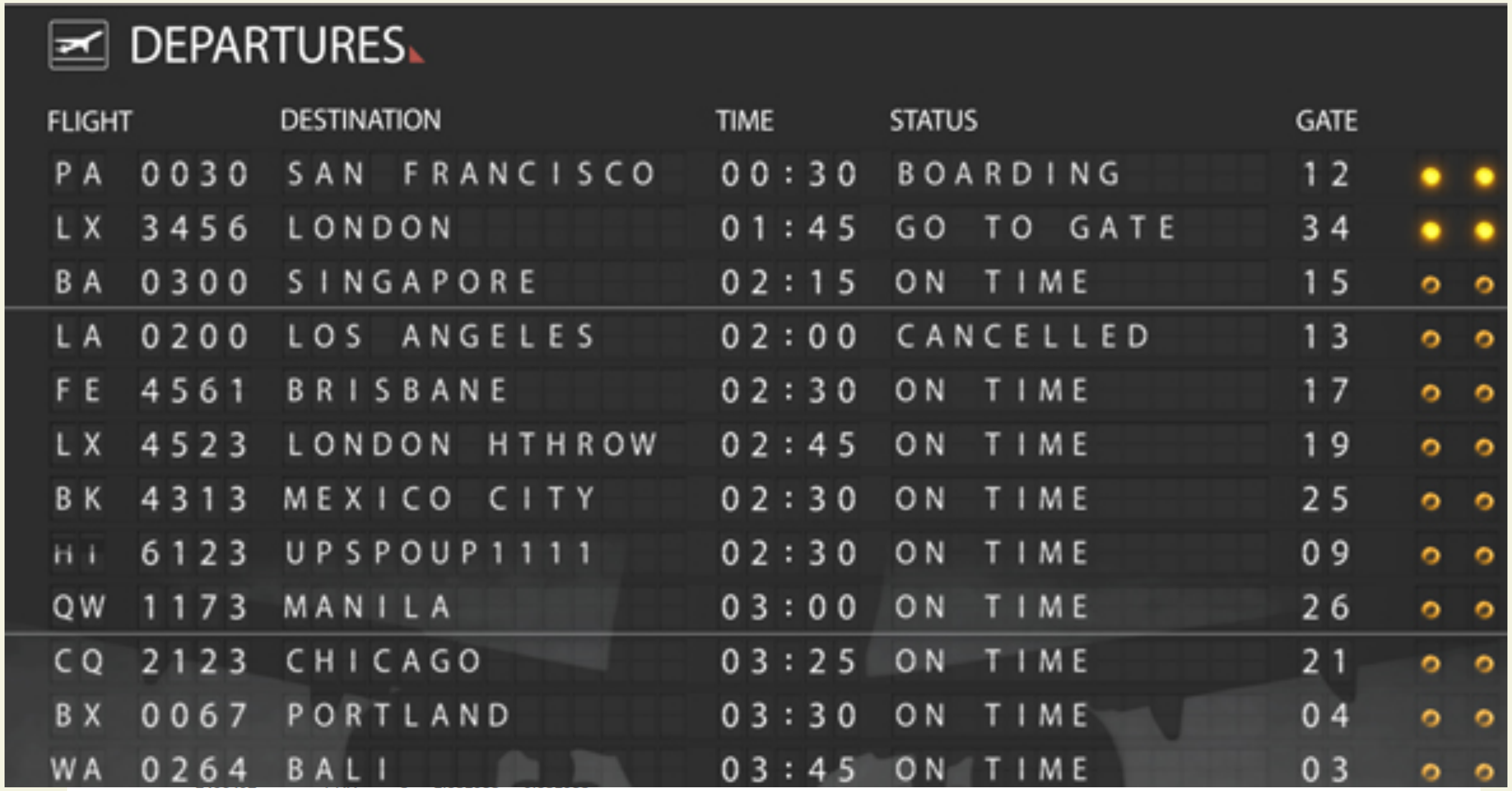
Predicting Flight Delays Using SparkML on AWS

Jason Carpenter | Christopher Csiszar | Nishan Madawanarachchi | Ryan Campa

After a long weekend in Vegas...

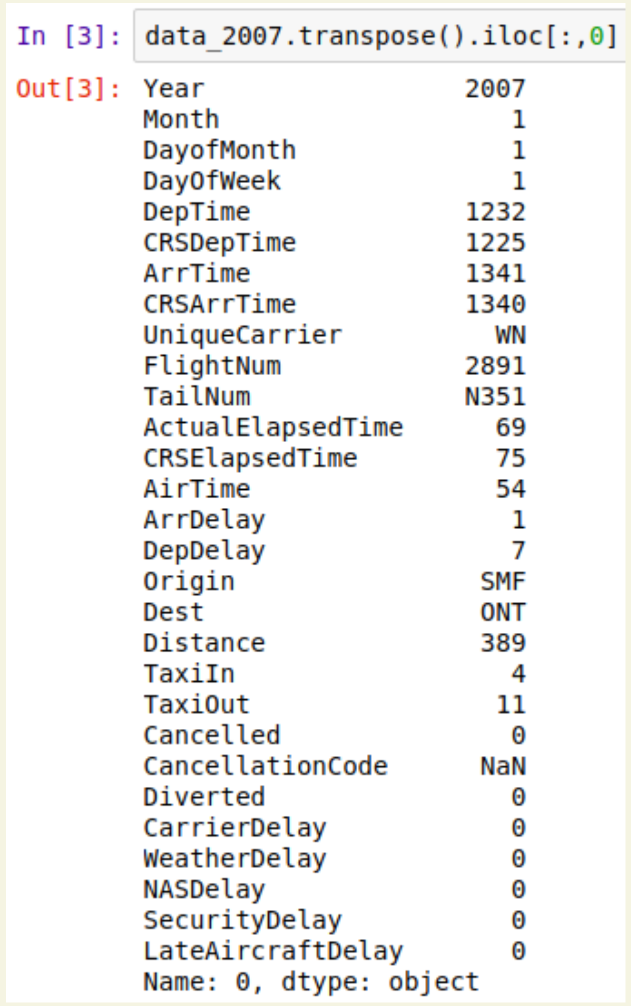


DEPARTURES

FLIGHT		DESTINATION	TIME	STATUS	GATE
PA	0030	SAN FRANCISCO	00:30	BOARDING	12
LX	3456	LONDON	01:45	GO TO GATE	34
BA	0300	SINGAPORE	02:15	ON TIME	15
LA	0200	LOS ANGELES	02:00	CANCELLED	13
FE	4561	BRISBANE	02:30	ON TIME	17
LX	4523	LONDON HTHROW	02:45	ON TIME	19
BK	4313	MEXICO CITY	02:30	ON TIME	25
HI	6123	UPSPOUP1111	02:30	ON TIME	09
QW	1173	MANILA	03:00	ON TIME	26
CQ	2123	CHICAGO	03:25	ON TIME	21
BX	0067	PORTLAND	03:30	ON TIME	04
WA	0264	BALI	03:45	ON TIME	03

Data

- DOT's Bureau of Transportation and Statistics
- Tracks on-time performance of domestic flights operated by large air carriers.
- Early, On time, delayed, diverted, cancelled
- 2006-2008
- CSV, ~2Gb, 21 million rows, 29 columns
- http://stat-computing.org/dataexpo/2009/the-data.html

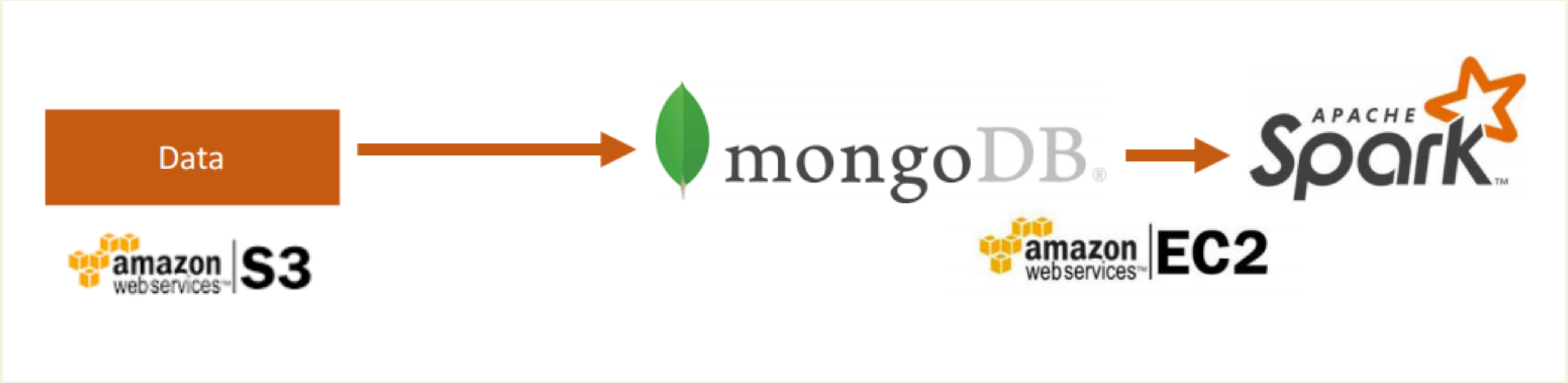


```
In [3]: data_2007.transpose().iloc[:,0]

Out[3]: Year                 2007
        Month                   1
        DayofMonth              1
        DayOfWeek               1
        DepTime              1232
        CRSDepTime           1225
        ArrTime              1341
        CRSArrTime           1340
        UniqueCarrier          WN
        FlightNum            2891
        TailNum              N351
        ActualElapsedTime      69
        CRSElapsedTime         75
        AirTime                54
        ArrDelay                1
        DepDelay                7
        Origin                SMF
        Dest                  ONT
        Distance              389
        TaxiIn                  4
        TaxiOut                11
        Cancelled               0
        CancellationCode      NaN
        Diverted                0
        CarrierDelay            0
        WeatherDelay            0
        NASDelay                0
        SecurityDelay           0
        LateAircraftDelay       0
        Name: 0, dtype: object
```

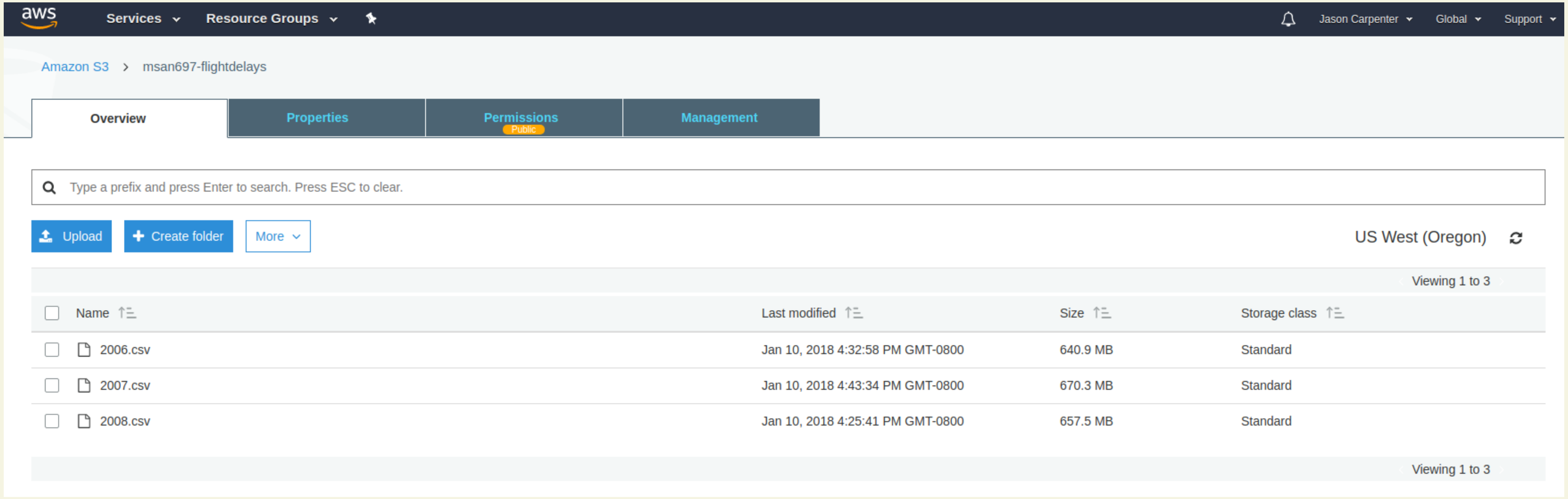
Objective

- Build Machine Learning Pipeline on AWS using Apache Spark and MongoDB
- Predict Arrival Delays
- Evaluate Performance of Model

Pipeline


Data
amazon web services S3
mongoDB
APACHE Spark
amazon web services EC2

Data Storage



Amazon S3 > msan697-flightdelays

US West (Oregon)

Viewing 1 to 3

Name	Last modified	Size	Storage class
2006.csv	Jan 10, 2018 4:32:58 PM GMT-0800	640.9 MB	Standard
2007.csv	Jan 10, 2018 4:43:34 PM GMT-0800	670.3 MB	Standard
2008.csv	Jan 10, 2018 4:25:41 PM GMT-0800	657.5 MB	Standard

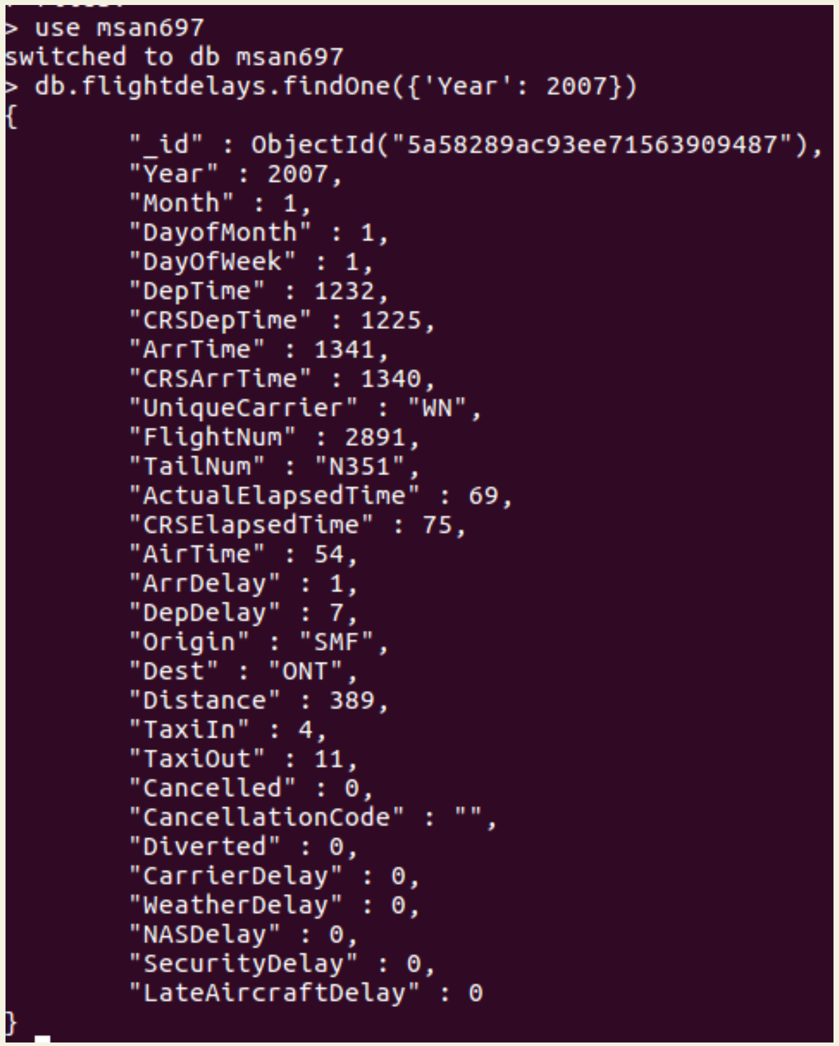
Viewing 1 to 3

AWS Configuration

- **m3.2xlarge** spark-ec2 stand-alone cluster with 1 master and 2 workers

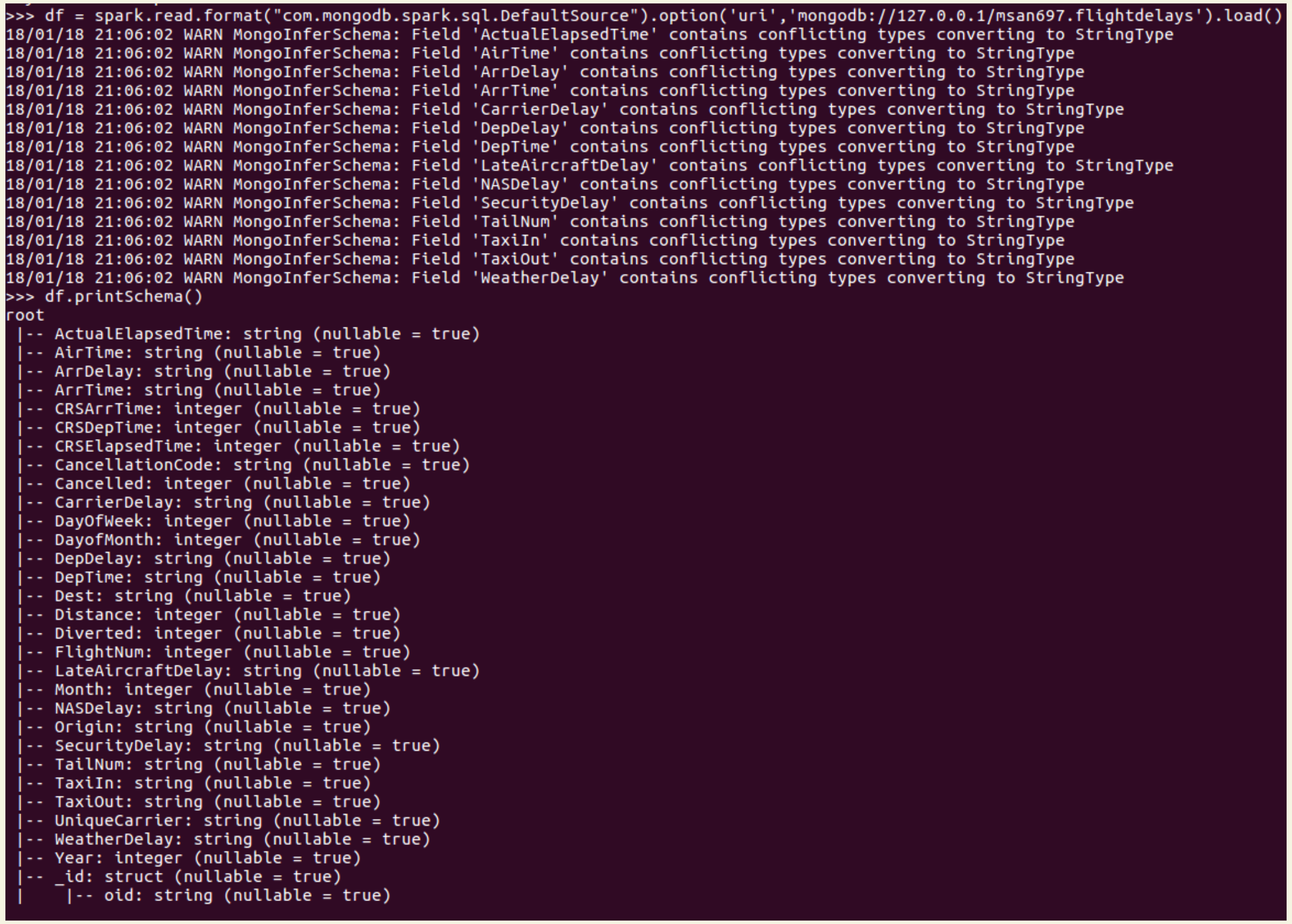
VCPU	ECU	Memory(GiB)	Instance Storage (GB)
8	26	30	2 x 80 (SSD)

MongoDB



```
> use msan697
switched to db msan697
> db.flightdelays.findOne({'Year': 2007})
{
        "_id" : ObjectId("5a58289ac93ee71563909487"),
        "Year" : 2007,
        "Month" : 1,
        "DayofMonth" : 1,
        "DayOfWeek" : 1,
        "DepTime" : 1232,
        "CRSDepTime" : 1225,
        "ArrTime" : 1341,
        "CRSArrTime" : 1340,
        "UniqueCarrier" : "WN",
        "FlightNum" : 2891,
        "TailNum" : "N351",
        "ActualElapsedTime" : 69,
        "CRSElapsedTime" : 75,
        "AirTime" : 54,
        "ArrDelay" : 1,
        "DepDelay" : 7,
        "Origin" : "SMF",
        "Dest" : "ONT",
        "Distance" : 389,
        "TaxiIn" : 4,
        "TaxiOut" : 11,
        "Cancelled" : 0,
        "CancellationCode" : "",
        "Diverted" : 0,
        "CarrierDelay" : 0,
        "WeatherDelay" : 0,
        "NASDelay" : 0,
        "SecurityDelay" : 0,
        "LateAircraftDelay" : 0
}
```

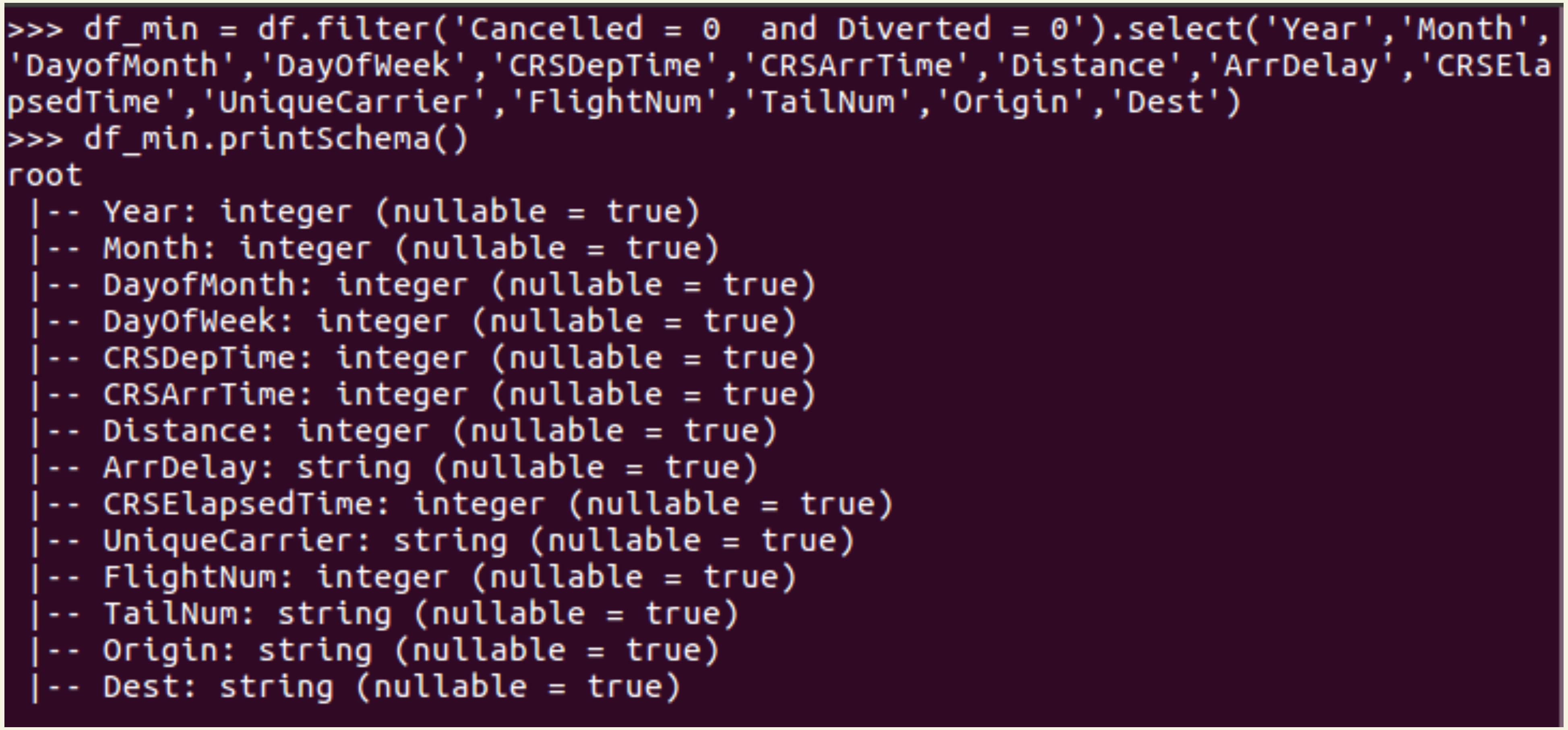
SparkSQL



```
>>> df = spark.read.format("com.mongodb.spark.sql.DefaultSource").option('uri','mongodb://127.0.0.1/msan697.flightdelays').load()
18/01/18 21:06:02 WARN MongoInferSchema: Field 'ActualElapsedTime' contains conflicting types converting to StringType
18/01/18 21:06:02 WARN MongoInferSchema: Field 'AirTime' contains conflicting types converting to StringType
18/01/18 21:06:02 WARN MongoInferSchema: Field 'ArrDelay' contains conflicting types converting to StringType
18/01/18 21:06:02 WARN MongoInferSchema: Field 'ArrTime' contains conflicting types converting to StringType
18/01/18 21:06:02 WARN MongoInferSchema: Field 'CarrierDelay' contains conflicting types converting to StringType
18/01/18 21:06:02 WARN MongoInferSchema: Field 'DepDelay' contains conflicting types converting to StringType
18/01/18 21:06:02 WARN MongoInferSchema: Field 'DepTime' contains conflicting types converting to StringType
18/01/18 21:06:02 WARN MongoInferSchema: Field 'LateAircraftDelay' contains conflicting types converting to StringType
18/01/18 21:06:02 WARN MongoInferSchema: Field 'NASDelay' contains conflicting types converting to StringType
18/01/18 21:06:02 WARN MongoInferSchema: Field 'SecurityDelay' contains conflicting types converting to StringType
18/01/18 21:06:02 WARN MongoInferSchema: Field 'TailNum' contains conflicting types converting to StringType
18/01/18 21:06:02 WARN MongoInferSchema: Field 'TaxiIn' contains conflicting types converting to StringType
18/01/18 21:06:02 WARN MongoInferSchema: Field 'TaxiOut' contains conflicting types converting to StringType
18/01/18 21:06:02 WARN MongoInferSchema: Field 'WeatherDelay' contains conflicting types converting to StringType
>>> df.printSchema()
root
 |-- ActualElapsedTime: string (nullable = true)
 |-- AirTime: string (nullable = true)
 |-- ArrDelay: string (nullable = true)
 |-- ArrTime: string (nullable = true)
 |-- CRSArrTime: integer (nullable = true)
 |-- CRSDepTime: integer (nullable = true)
 |-- CRSElapsedTime: integer (nullable = true)
 |-- CancellationCode: string (nullable = true)
 |-- Cancelled: integer (nullable = true)
 |-- CarrierDelay: string (nullable = true)
 |-- DayOfWeek: integer (nullable = true)
 |-- DayofMonth: integer (nullable = true)
 |-- DepDelay: string (nullable = true)
 |-- DepTime: string (nullable = true)
 |-- Dest: string (nullable = true)
 |-- Distance: integer (nullable = true)
 |-- Diverted: integer (nullable = true)
 |-- FlightNum: integer (nullable = true)
 |-- LateAircraftDelay: string (nullable = true)
 |-- Month: integer (nullable = true)
 |-- NASDelay: string (nullable = true)
 |-- Origin: string (nullable = true)
 |-- SecurityDelay: string (nullable = true)
 |-- TailNum: string (nullable = true)
 |-- TaxiIn: string (nullable = true)
 |-- TaxiOut: string (nullable = true)
 |-- UniqueCarrier: string (nullable = true)
 |-- WeatherDelay: string (nullable = true)
 |-- Year: integer (nullable = true)
 |-- _id: struct (nullable = true)
 |    |-- oid: string (nullable = true)
```

Creating Data Frame From MongoDB

SparkSQL

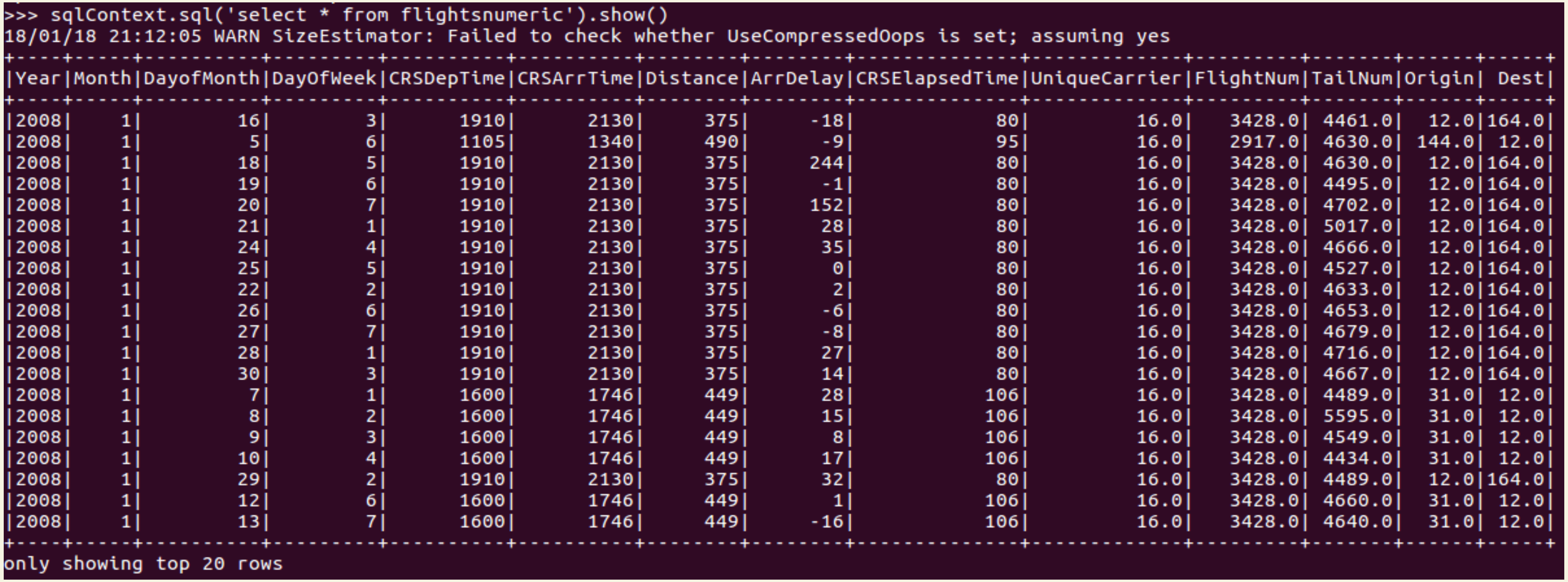


```
>>> df_min = df.filter('Cancelled = 0  and Diverted = 0').select('Year','Month',
'DayofMonth','DayOfWeek','CRSDepTime','CRSArrTime','Distance','ArrDelay','CRSEla
psedTime','UniqueCarrier','FlightNum','TailNum','Origin','Dest')
>>> df_min.printSchema()
root
 |-- Year: integer (nullable = true)
 |-- Month: integer (nullable = true)
 |-- DayofMonth: integer (nullable = true)
 |-- DayOfWeek: integer (nullable = true)
 |-- CRSDepTime: integer (nullable = true)
 |-- CRSArrTime: integer (nullable = true)
 |-- Distance: integer (nullable = true)
 |-- ArrDelay: string (nullable = true)
 |-- CRSElapsedTime: integer (nullable = true)
 |-- UniqueCarrier: string (nullable = true)
 |-- FlightNum: integer (nullable = true)
 |-- TailNum: string (nullable = true)
 |-- Origin: string (nullable = true)
 |-- Dest: string (nullable = true)
```

Filtered out cancelled and diverted flights and removed data leakage columns

SparkSQL

Using SparkSQL, we converted strings to category codes



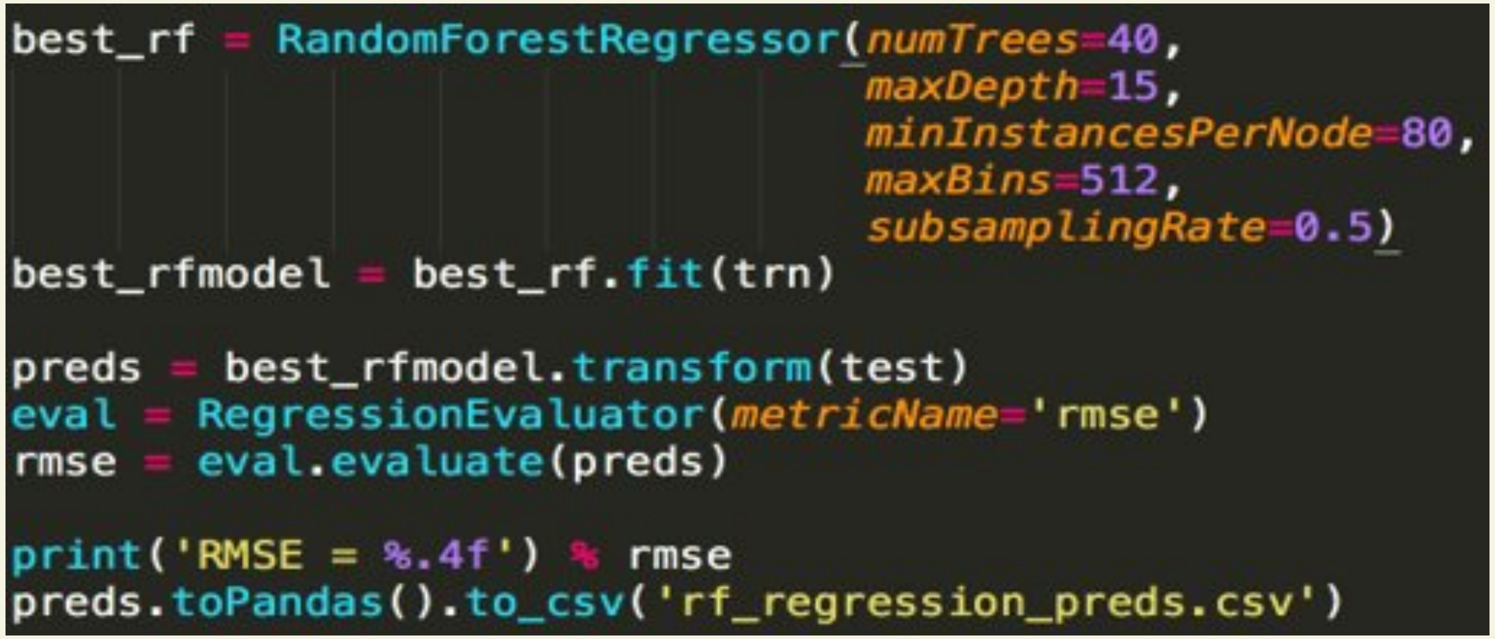
```
>>> sqlContext.sql('select * from flightsnumeric').show()
18/01/18 21:12:05 WARN SizeEstimator: Failed to check whether UseCompressedOops is set; assuming yes
+----+-----+----------+---------+----------+----------+--------+--------+--------------+-------------+---------+-------+------+-----+
|Year|Month|DayofMonth|DayOfWeek|CRSDepTime|CRSArrTime|Distance|ArrDelay|CRSElapsedTime|UniqueCarrier|FlightNum|TailNum|Origin| Dest|
+----+-----+----------+---------+----------+----------+--------+--------+--------------+-------------+---------+-------+------+-----+
|2008|    1|        16|        3|      1910|      2130|     375|     -18|            80|         16.0|   3428.0| 4461.0|  12.0|164.0|
|2008|    1|         5|        6|      1105|      1340|     490|      -9|            95|         16.0|   2917.0| 4630.0| 144.0| 12.0|
|2008|    1|        18|        5|      1910|      2130|     375|     244|            80|         16.0|   3428.0| 4630.0|  12.0|164.0|
|2008|    1|        19|        6|      1910|      2130|     375|      -1|            80|         16.0|   3428.0| 4495.0|  12.0|164.0|
|2008|    1|        20|        7|      1910|      2130|     375|     152|            80|         16.0|   3428.0| 4702.0|  12.0|164.0|
|2008|    1|        21|        1|      1910|      2130|     375|      28|            80|         16.0|   3428.0| 5017.0|  12.0|164.0|
|2008|    1|        24|        4|      1910|      2130|     375|      35|            80|         16.0|   3428.0| 4666.0|  12.0|164.0|
|2008|    1|        25|        5|      1910|      2130|     375|       0|            80|         16.0|   3428.0| 4527.0|  12.0|164.0|
|2008|    1|        22|        2|      1910|      2130|     375|       2|            80|         16.0|   3428.0| 4633.0|  12.0|164.0|
|2008|    1|        26|        6|      1910|      2130|     375|      -6|            80|         16.0|   3428.0| 4653.0|  12.0|164.0|
|2008|    1|        27|        7|      1910|      2130|     375|      -8|            80|         16.0|   3428.0| 4679.0|  12.0|164.0|
|2008|    1|        28|        1|      1910|      2130|     375|      27|            80|         16.0|   3428.0| 4716.0|  12.0|164.0|
|2008|    1|        30|        3|      1910|      2130|     375|      14|            80|         16.0|   3428.0| 4667.0|  12.0|164.0|
|2008|    1|         7|        1|      1600|      1746|     449|      28|           106|         16.0|   3428.0| 4489.0|  31.0| 12.0|
|2008|    1|         8|        2|      1600|      1746|     449|      15|           106|         16.0|   3428.0| 5595.0|  31.0| 12.0|
|2008|    1|         9|        3|      1600|      1746|     449|       8|           106|         16.0|   3428.0| 4549.0|  31.0| 12.0|
|2008|    1|        10|        4|      1600|      1746|     449|      17|           106|         16.0|   3428.0| 4434.0|  31.0| 12.0|
|2008|    1|        29|        2|      1910|      2130|     375|      32|            80|         16.0|   3428.0| 4489.0|  12.0|164.0|
|2008|    1|        12|        6|      1600|      1746|     449|       1|           106|         16.0|   3428.0| 4660.0|  31.0| 12.0|
|2008|    1|        13|        7|      1600|      1746|     449|     -16|           106|         16.0|   3428.0| 4640.0|  31.0| 12.0|
+----+-----+----------+---------+----------+----------+--------+--------+--------------+-------------+---------+-------+------+-----+
only showing top 20 rows
```

Querying Data Using SparkSQL

SparkML

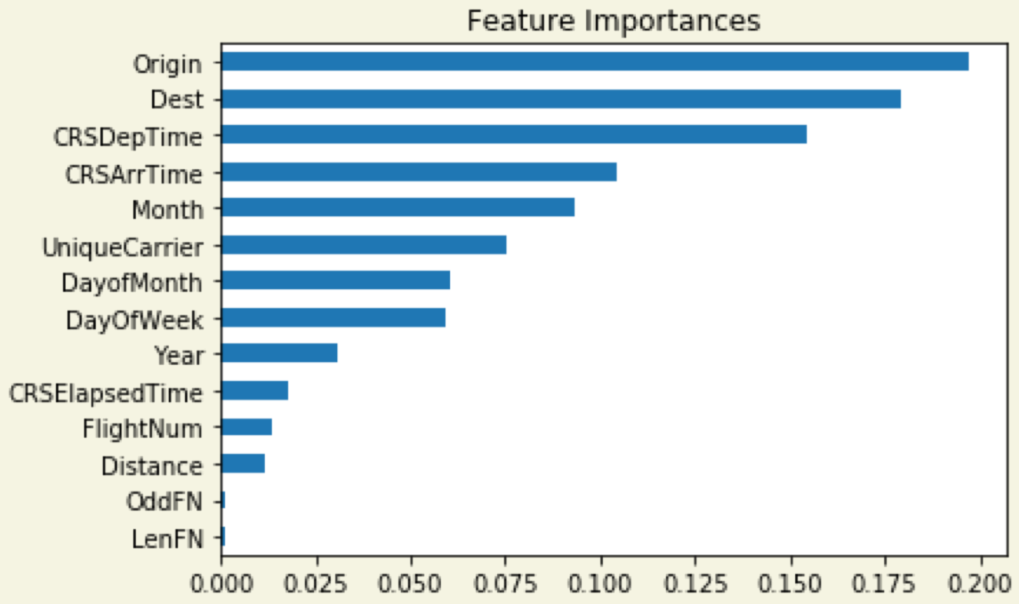
- Feature Engineering:
  - Odd/Even Flight Num
    - odd flight numbers correspond to westbound/southbound flights
    - even flight numbers correspond to eastbound/northbound flights
  - Num Digits of Flight Num
    - Less than 3 digits correspond to long-haul or otherwise premium flights

SparkML

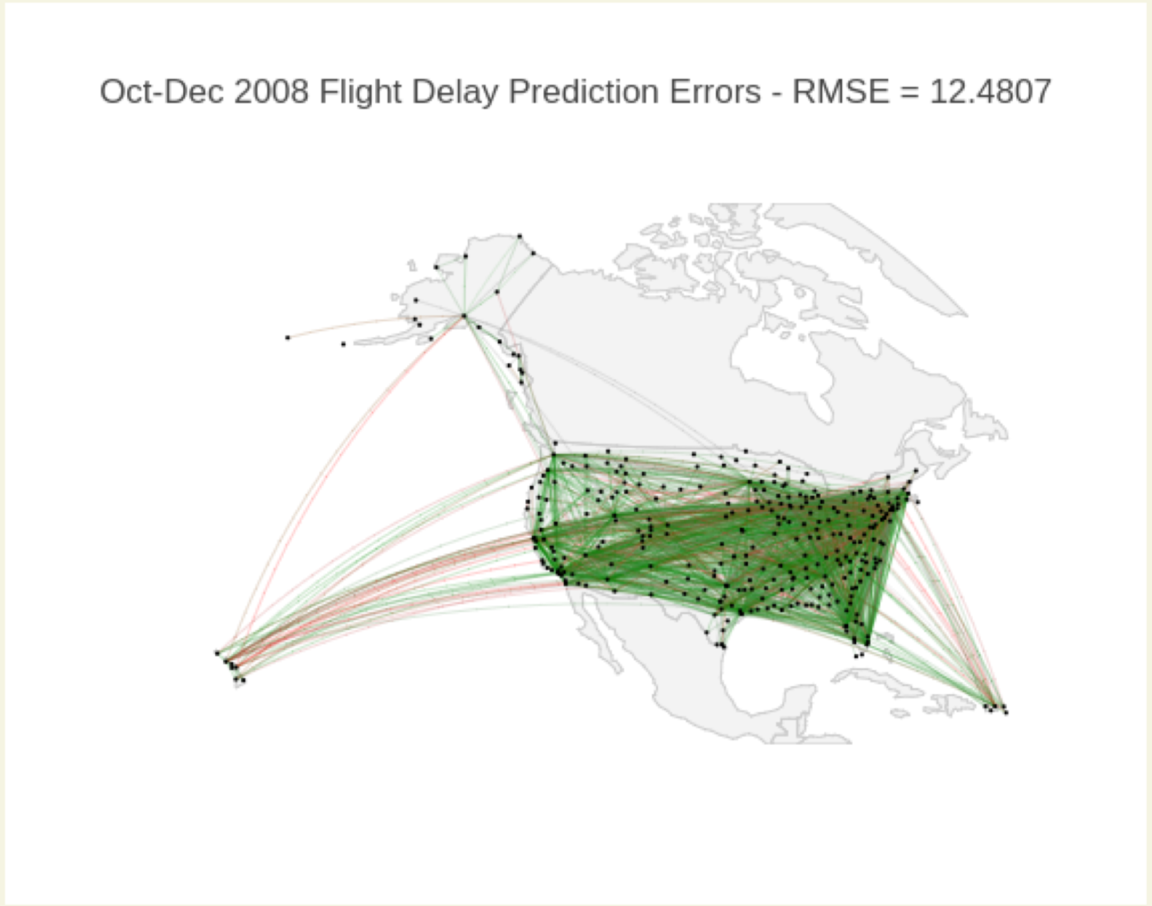


```
best_rf = RandomForestRegressor(numTrees=40,
                                maxDepth=15,
                                minInstancesPerNode=80,
                                maxBins=512,
                                subsamplingRate=0.5)
best_rfmodel = best_rf.fit(trn)

preds = best_rfmodel.transform(test)
eval = RegressionEvaluator(metricName='rmse')
rmse = eval.evaluate(preds)

print('RMSE = %.4f') % rmse
preds.toPandas().to_csv('rf_regression_preds.csv')
```

Results



Feature Importances
Origin
Dest
CRSDepTime
CRSArrTime
Month
UniqueCarrier
DayofMonth
DayOfWeek
Year
CRSElapsedTime
FlightNum
Distance
OddFN
LenFN
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200



Oct-Dec 2008 Flight Delay Prediction Errors - RMSE = 12.4807

Lessons Learned

- Having undesirable number of partitions led the machine learning code to run 3x slower
- Running Random Forest Regression with 3 m1.large instances (default machines in ec2-cluster) led to memory issues.
- Diagnose code issues by first running only a fraction of your data, not all of it.
- Due to time constraints, we used maxDepth=15, preventing our model from capturing the high cardinality of the Origin/Destination features