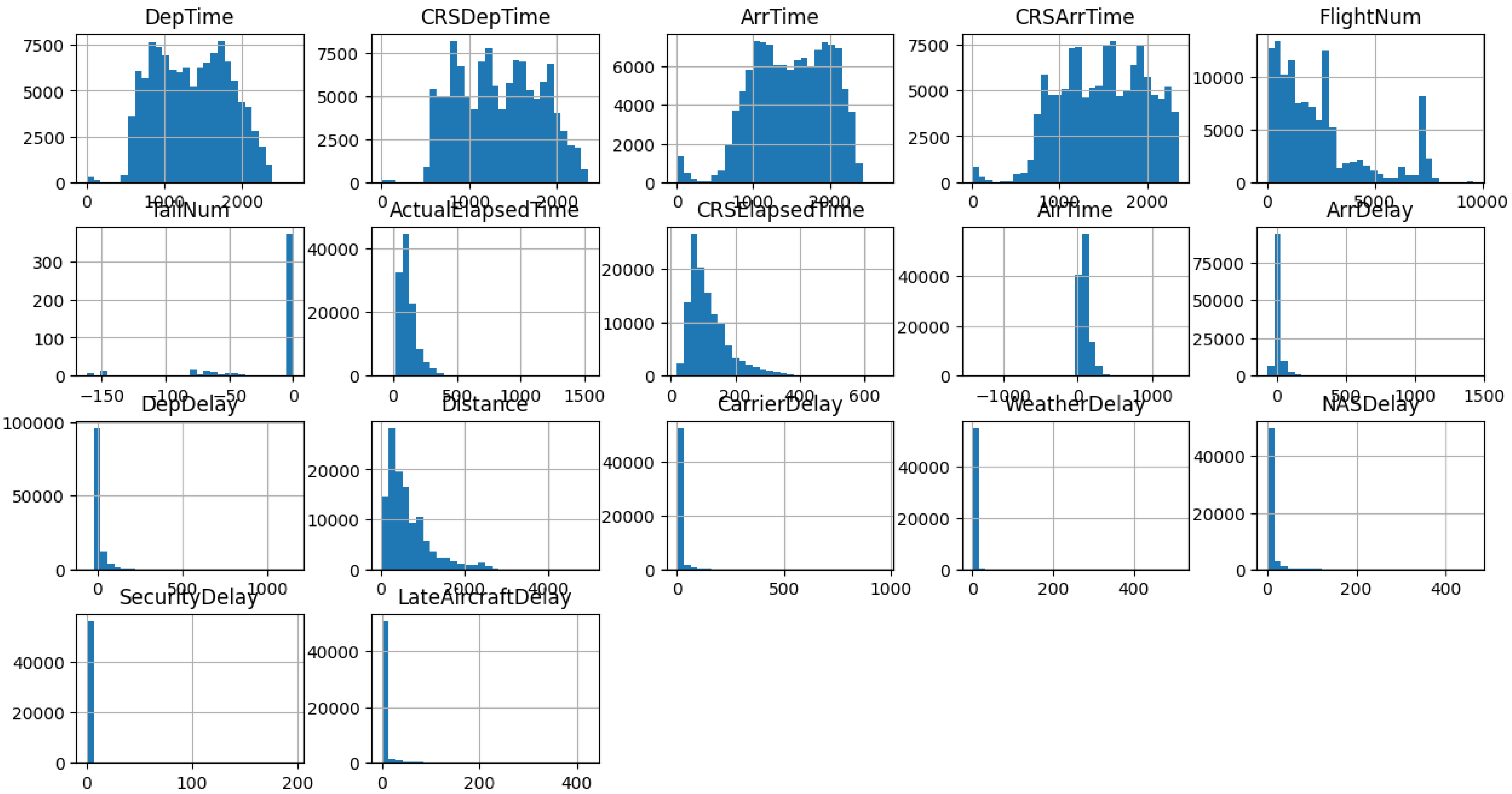


```
# Define target and features
X = df.drop(columns=['Delayed'])
y = df['Delayed']
```

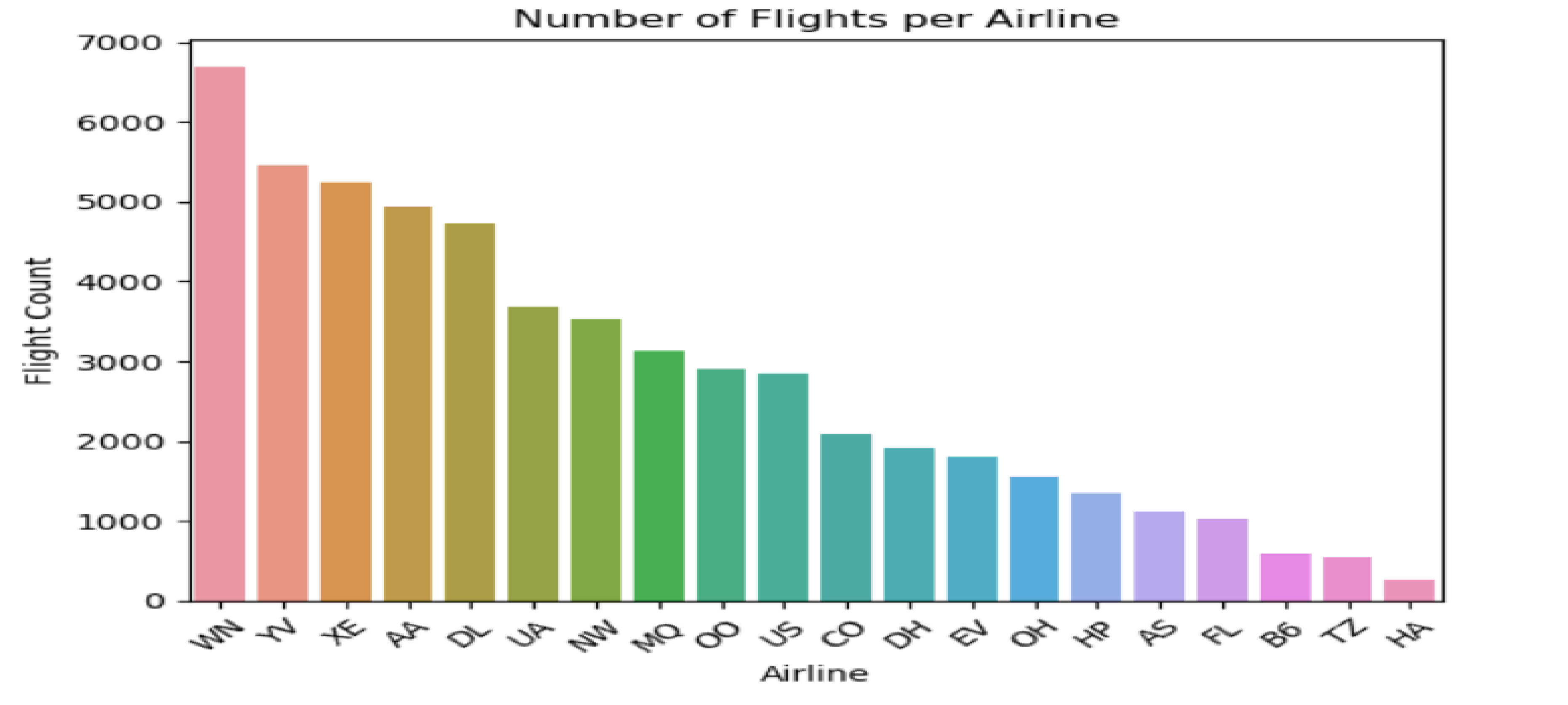
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
print(f"Training set size: {X_train.shape[0]}")
print(f"Testing set size: {X_test.shape[0]}")
```

```
Training set size: 93765
Testing set size: 23442
```

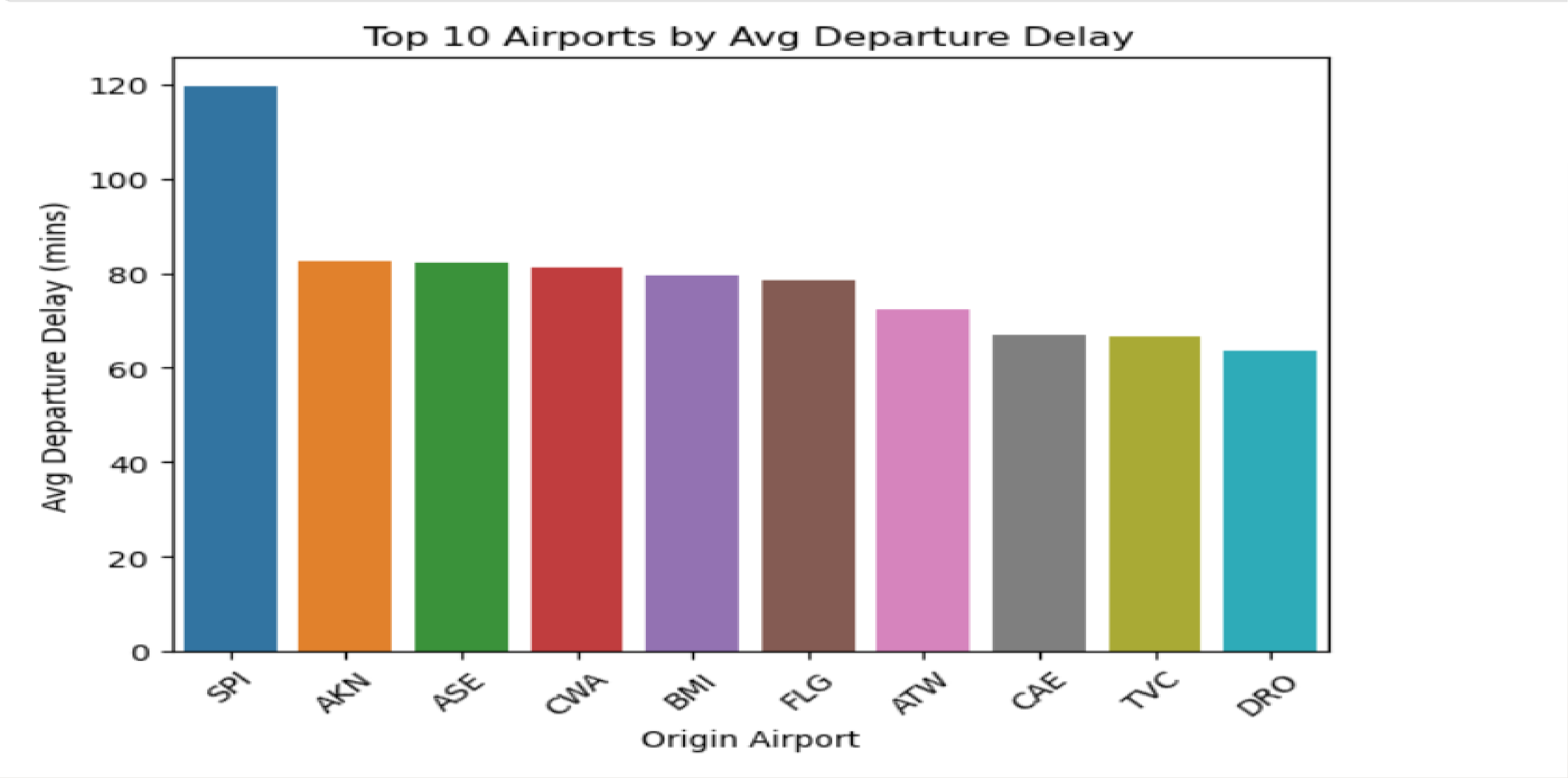
Data Scaling



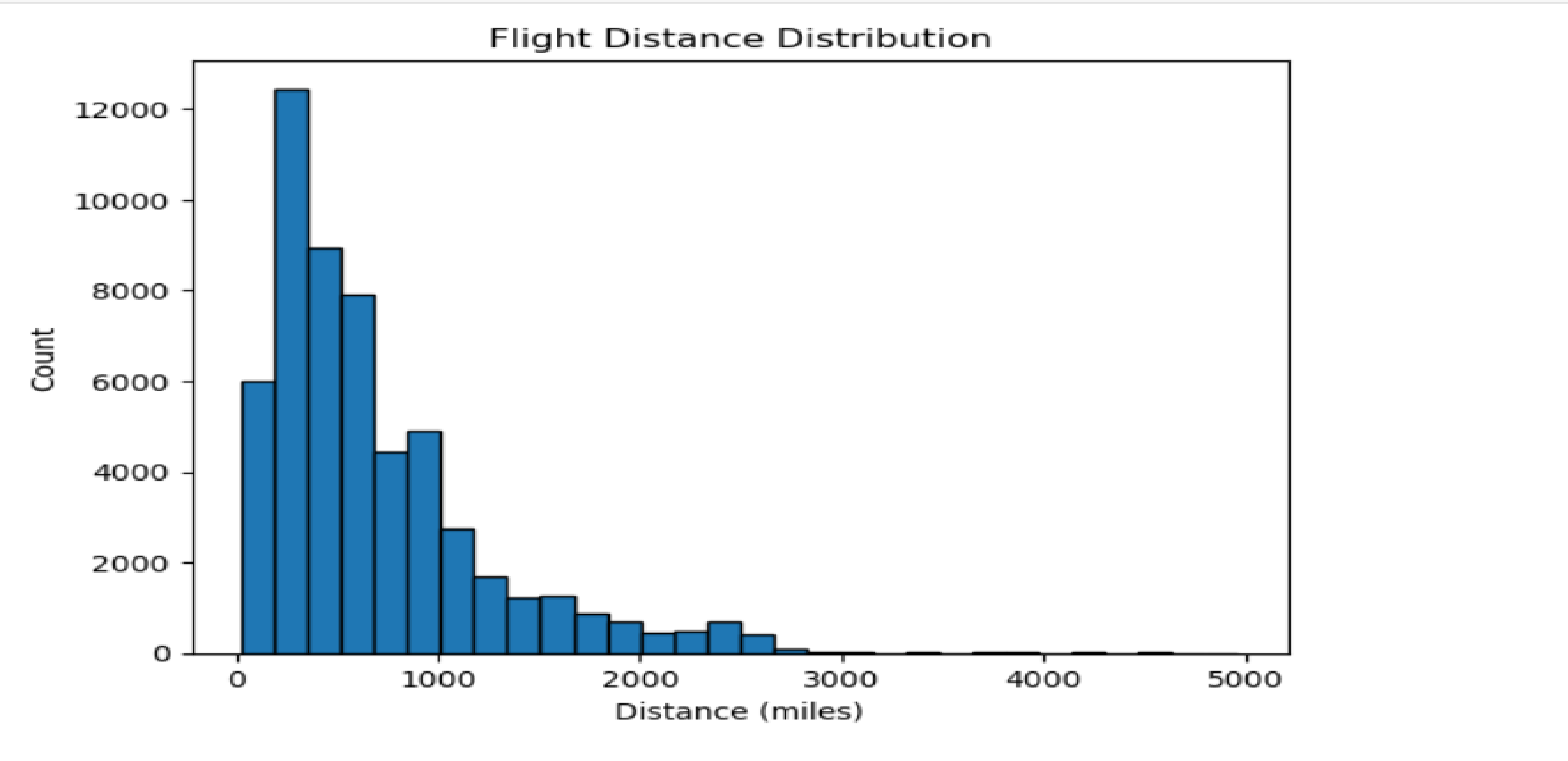
Number of Flights per Airline: -



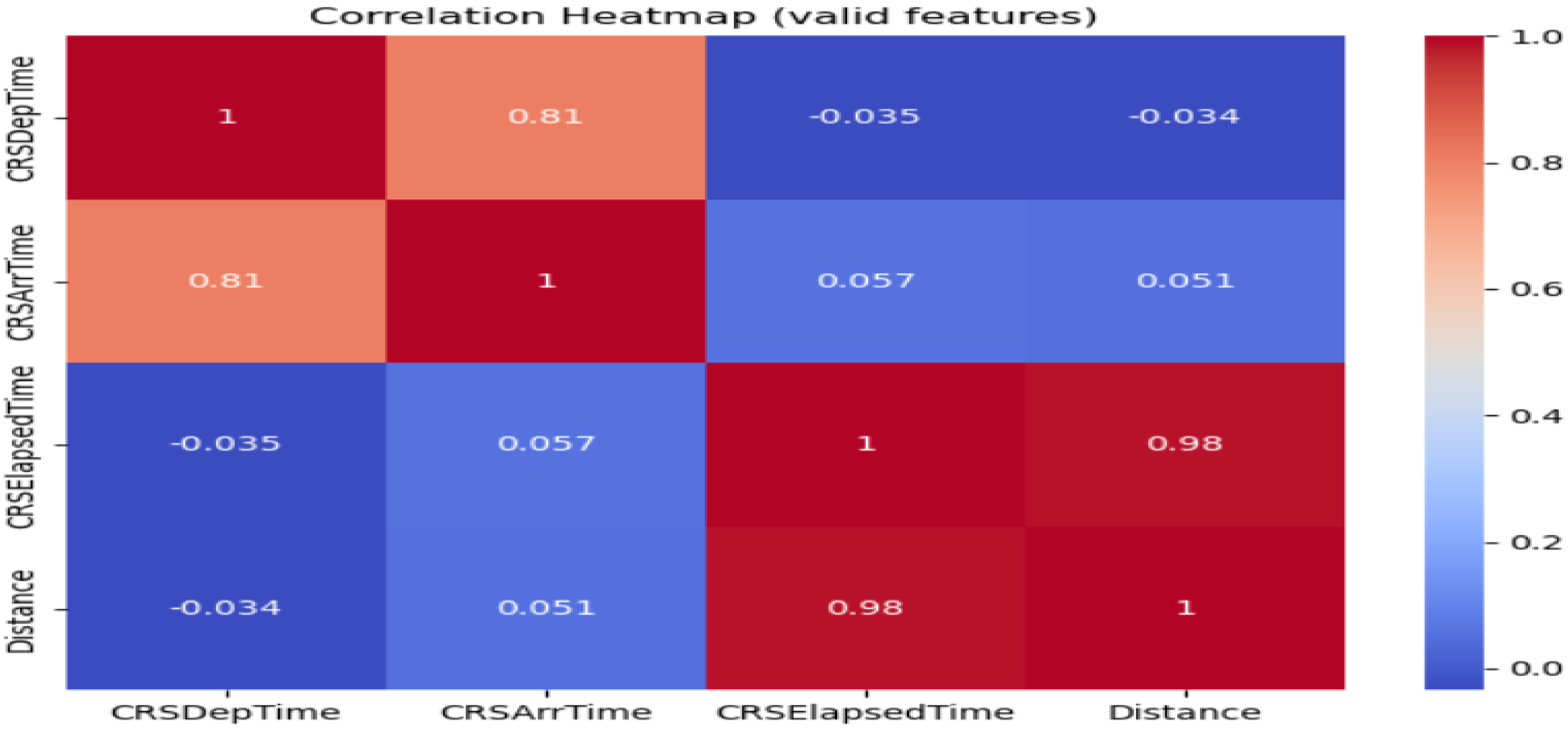
Top 10 Airports by Avg Departure Delay: -



Flight Distance Distribution: -

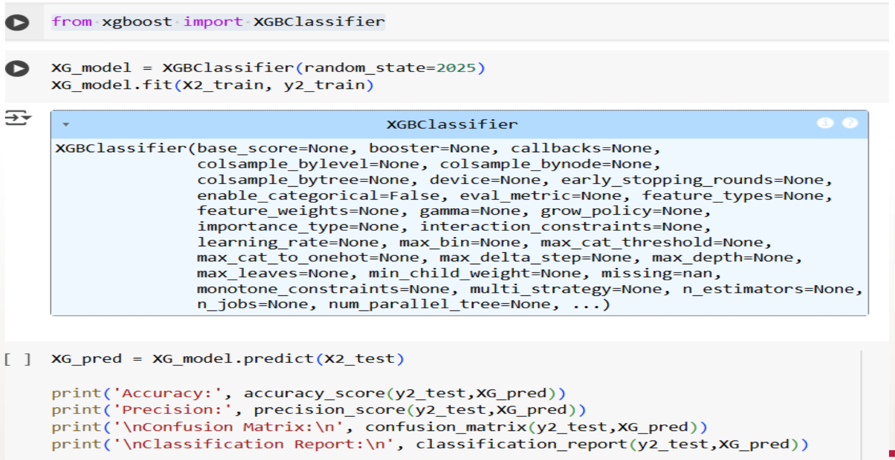


Correlation Plot: -



Model Training: -

XGBOOST



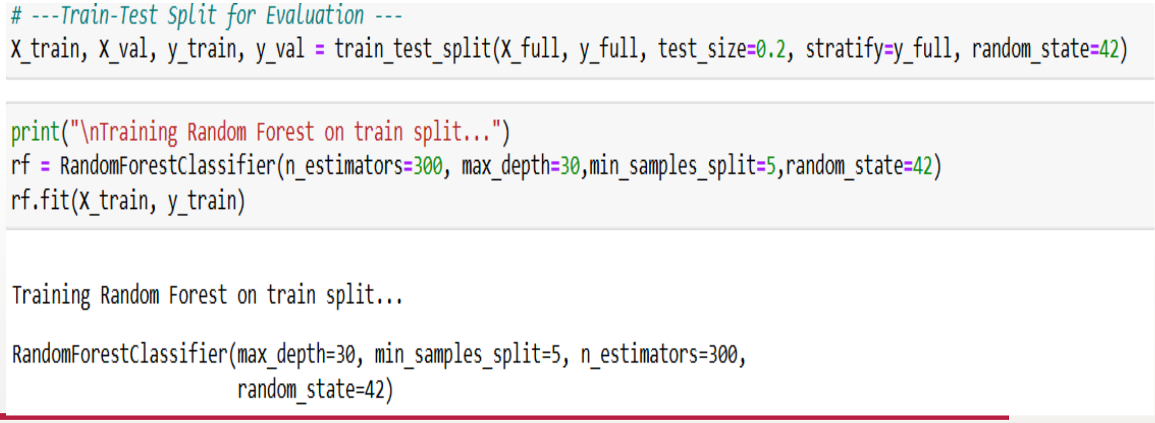
```
from xgboost import XGBClassifier
```

```
XG_model = XGBClassifier(random_state=2025)
XG_model.fit(X2_train, y2_train)
```

```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              feature_weights=None, gamma=None, grow_policy=None,
              importance_type=None, interaction_constraints=None,
              learning_rate=None, max_bin=None, max_cat_threshold=None,
              max_cat_to_onehot=None, max_delta_step=None, max_depth=None,
              max_leaves=None, min_child_weight=None, missing=nan,
              monotone_constraints=None, multi_strategy=None, n_estimators=None,
              n_jobs=None, num_parallel_tree=None, ...)
```

```
[ ] XG_pred = XG_model.predict(X2_test)

    print('Accuracy:', accuracy_score(y2_test,XG_pred))
    print('Precision:', precision_score(y2_test,XG_pred))
    print('\nConfusion Matrix:\n', confusion_matrix(y2_test,XG_pred))
    print('\nClassification Report:\n', classification_report(y2_test,XG_pred))
```

Random Forest

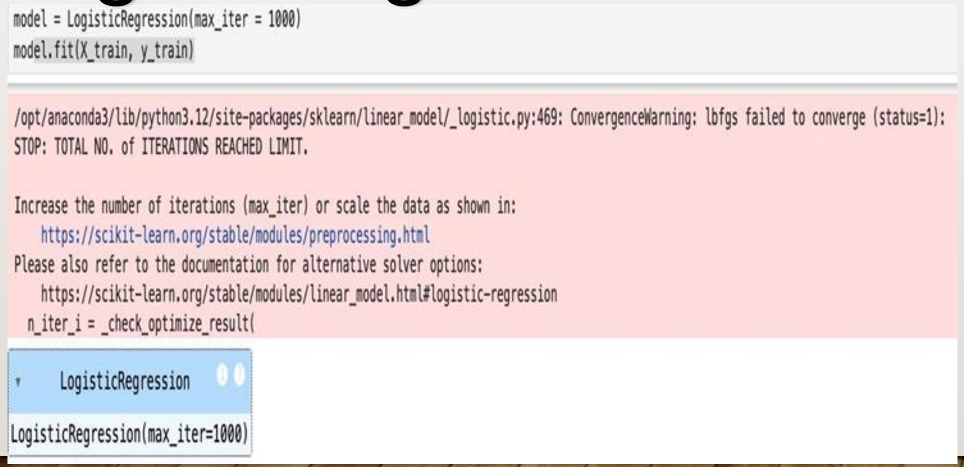


```
# ---Train-Test Split for Evaluation ---
X_train, X_val, y_train, y_val = train_test_split(X_full, y_full, test_size=0.2, stratify=y_full, random_state=42)
```

```
print("\nTraining Random Forest on train split...")
rf = RandomForestClassifier(n_estimators=300, max_depth=30,min_samples_split=5,random_state=42)
rf.fit(X_train, y_train)
```

```
Training Random Forest on train split...

RandomForestClassifier(max_depth=30, min_samples_split=5, n_estimators=300,
                       random_state=42)
```

Logistic Regression

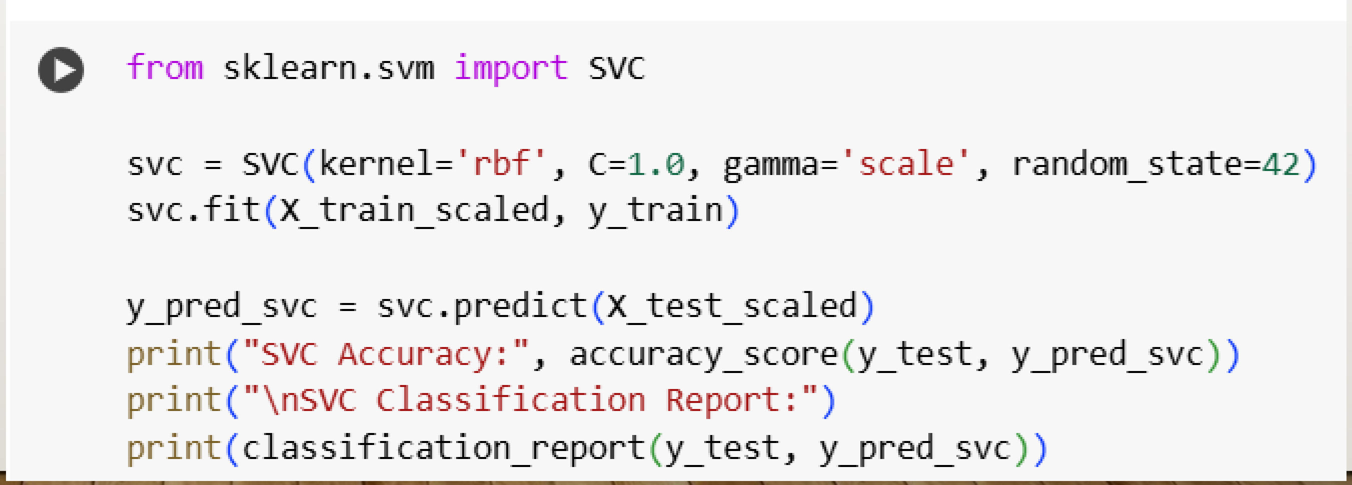


```
model = LogisticRegression(max_iter = 1000)
model.fit(X_train, y_train)
```

```
/opt/anaconda3/lib/python3.12/site-packages/sklearn/linear_model/_logistic.py:469: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
LogisticRegression(max_iter=1000)
```

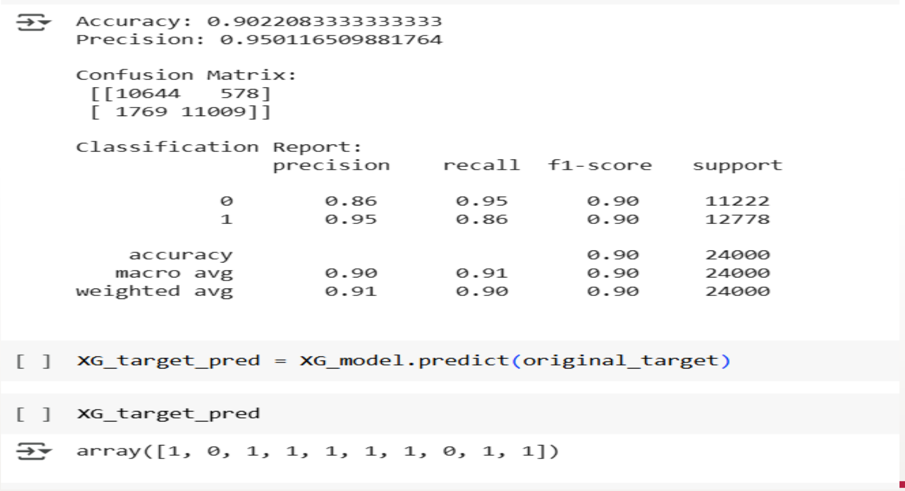
Support Vector Classifier



```
from sklearn.svm import SVC

svc = SVC(kernel='rbf', C=1.0, gamma='scale', random_state=42)
svc.fit(X_train_scaled, y_train)

y_pred_svc = svc.predict(X_test_scaled)
print("SVC Accuracy:", accuracy_score(y_test, y_pred_svc))
print("\nSVC Classification Report:")
print(classification_report(y_test, y_pred_svc))
```

Model Evaluation

XGBOOST



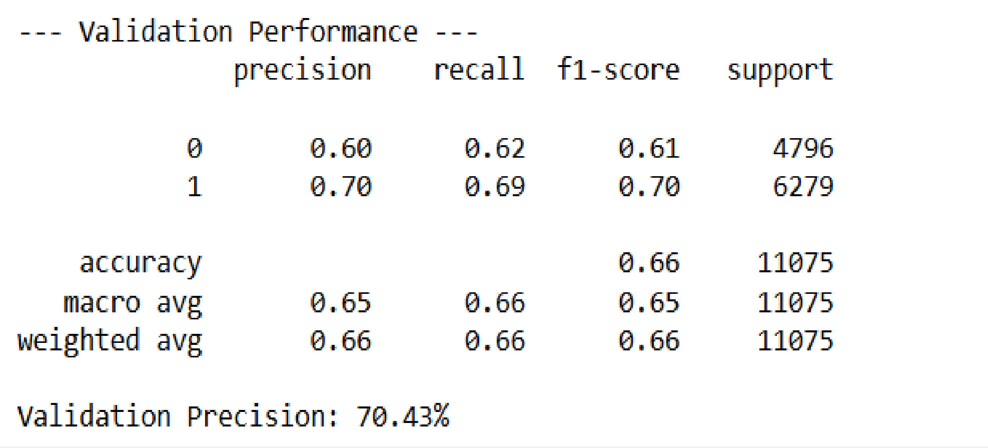
```
Accuracy: 0.9022083333333333
Precision: 0.950116509881764

Confusion Matrix:
 [[10644   578]
 [ 1769 11009]]

Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.95      0.90     11222
           1       0.95      0.86      0.90     12778

    accuracy                           0.90     24000
   macro avg       0.90      0.91      0.90     24000
weighted avg       0.91      0.90      0.90     24000
```

```
[ ] XG_target_pred = XG_model.predict(original_target)
```

```
[ ] XG_target_pred
```

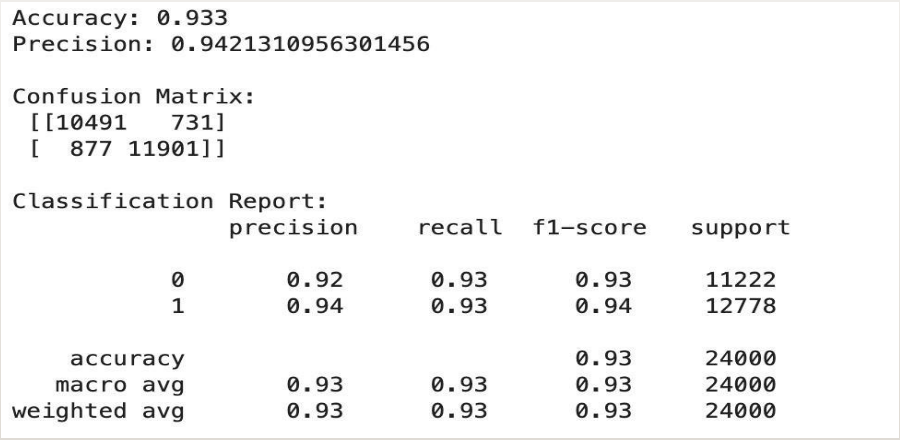
```
array([1, 0, 1, 1, 1, 1, 1, 0, 1, 1])
```

Random Forest



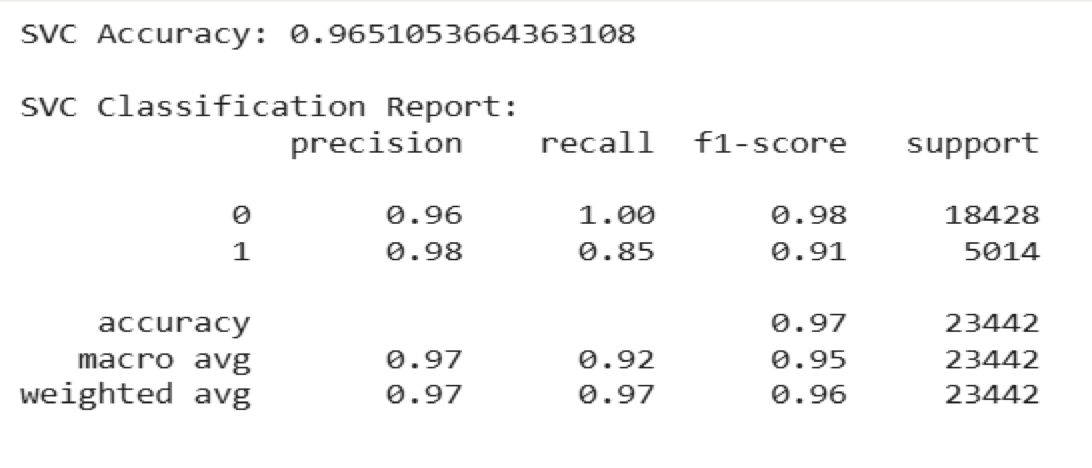
```
--- Validation Performance ---
              precision    recall  f1-score   support

           0       0.60      0.62      0.61      4796
           1       0.70      0.69      0.70      6279

    accuracy                           0.66     11075
   macro avg       0.65      0.66      0.65     11075
weighted avg       0.66      0.66      0.66     11075

Validation Precision: 70.43%
```

Logistic Regression



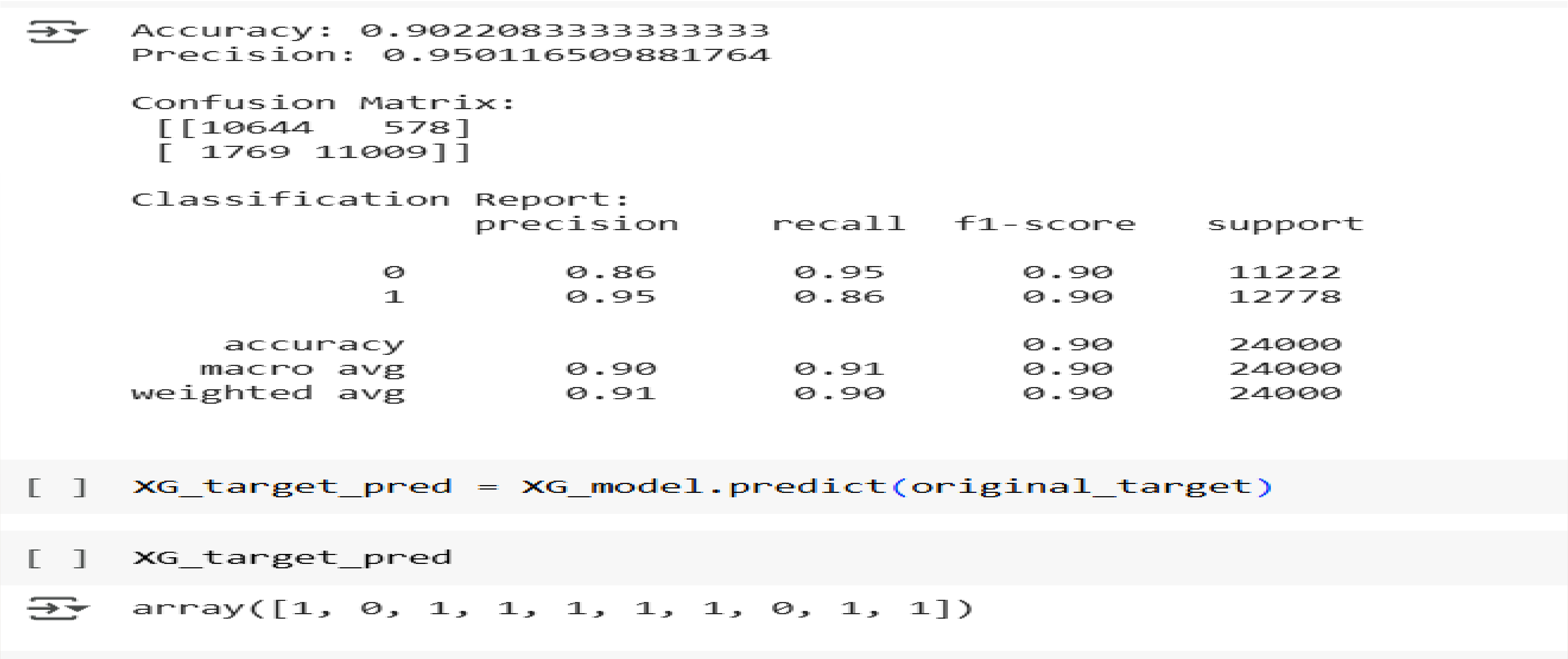
```
Accuracy: 0.933
Precision: 0.9421310956301456

Confusion Matrix:
 [[10491   731]
 [  877 11901]]

Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.93      0.93     11222
           1       0.94      0.93      0.94     12778

    accuracy                           0.93     24000
   macro avg       0.93      0.93      0.93     24000
weighted avg       0.93      0.93      0.93     24000
```

Support vector classifier



```
SVC Accuracy: 0.9651053664363108

SVC Classification Report:
              precision    recall  f1-score   support

           0       0.96      1.00      0.98     18428
           1       0.98      0.85      0.91      5014

    accuracy                           0.97     23442
   macro avg       0.97      0.92      0.95     23442
weighted avg       0.97      0.97      0.96     23442
```

Model Training: - Output



```
Accuracy: 0.9022083333333333
Precision: 0.950116509881764

Confusion Matrix:
 [[10644   578]
 [ 1769 11009]]

Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.95      0.90     11222
           1       0.95      0.86      0.90     12778

    accuracy                           0.90     24000
   macro avg       0.90      0.91      0.90     24000
weighted avg       0.91      0.90      0.90     24000
```

```
[ ]  XG_target_pred = XG_model.predict(original_target)
```

```
[ ]  XG_target_pred
```

```
array([1, 0, 1, 1, 1, 1, 1, 0, 1, 1])
```

CONCLUSION:

THE MODEL WE DECIDED TO PROCEED WITH WAS XG BOOST AS IT WAS A GOOD FITTED MODEL. THE SVC MODEL DID PERFORM GOOD BUT IT WAS INCLINING MORE TOWARDS OVERFITTING SO ULTIMATELY WE FINALIZED TO GO WITH XG BOOST.