Exploring and visualizing the baby weight dataset with tensor Flow

- By - Gautam Sharma

Create ML dataset by sampling using BigQuery

Let's sample the BigQuery data to create smaller datasets.

In [8]:
```
# Create SQL query using natality data after the year 2000
query = """
SELECT
  weight_pounds,
  is_male,
  mother_age,
  plurality,
  gestation_weeks,
  FARM_FINGERPRINT(CONCAT(CAST(YEAR AS STRING), CAST(month AS STRING))) AS hashm
FROM
  publicdata.samples.natality
WHERE year > 2000
"""
```

There are limited number of year and months in the datasets.Let's ses What the hashmonths are

In [3]:
```
from google.cloud import bigquery
```

In [10]:
```
# Call Bigquery but GROUP by hashmonth and see number of recordsfor eACH GROUP T
df = bigquery.Client().query("SELECT hashmonth, COUNT(weight_pounds) AS num_babi
```

In [12]:
```
print("There are {} unique hasmonths. ".format(len(df)))
```

```
There are 96 unique hasmonths.
```

In [13]:
```
df.head()
```

Out[13]:

	hashmonth	num_babies
0	3408502330831153141	323970
1	1088037545023002395	362125
2	-9068386407968572094	330863
3	7186614341837170520	346919
4	-6782146986770280327	345092

Here's a way to get a well distributed portion of the data in such a way that the test and train sets do not overlap:

In [14]:
```
# Add the RAND() so that we can now subsample from each ofthe hashmots to get ap
trainQuery = "SELECT * FROM (" + query + ") WHERE ABS(MOD(hashmonth, 4)) < 3 AND
evalQuery = "SELECT * FROM (" + query + ") WHERE ABS(MOD(hashmonth, 4)) = 3 AND
```

In [15]:
```
traindf = bigquery.Client().query(trainQuery).to_dataframe()
traindf.head()
```

Out[15]:

	weight_pounds	is_male	mother_age	plurality	gestation_weeks	hashmonth
0	8.249698	True	32	1	38.0	3095933535584005890
1	8.344497	False	17	1	39.0	3095933535584005890
2	7.687519	True	22	1	40.0	3095933535584005890
3	6.206013	True	30	1	37.0	3095933535584005890
4	5.820204	True	31	1	39.0	3095933535584005890

In [16]:
```
evaldf = bigquery.Client().query(evalQuery).to_dataframe()
evaldf.head()
```

Out[16]:

	weight_pounds	is_male	mother_age	plurality	gestation_weeks	hashmonth
0	7.749249	True	25	1	40.0	-7146494315947640619
1	7.738225	True	31	1	39.0	8904940584331855459
2	5.485101	True	26	1	39.0	-1866590652208008467
3	7.341393	True	28	1	37.0	-1891060869255459203
4	6.073735	True	21	1	39.0	3182182455926341111

Preprocess data using pandas

In [18]:
```
traindf.describe()
```

Out[18]:

	weight_pounds	mother_age	plurality	gestation_weeks	hashmonth
count	13176.000000	13191.000000	13191.000000	13109.000000	1.319100e+04
mean	7.214085	27.428929	1.036464	38.604318	3.062626e+17
std	1.329190	6.135315	0.196146	2.596425	5.181922e+18
min	0.562179	12.000000	1.000000	19.000000	-9.183606e+18
25%	6.560957	23.000000	1.000000	38.000000	-3.340563e+18
50%	7.312733	27.000000	1.000000	39.000000	-3.280124e+17

	weight_pounds	mother_age	plurality	gestation_weeks	hashmonth
75%	8.062305	32.000000	1.000000	40.000000	4.331750e+18
max	12.625874	50.000000	4.000000	47.000000	8.599690e+18

In [31]:
```
# It is always crucial to clean raw data befire using in ML , so we have a prepr
import pandas as pd
def preprocess(df):
    df = df[df.weight_pounds > 0] # Clean up data
    df = df[df.mother_age > 0] #
    df = df[df.gestation_weeks > 0]
    df = df[df.plurality > 0]

    # Modify plurality field to be a string
    twins_etc = dict(zip([1,2,3,4,5],['Single(1)','Twins(2)', 'Triplates(3)', 'Q

    df['plurality'].replace(twins_etc, inplace = True)

    # Now create extra rows to simulate lack of ultrasound
    nous = df.copy(deep=True)
    nous.loc[nous['plurality'] != 'Single(1)', 'plurality'] = 'Multiple(2+)'
    nous['is_male'] = 'Unkown'

    return pd.concat([df,nous])
```

In [29]:
```
# Let's see a small sample of the training data now after our preprocessing
traindf.head()
```

Out[29]:

	weight_pounds	is_male	mother_age	plurality	gestation_weeks	hashmonth
0	8.249698	True	32	1	38.0	3095933535584005890
1	8.344497	False	17	1	39.0	3095933535584005890
2	7.687519	True	22	1	40.0	3095933535584005890
3	6.206013	True	30	1	37.0	3095933535584005890
4	5.820204	True	31	1	39.0	3095933535584005890

In [32]:
```
traindf = preprocess(traindf)
```

In [33]:
```
evaldf = preprocess(evaldf)
```

In [34]:
```
traindf.head()
```

Out[34]:

	weight_pounds	is_male	mother_age	plurality	gestation_weeks	hashmonth
0	8.249698	True	32	Single(1)	38.0	3095933535584005890
1	8.344497	False	17	Single(1)	39.0	3095933535584005890
2	7.687519	True	22	Single(1)	40.0	3095933535584005890

	weight_pounds	is_male	mother_age	plurality	gestation_weeks	hashmonth
3	6.206013	True	30	Single(1)	37.0	3095933535584005890
4	5.820204	True	31	Single(1)	39.0	3095933535584005890

In [35]:
```
traindf.tail()
```

Out[35]:

	weight_pounds	is_male	mother_age	plurality	gestation_weeks	hashmonth
13186	8.249698	Unkown	31	Single(1)	39.0	-774501970389208065
13187	8.311427	Unkown	29	Single(1)	40.0	-774501970389208065
13188	6.750554	Unkown	33	Multiple(2+)	37.0	-774501970389208065
13189	2.436108	Unkown	34	Multiple(2+)	29.0	-774501970389208065
13190	7.687519	Unkown	36	Single(1)	37.0	-774501970389208065

In [36]:
```
# Describe only does numeric columns, so you won't see plurality
traindf.describe()
```

Out[36]:

	weight_pounds	mother_age	gestation_weeks	hashmonth
count	26198.000000	26198.000000	26198.000000	2.619800e+04
mean	7.213554	27.431102	38.606917	3.070365e+17
std	1.329456	6.129972	2.588245	5.181774e+18
min	0.562179	12.000000	19.000000	-9.183606e+18
25%	6.560957	23.000000	38.000000	-3.340563e+18
50%	7.312733	27.000000	39.000000	-3.280124e+17
75%	8.062305	32.000000	40.000000	4.331750e+18
max	12.625874	50.000000	47.000000	8.599690e+18

Write out

In the final versions, we want to read from files, not Pandas dataframes. So, write the Pandas dataframes out as CSV files. Using CSV files gives us the advantage of shuffling during read. This is important for distributed training because some workers might be slower than others, and shuffling the data helps prevent the same data from being assigned to the slow workers.

In [38]:
```
traindf.to_csv('train.csv', index=False, header=False)
evaldf.to_csv('eval.csv',index=False, header=False)
```

In [39]:
```
%%bash
```

```
wc -l *.csv
head *.csv
tail *.csv
```

```
   6634 eval.csv
  26198 train.csv
  32832 total
==> eval.csv <==
7.7492485093,True,25,Single(1),40.0,-7146494315947640619
7.7382253962,True,31,Single(1),39.0,8904940584331855459
5.48510107856,True,26,Single(1),39.0,-1866590652208008467
7.3413933246,True,28,Single(1),37.0,-1891060869255459203
6.0737353181,True,21,Single(1),39.0,3182182455926341111
9.18666245754,True,38,Single(1),40.0,3182182455926341111
7.605948039,False,22,Single(1),42.0,-7517141034410775575
6.06050758238,True,23,Single(1),39.0,-1866590652208008467
3.12615487516,False,25,Twins(2),36.0,-6141045177192779423
8.8515598193,True,27,Single(1),44.0,6365946696709051755

==> train.csv <==
8.24969784404,True,32,Single(1),38.0,3095933535584005890
8.344496616699999,False,17,Single(1),39.0,3095933535584005890
7.68751907594,True,22,Single(1),40.0,3095933535584005890
6.2060126753,True,30,Single(1),37.0,3095933535584005890
5.8202037168,True,31,Single(1),39.0,3095933535584005890
6.8012607827,False,28,Single(1),38.0,3095933535584005890
8.1901730333,True,25,Single(1),40.0,3095933535584005890
6.75055446244,False,30,Single(1),41.0,3095933535584005890
8.18796841068,True,29,Single(1),39.0,3095933535584005890
8.82069510262,True,28,Single(1),42.0,3095933535584005890
==> eval.csv <==
7.62578964258,Unkown,26,Single(1),39.0,-1891060869255459203
6.686620406459999,Unkown,27,Single(1),37.0,74931465496927487
6.7020527647999995,Unkown,36,Single(1),40.0,74931465496927487
9.56365292556,Unkown,24,Single(1),40.0,6392072535155213407
6.0009827716399995,Unkown,23,Single(1),36.0,-7517141034410775575
6.9225150268,Unkown,27,Single(1),38.0,270792696282171059
6.4992274837599995,Unkown,31,Single(1),38.0,1569531340167098963
6.686620406459999,Unkown,31,Single(1),39.0,2246942437170405963
8.37315671076,Unkown,35,Single(1),38.0,-1866590652208008467
8.377565956,Unkown,30,Single(1),40.0,-1891060869255459203

==> train.csv <==
8.58039123704,Unkown,23,Single(1),40.0,-774501970389208065
5.74524654772,Unkown,35,Single(1),40.0,-774501970389208065
4.98685636644,Unkown,39,Multiple(2+),38.0,-774501970389208065
6.4992274837599995,Unkown,18,Single(1),38.0,-774501970389208065
6.8122838958,Unkown,26,Single(1),39.0,-774501970389208065
8.24969784404,Unkown,31,Single(1),39.0,-774501970389208065
8.3114272774,Unkown,29,Single(1),40.0,-774501970389208065
6.75055446244,Unkown,33,Multiple(2+),37.0,-774501970389208065
2.4361079951,Unkown,34,Multiple(2+),29.0,-774501970389208065
7.68751907594,Unkown,36,Single(1),37.0,-774501970389208065
```

In [ ]: