Google Cloud

Generalization & Sampling

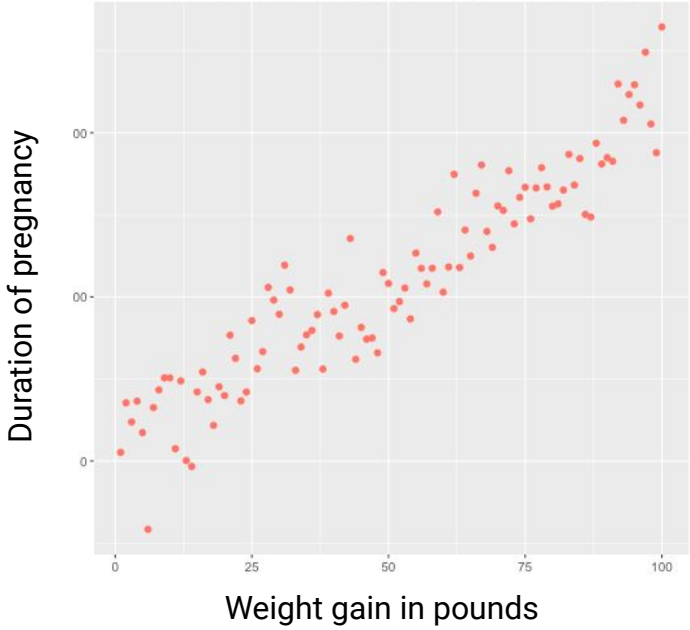
Agenda

Generalization

Sampling

Suppose we want to predict duration of pregnancy based on mother's weight gain in pounds

What is the error measure to optimize?

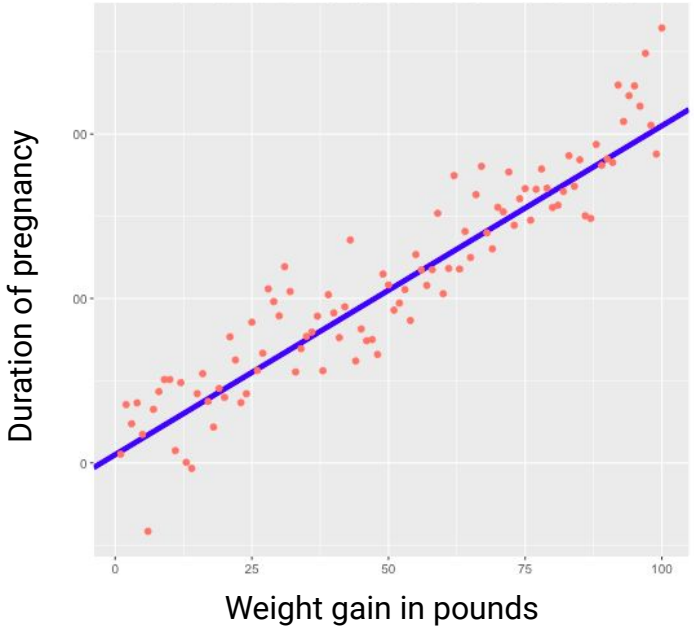


Model 1 is a linear model using linear regression

Red = training examples

Blue = model prediction for each baby

RMSE = 2.224



Model 2 has more free parameters

RMSE = 0

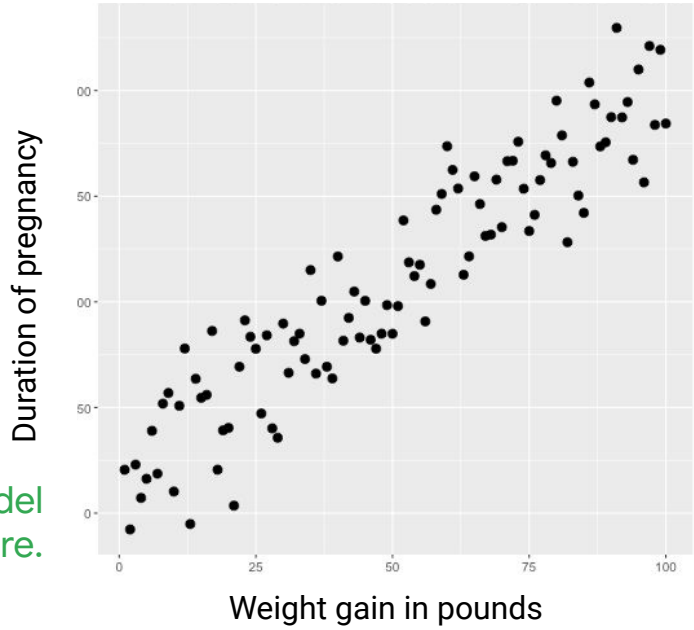
Which model is better?

How can you tell?

Does the model generalize to new data?

Need data that were not used in training.

New data the model hasn't seen before.



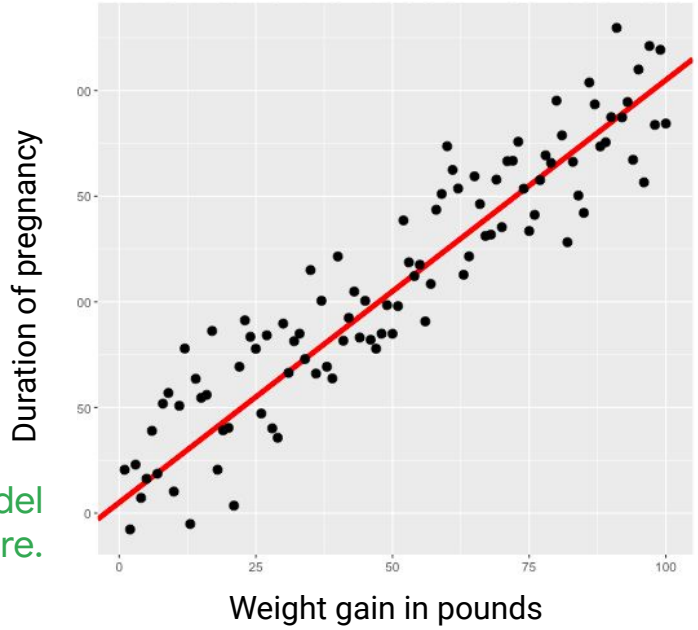
Model 1 generalizes well

Old RMSE = 2.224

New RMSE = 2.198

Pretty similar = good

New data the model hasn't seen before.



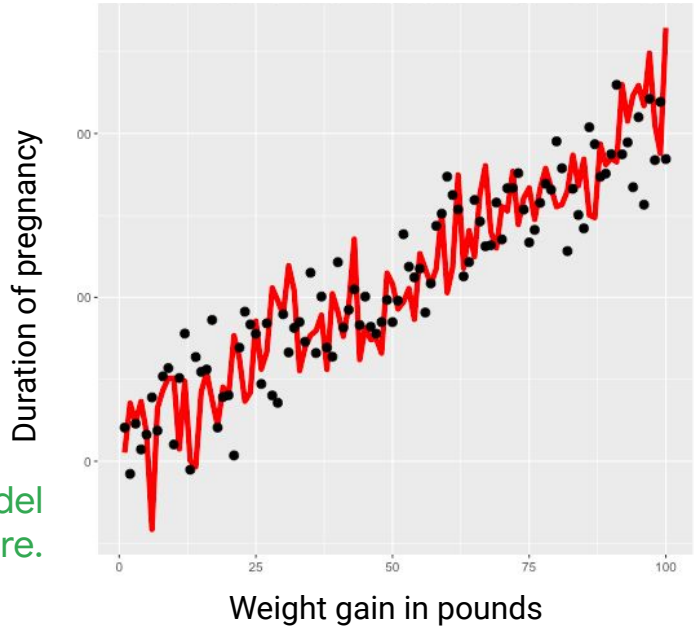
Model 2 does not generalize well

Old RMSE = 0

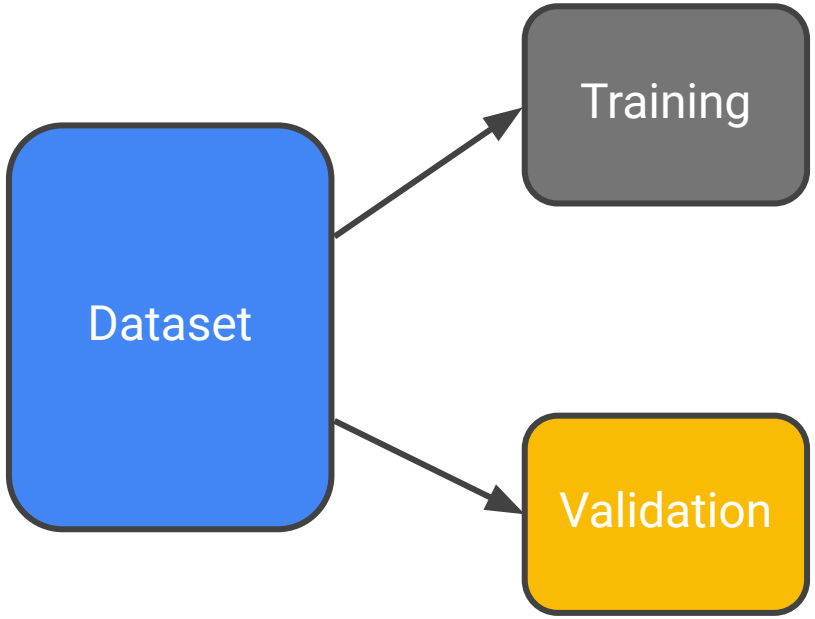
New RMSE = 3.2

This is a red flag

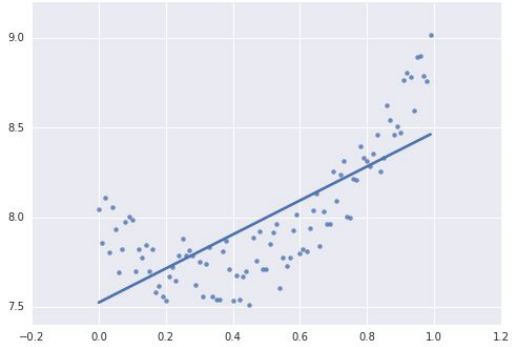
New data the model hasn't seen before.



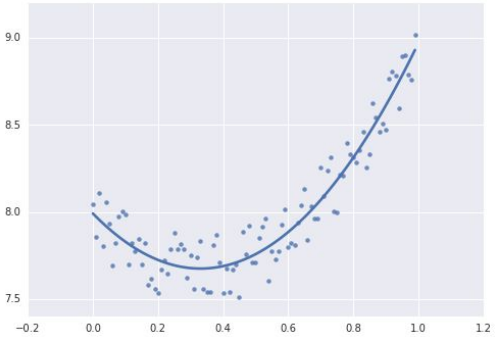
Split the dataset and experiment with models



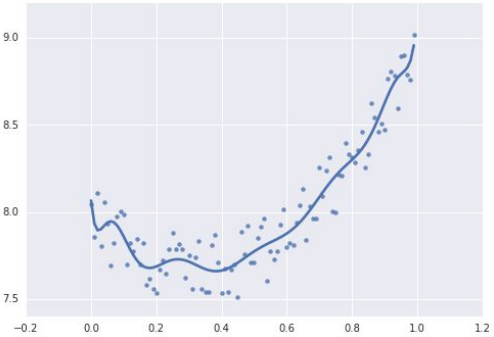
Beware of overfitting as you increase model complexity



Underfit

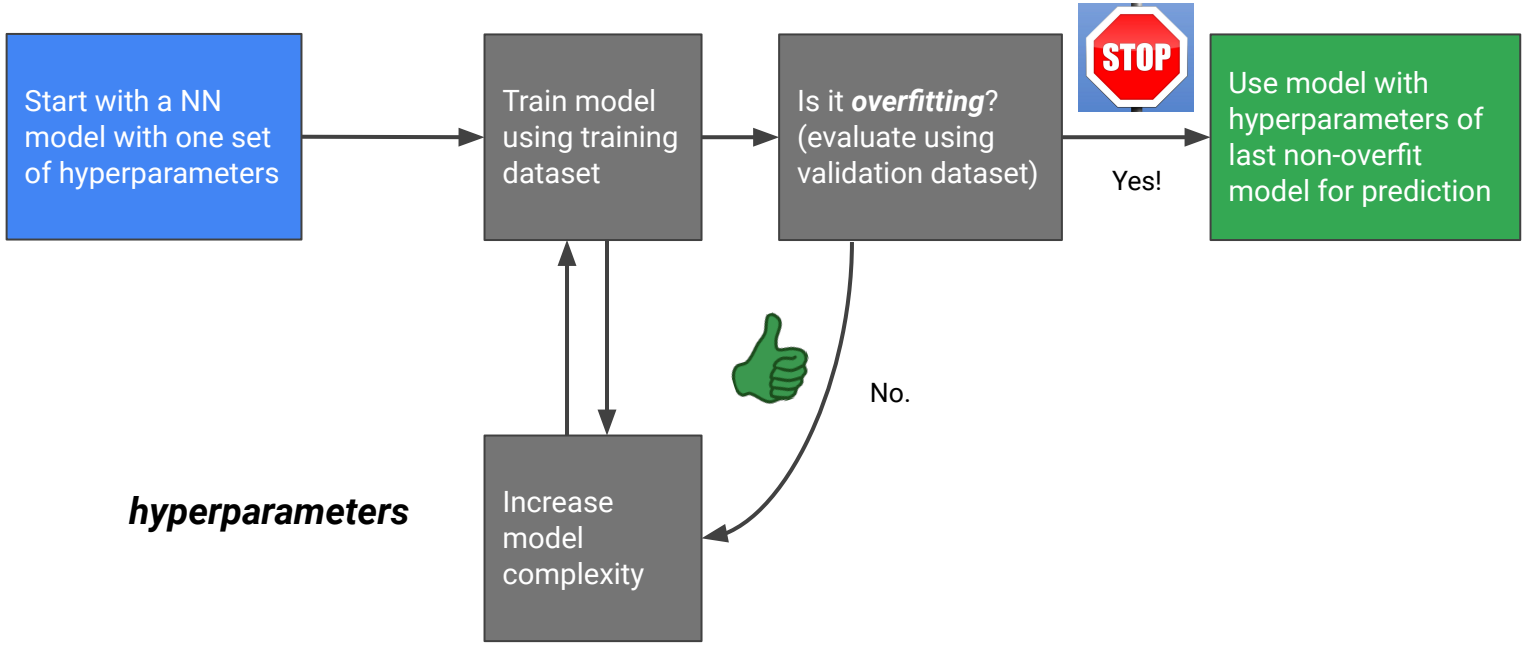


Fit

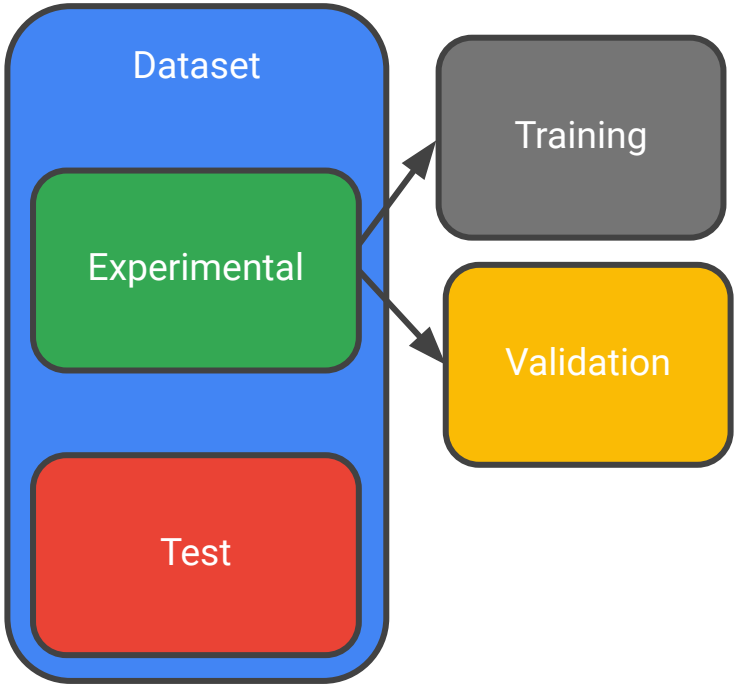


Overfit

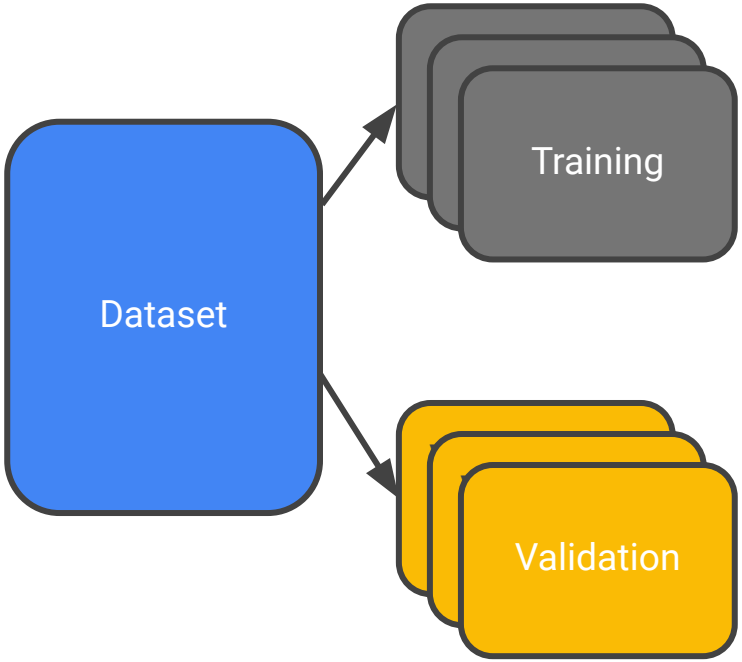
You can use the validation dataset to experiment with model complexity



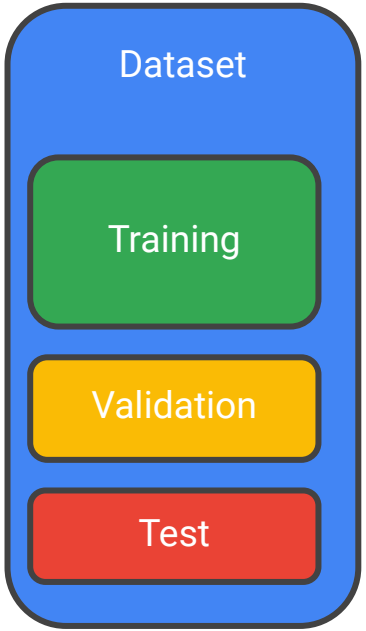
Evaluate the final model with independent test data



Evaluate the final model with cross-validation



We often have large datasets in BigQuery that we want to use for machine learning



Row	date	airline	departure_airport	departure_schedule	arrival_airport	arrival_delay
1	2004-08-07	TZ	SRQ	1255	IND	-14.0
2	2004-03-05	TZ	SRQ	2117	IND	-9.0
3	2004-04-12	TZ	SRQ	2000	IND	-17.0
4	2003-04-16	TZ	SRQ	1215	IND	-5.0
5	2005-03-20	TZ	SRQ	645	IND	14.0
6	2003-04-06	TZ	SRQ	1235	IND	-8.0

It's easy to get a random 80% of your dataset for training

```
#standardSQL
SELECT
  date,
  airline,
  departure_airport,
  departure_schedule,
  arrival_airport,
  arrival_delay
FROM
  `bigquery-samples.airline_ontime_data.flights`
WHERE
  RAND() < 0.8
```

RAND will return a number between 0 and 1.

However, experimentation requires repeatability

You need to know which specific data was involved in training, validation, and testing.

Naive random splitting is not repeatable

Order of rows in BigQuery is not certain without ORDER BY.

Hard to identify and split the remaining 20% of data for validation and testing.

RAND() will return different results each time →

New Query

Query Editor | UDF Editor

SQL

```sql
#standardSQL
SELECT
  date,
  airline,
  departure_airport,
  departure_schedule,
  arrival_airport,
  arrival_delay
FROM
  `bigquery-samples.airline_ontime_data.flights`
WHERE
  RAND() < 0.8 # returns different records each time

LIMIT 5;
```

Ctrl + Enter: run query, Tab or Ctrl + Space: autocomplete.

Cancel Query | Save Query | Save View | Format Query | Show Options | Query running (1.0s)...

Results | Explanation | Job Information | Download as CSV | Download as JSON | Save as Table | Save to Google Sheets

Row	date	airline	departure_airport	departure_schedule	arrival_airport	arrival_delay
1	2005-07-07	NW	DTW	700	MSP	-9.0
2	2005-07-04	NW	DTW	700	MSP	-9.0
3	2005-07-06	NW	DTW	700	MSP	19.0
4	2005-07-08	NW	DTW	700	MSP	-19.0
5	2005-07-05	NW	DTW	700	MSP	36.0

Table | JSON

Solution: Split a dataset into training/validation/test using the hashing and modulo operators

```
#standardSQL
SELECT
  date,
  airline,
  departure_airport,
  departure_schedule,
  arrival_airport,
  arrival_delay
FROM
 `bigquery-samples.airline_ontime_data.flights`

WHERE
  MOD(ABS(FARM_FINGERPRINT(date)),10) < 8
```

Note: Even though we select date, our model wouldn't actually use it during training.

Hash value on the Date will always return the same value.

Then we can use a modulo operator to only pull 80% of that data based on the last few hash digits.

Carefully choose which field will split your data

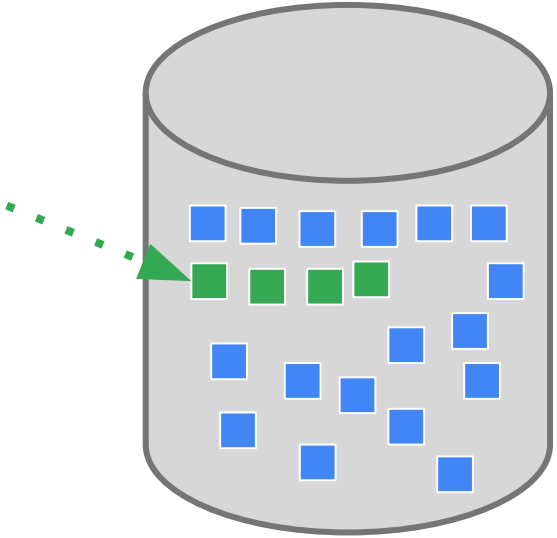
We hypothesize that flight delay depends on the carrier, time of day, weather, and airport characteristics (# of runways, etc.) We want to predict flight delays. What field should we split our data on?

- Hash on date?
- Hash on airport?
- Hash on carrier name?

Split your data on a field you can afford to lose.

Developing the ML model software on the entire dataset can be expensive; you want to to develop on a smaller sample

Develop your TensorFlow code on a small subset of data, then scale it out to the cloud.



Full Dataset

Pitfall: Chaining hashes to create subsets won't work

```
#standardSQL
SELECT
  date,
  airline,
  departure_airport,
  departure_schedule,
  arrival_airport,
  arrival_delay
FROM
 `bigquery-samples.airline_ontime_data.flights`
WHERE
  MOD(ABS(FARM_FINGERPRINT(date)),70) = 0
     AND
  MOD(ABS(FARM_FINGERPRINT(date)),10) < 8
```

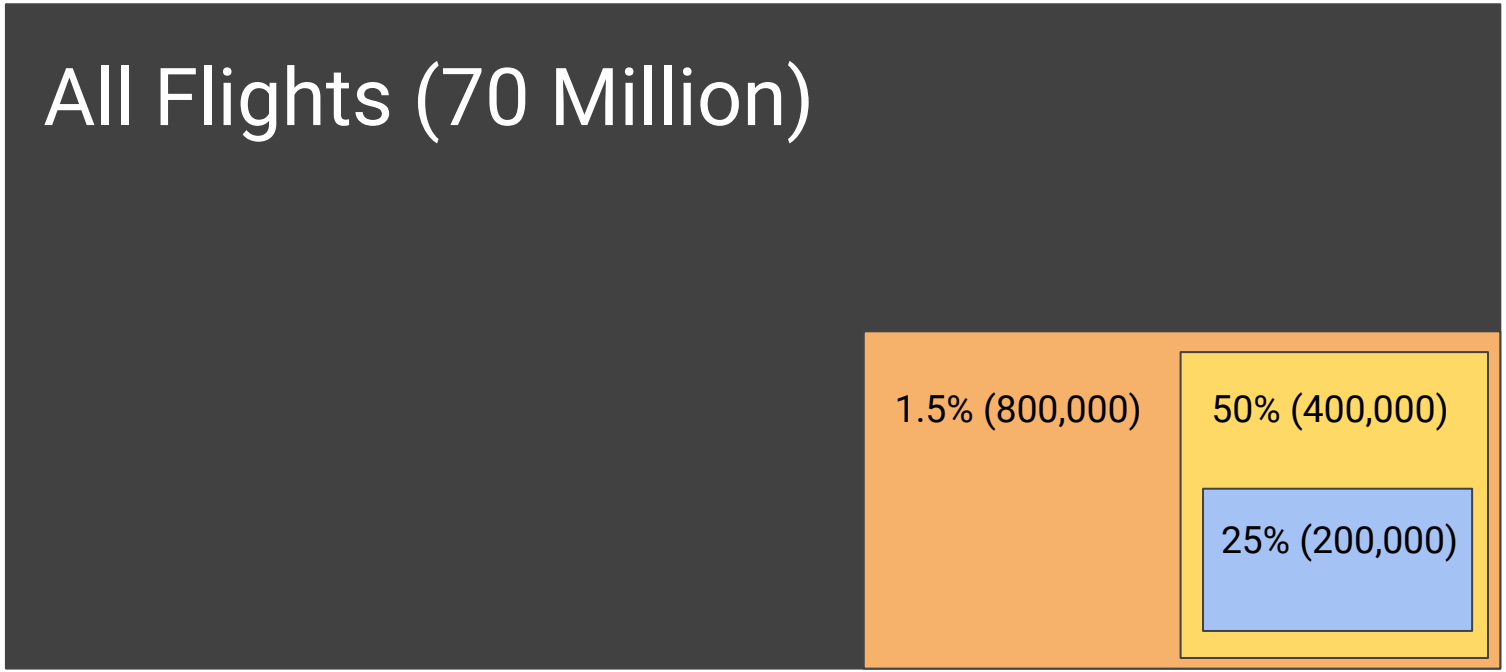
Then take 1 in 70 flights.

Take 80% of the dataset? Incorrect!

All records here will also be divisible by 10 (there is no new filtering happening!)

Demo of Splitting Datasets in BigQuery

How we want to split our data



We can extend this to creating 3 splits

```
#standardSQL
SELECT
  date,
  airline,
  departure_airport,
  departure_schedule,
  arrival_airport,
  arrival_delay
FROM
 `bigquery-samples.airline_ontime_data.flights`
WHERE
  MOD(ABS(FARM_FINGERPRINT(date)),70) = 0
     AND
  MOD(ABS(FARM_FINGERPRINT(date)),700) >= 350
    AND
  MOD(ABS(FARM_FINGERPRINT(date)),700) < 525
```

Then take 1 in 70 flights.

Ignore the 50% of the dataset (training).

Choose data between 350 and 524 which is a new 25% sample for Validation.

Lab

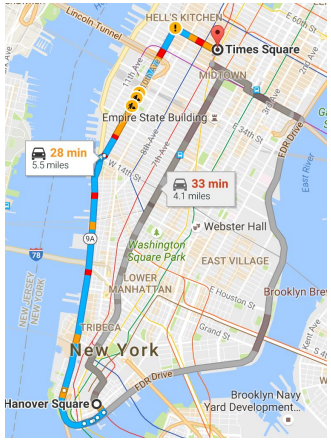
Explore and clean ML datasets to estimate cab fare

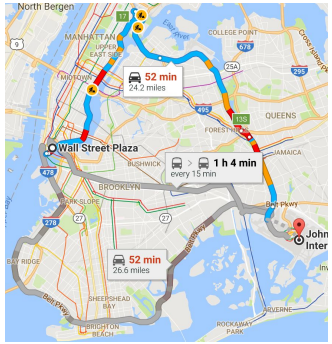
In this lab, you will estimate taxi fares in New York City.

training-data-analyst/courses/ machine_learning/deepdive/ 01_bigquery/a_sample_explore_clean.ipynb

Taxi fares:

$2.50 initial charge
+
50c per ⅕ mile
(or)
50c per minute if stopped
+
Passenger pays tolls
+
Various special charges





Lab: Setup environment

Step 0: Run setup.ipynb

Set-up procedure

Update notebooks with your project, bucket, and region

Step 0

Create a bucket if you haven't already.

Step 2

Fill in with your GCP project, bucket, and desired region.

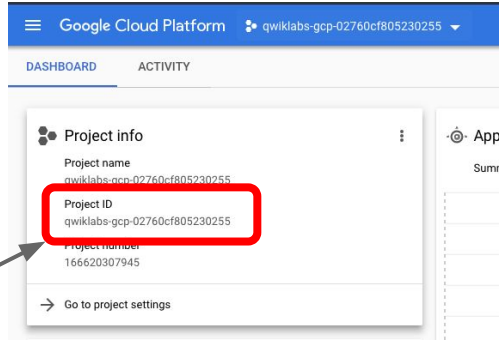
```
[1]: !ls

01_bigquery            04_advanced_preprocessing  09_sequence     setup.ipynb
02_tensorflow          05_review                  10_recommend
03_model_performance   08_image                   challenge_soln

[2]: # e.g., %env PROJECT=qwiklabs-...
     %env PROJECT=crawles-sandbox
     %env BUCKET=crawles-sandbox
     %env REGION=us-central1

     env: PROJECT=crawles-sandbox
     env: BUCKET=crawles-sandbox
     env: REGION=us-central1
```

Update with your project, bucket, region



project id

training-data-analyst/courses/ machine_learning/deepdive/setup.ipynb

Benchmarks are important to know what error metric is “reasonable” and/or “great” for the problem

The benchmark helps you set a goal for a good value for the error metric.

Often a simple heuristic rule can function as a good benchmark.

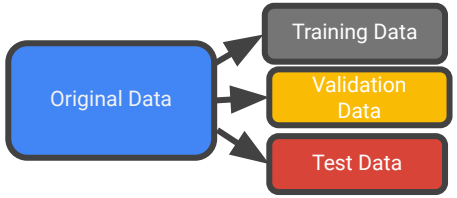
What’s a good benchmark for the taxi fare prediction?

Lab

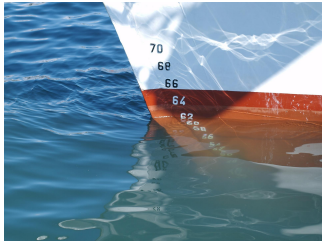
Create repeatable splits and build a benchmark

In this lab, you will explore a dataset using BigQuery; sample the dataset and create training, validation, and testing datasets for local development of TensorFlow models; and create a benchmark to evaluate the performance of ML against.

training-data-analyst/courses/machine_learning/deepdive/01_bigquery/c_extract_and_benchmark.ipynb



1. Create ML Datasets



2. Benchmark

Introducing BigQuery ML

Syntax for creating a model

```
#standardSQL
CREATE or REPLACE MODEL
     bqml_airplanes.airplane_delay_model
OPTIONS(model_type='linear_reg',
        input_label_cols=['label']) AS
SELECT
  airline,
  departure_airport,
  departure_schedule,
  arrival_airport,
  arrival_delay * departure_delay AS label
FROM
 `bigquery-samples.airline_ontime_data.flights`
WHERE
 MOD(ABS(FARM_FINGERPRINT(date)), 100) = 0
```

Defining the model name, type, and training label

Select data to train on like a normal SQL query

Get training statistics

```
#standardSQL
SELECT *
FROM ML.TRAINING_INFO(MODEL `bqml_airplanes.airplane_delay_model`)
```

Make a prediction

```
SELECT predicted_label
FROM
  ML.PREDICT(MODEL `bqml_airplanes.airplane_delay_model`,
    (
    SELECT
      'OO' as airline,
      'ATL' as departure_airport,
      941 as departure_schedule,
      'HOU' as arrival_airport
    ))
```

Lab

Build a model in BigQuery to estimate cab fare

In this lab, you will build a machine learning model using BigQueryML

training-data-analyst/courses/machine_learning/deepdive/01_bigquery/b_bqml.ipynb