Launching into ML

Day 2

Machine Learning on Google Cloud learning path

1. How Google Does Machine Learning
2. Launching into ML
3. Introduction to TensorFlow
4. Feature Engineering
5. The Art and Science of ML

Learn how to ...

Improve data quality and perform exploratory data analysis.

Identify why deep learning is currently popular.

Optimize and evaluate models using loss functions and performance metrics.

Mitigate common problems that arise in machine learning.

Create repeatable training, evaluation, and test datasets.

Course Agenda

Improve Data Quality

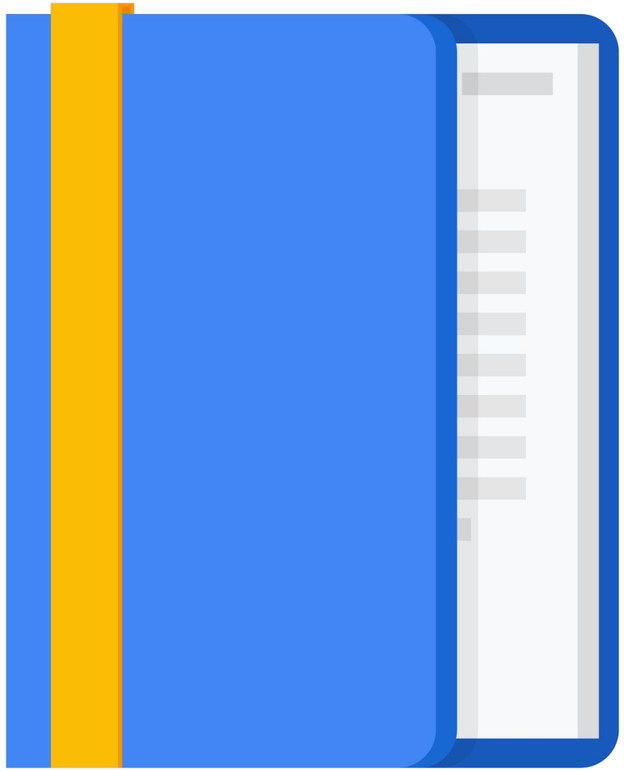
Exploratory Data Analysis

ML in Practice

Optimization

Generalization and Sampling

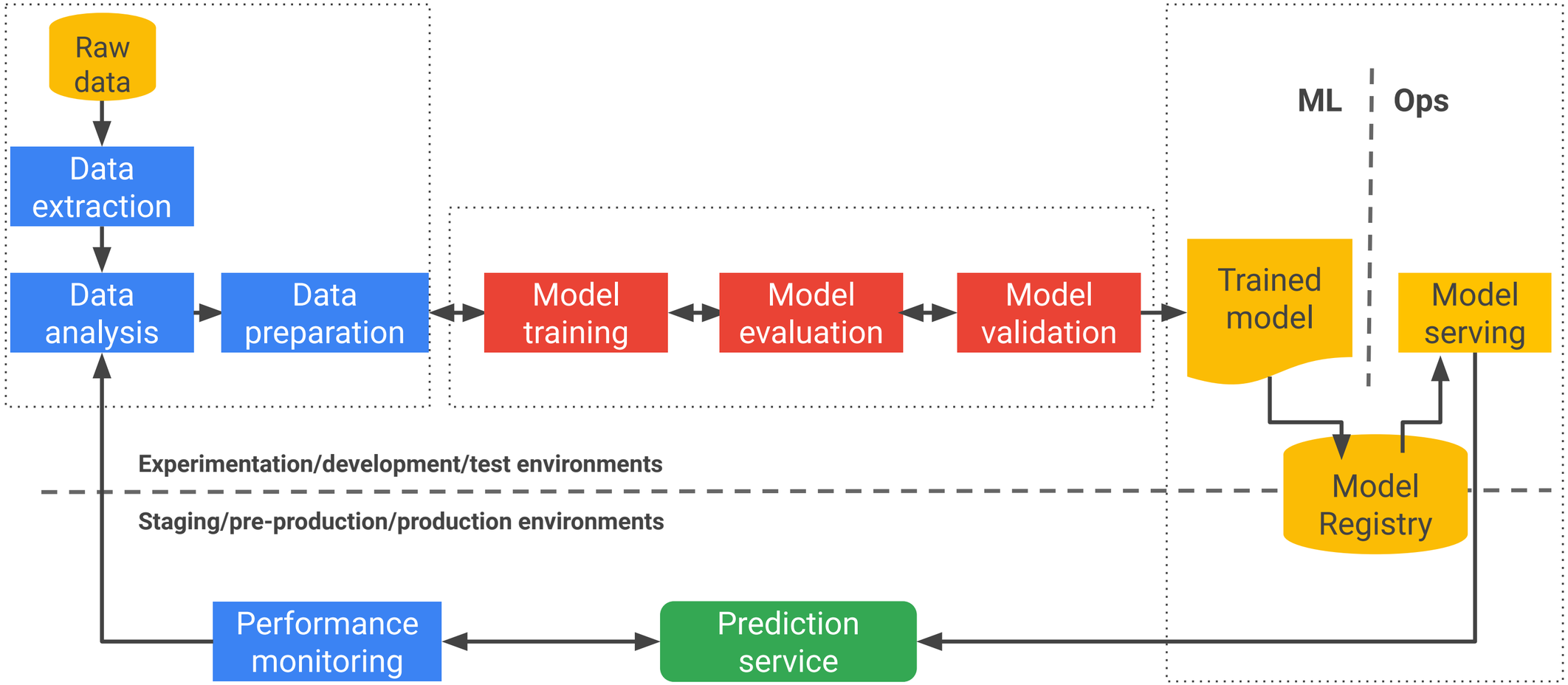
Course Summary



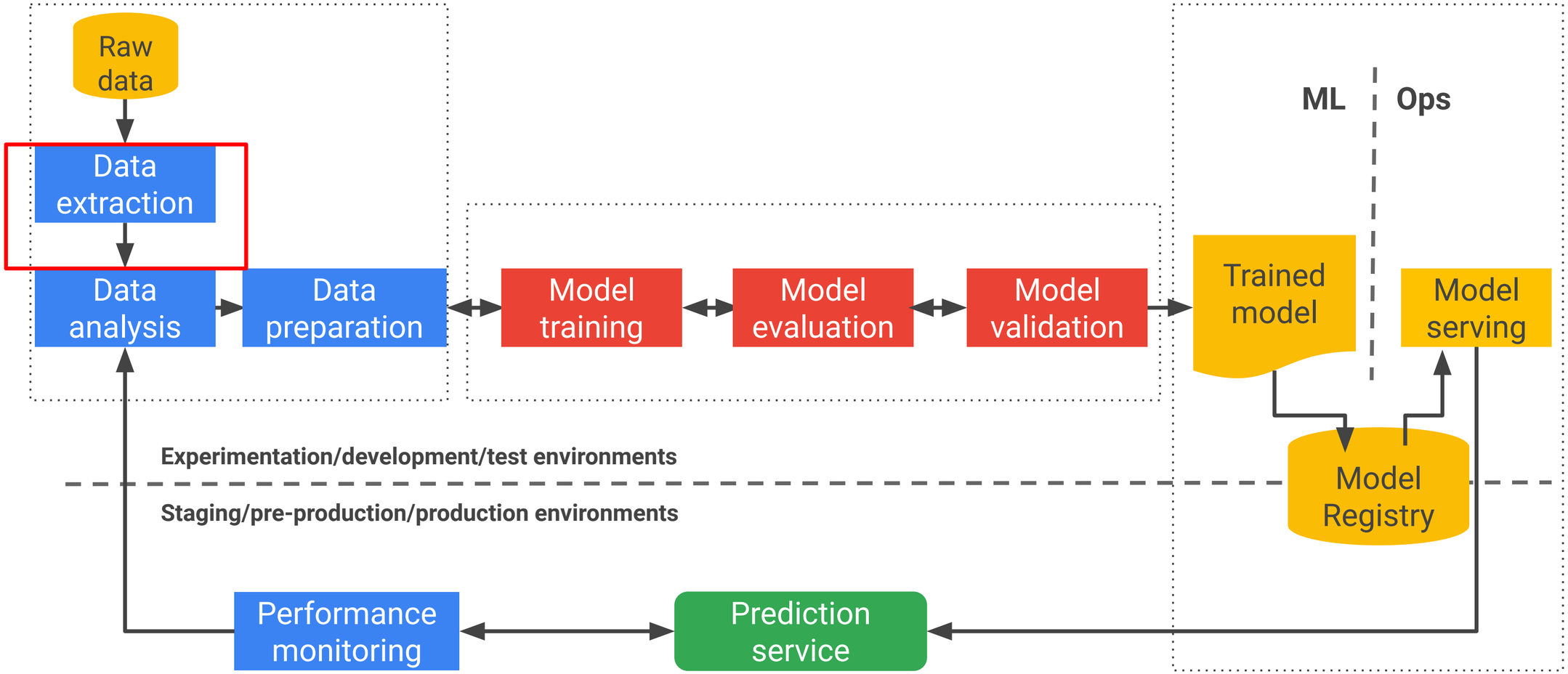
Launching into ML: Improve Data Quality

Machine Learning Phases

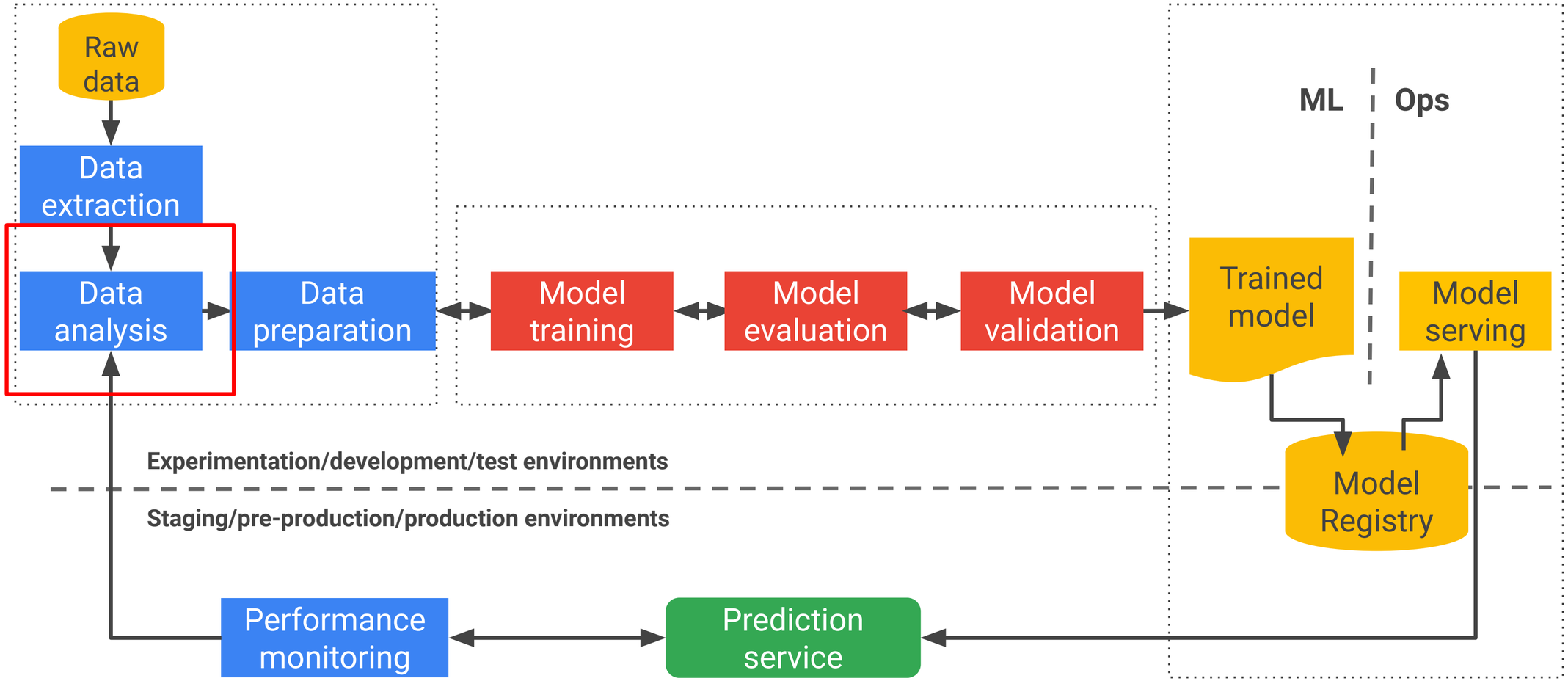
An ML Pipeline Recap



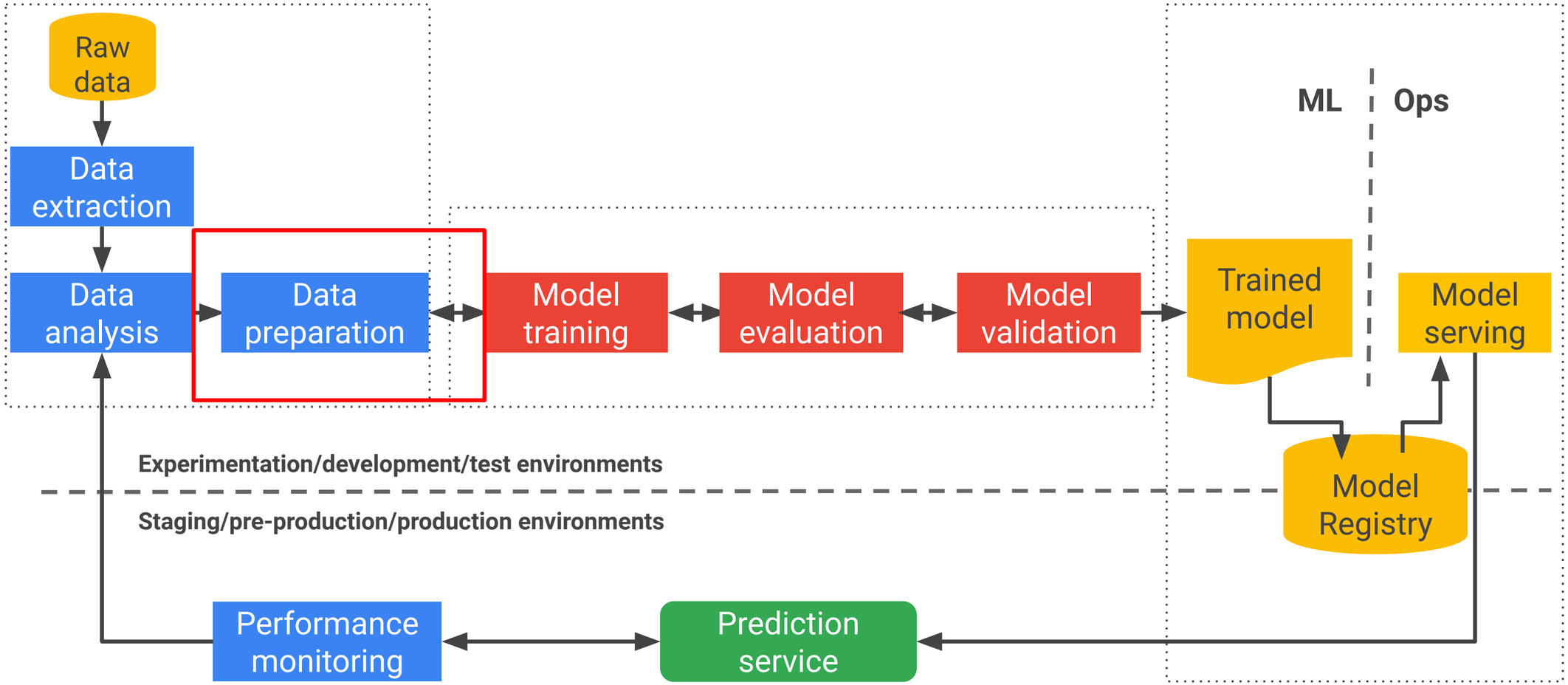
An ML Pipeline Recap



An ML Pipeline Recap

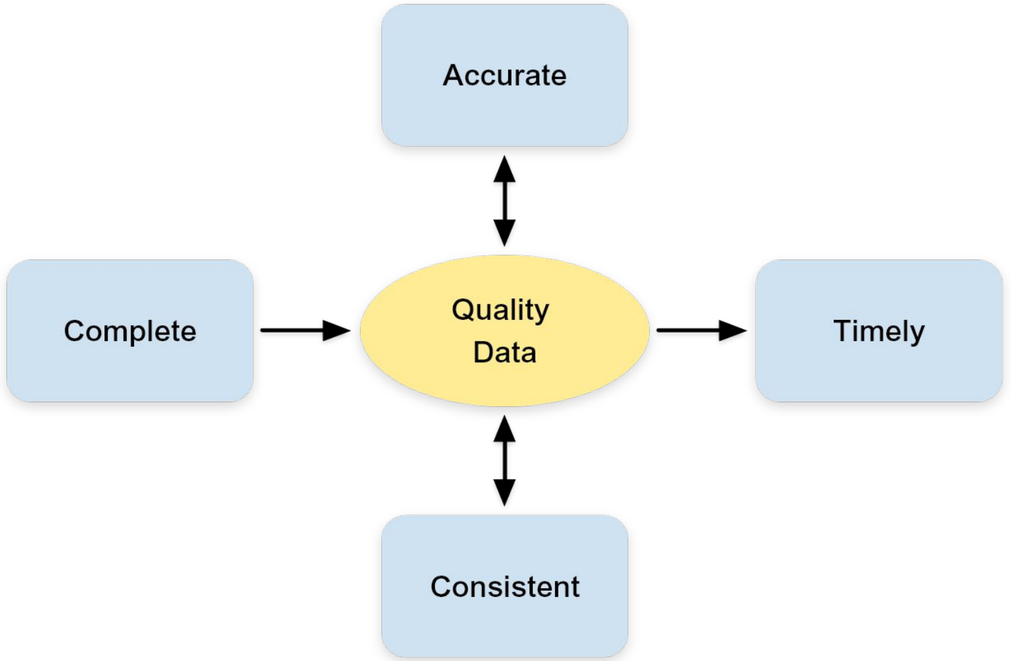


An ML Pipeline Recap



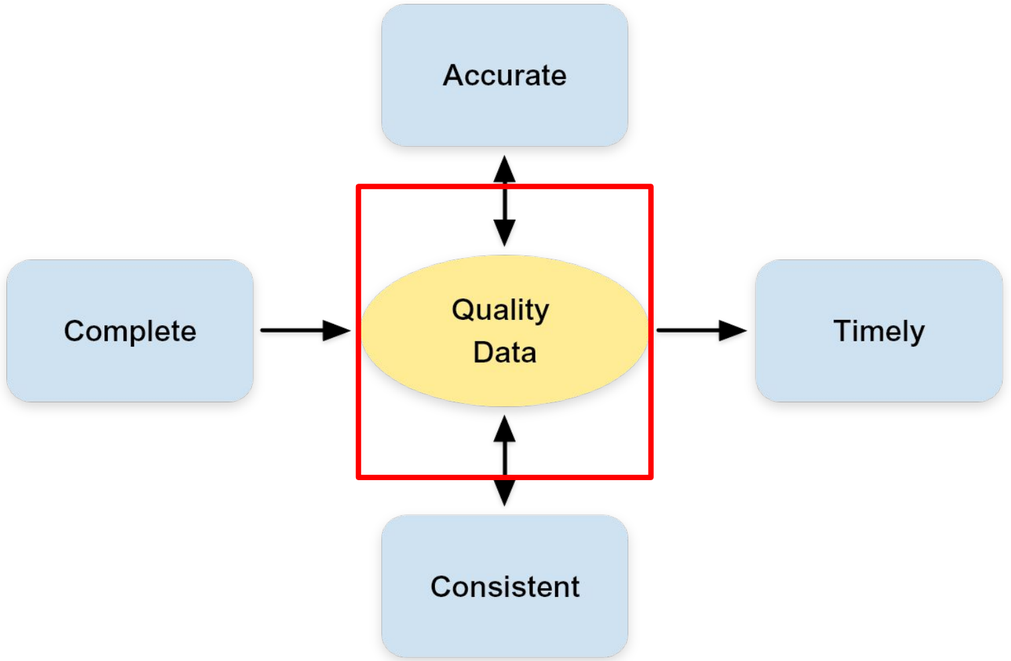
Attributes related to the Data Quality

1. Accuracy of Data
2. Consistency of Data
3. Timeliness of Data
4. Completeness of Data



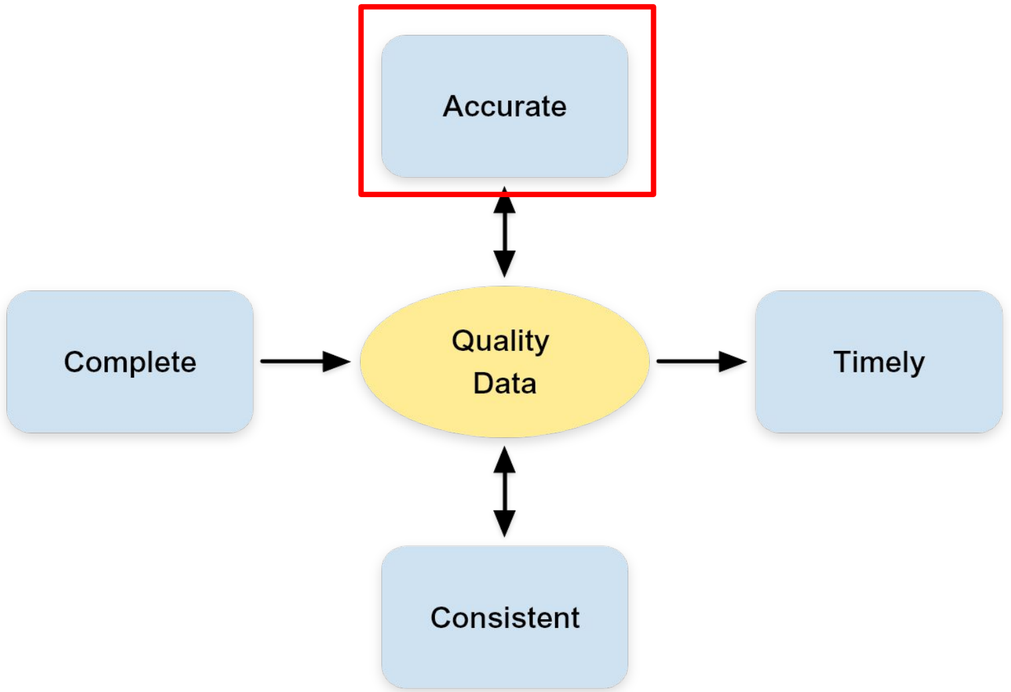
Attributes related to the Data Quality

1. Accuracy of Data
2. Consistency of Data
3. Timeliness of Data
4. Completeness of Data



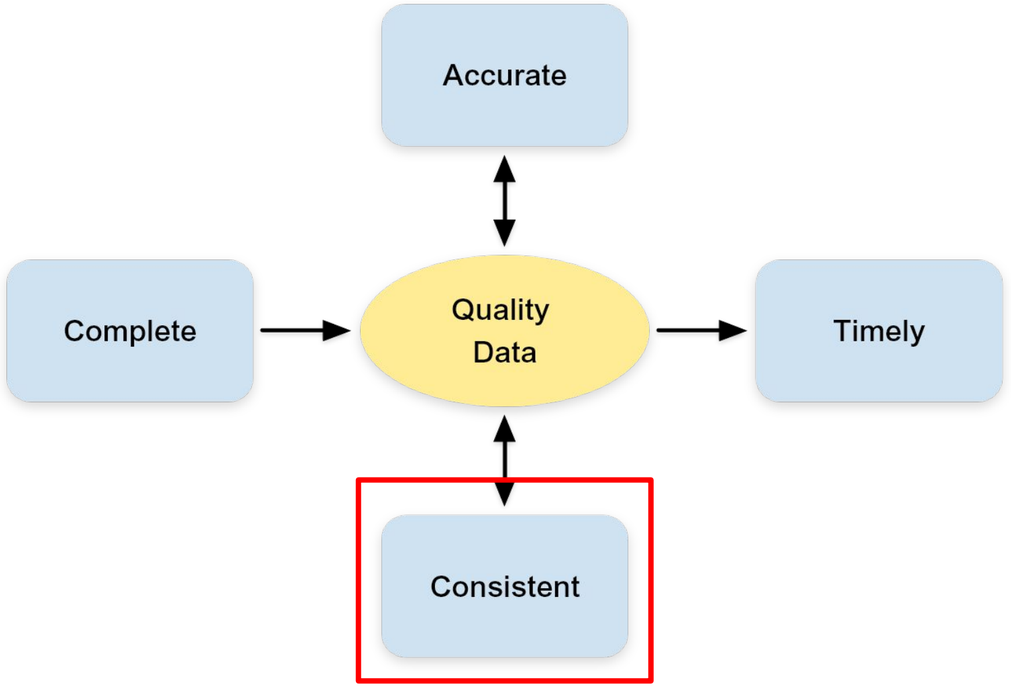
Attributes related to the Data Quality

1. Accuracy of Data
2. Consistency of Data
3. Timeliness of Data
4. Completeness of Data



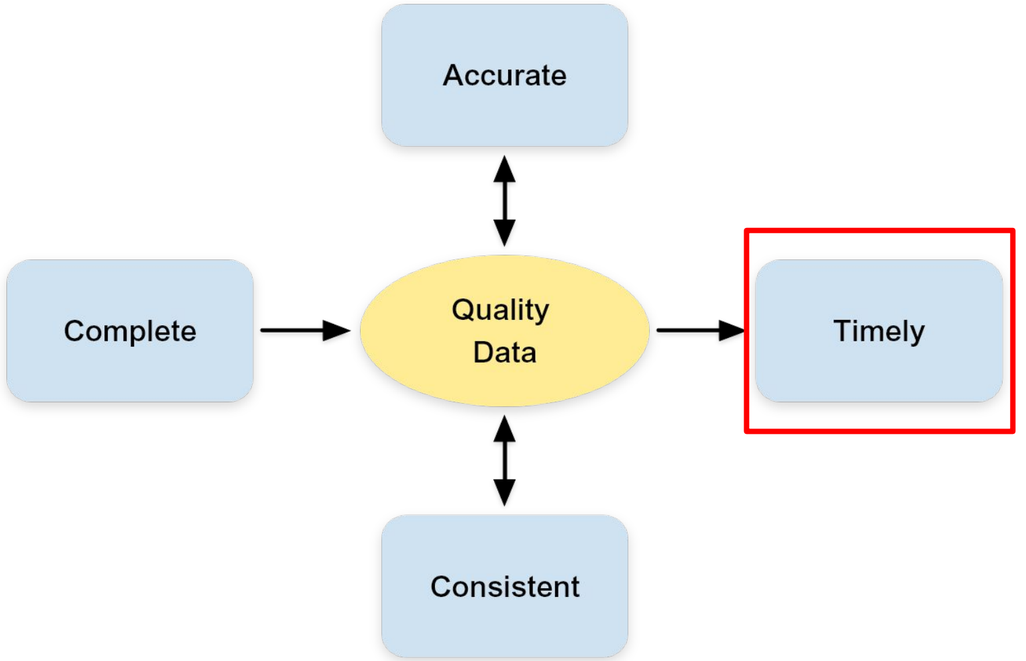
Attributes related to the Data Quality

1. Accuracy of Data
2. Consistency of Data
3. Timeliness of Data
4. Completeness of Data



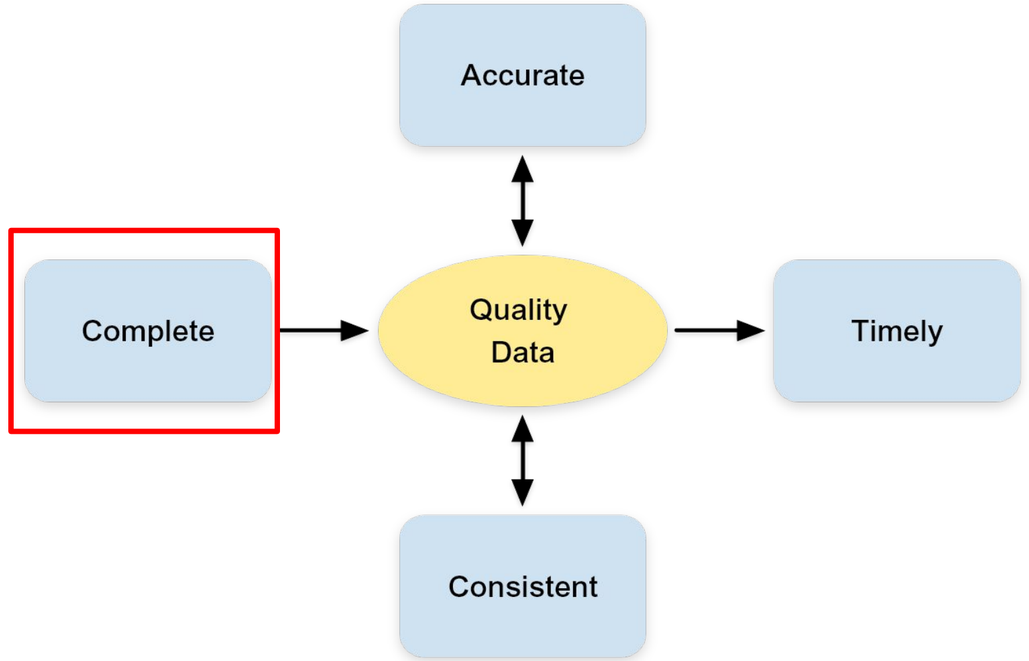
Attributes related to the Data Quality

1. Accuracy of Data
2. Consistency of Data
3. Timeliness of Data
4. Completeness of Data



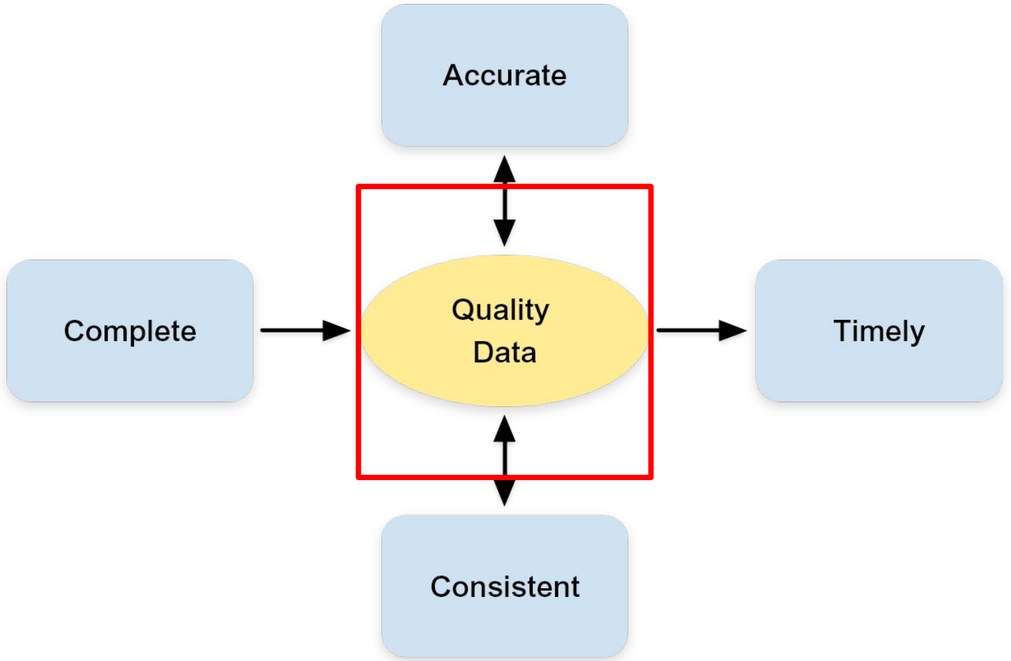
Attributes related to the Data Quality

1. Accuracy of Data
2. Consistency of Data
3. Timeliness of Data
4. Completeness of Data



Attributes related to the Data Quality

1. Accuracy of Data
2. Consistency of Data
3. Timeliness of Data
4. Completeness of Data



Ways to Improve Data Quality

1. Resolve Missing Values
2. Convert the Date feature column to Datetime Format
3. Parse date/time features
4. Remove unwanted values
5. Convert categorical columns to “one-hot encodings”

Missing Values

Let's show the null values for all features in the DataFrame.

In [10]:
```
df_transport.isnull().sum()
```

Out[10]:
```
Date          2
Zip Code      2
Model Year    2
Fuel          3
Make          3
Light_Duty    3
Vehicles      3
dtype: int64
```

Missing Values

Let's show the null values for all features in the DataFrame.

```
In [10]: df_transport.isnull().sum()

Out[10]: Date          2
         Zip Code      2
         Model Year    2
         Fuel          3
         Make          3
         Light_Duty    3
         Vehicles      3
         dtype: int64
```

Missing Values

Let's show the null values for all features in the DataFrame.

```
In [10]: df_transport.isnull().sum()

Out[10]: Date          2
         Zip Code      2
         Model Year    2
         Fuel          3
         Make          3
         Light_Duty    3
         Vehicles      3
         dtype: int64
```

Missing Values

```
In [11]: print (df_transport['Date'])
         print (df_transport['Date'].isnull())
```

```
0       10/1/2018
1       10/1/2018
2             NaN
3       10/1/2018
4       10/1/2018
          ...
994      6/7/2019
995      6/8/2019
996      6/9/2019
997     6/10/2019
998     6/11/2019
Name: Date, Length: 999, dtype: object
0       False
1       False
2        True
3       False
```

Missing Values

In [14]:
```
print ("Rows     : " ,df_transport.shape[0])
print ("Columns  : " ,df_transport.shape[1])
print ("\nFeatures : \n" ,df_transport.columns.tolist())
print ("\nUnique values :  \n",df_transport.nunique())
print ("\nMissing values :  ", df_transport.isnull().sum().values.sum())
```

```
Rows     :  999
Columns  :  7

Features :
 ['Date', 'Zip Code', 'Model Year', 'Fuel', 'Make', 'Light_Duty', 'Vehicles']

Unique values :
 Date          248
Zip Code         6
Model Year      15
Fuel             8
Make            43
Light_Duty       3
Vehicles       210
dtype: int64

Missing values :   18
```

Missing Values

In [14]:
```
print ("Rows     : " ,df_transport.shape[0])
print ("Columns  : " ,df_transport.shape[1])
print ("\nFeatures : \n" ,df_transport.columns.tolist())
print ("\nUnique values :  \n",df_transport.nunique())
print ("\nMissing values :  ", df_transport.isnull().sum().values.sum())
```

```
Rows     :  999
Columns  :  7

Features :
 ['Date', 'Zip Code', 'Model Year', 'Fuel', 'Make', 'Light_Duty', 'Vehicles']

Unique values :
 Date          248
Zip Code         6
Model Year      15
Fuel             8
Make            43
Light_Duty       3
Vehicles       210
dtype: int64

Missing values :   18
```

Missing Values = “Messy Data”

[5]:
```
df_transport = pd.read_csv('../data/transport/untidy_vehicle_data.csv')
df_transport.head() # Output the first five rows.
```

[5]:

	Date	Zip Code	Model Year	Fuel	Make	Light_Duty	Vehicles
0	10/1/2018	90000	2006	Gasoline	OTHER/UNK	NaN	1.0
1	10/1/2018	NaN	2014	Gasoline	NaN	Yes	1.0
2	NaN	90000	NaN	Gasoline	OTHER/UNK	Yes	NaN
3	10/1/2018	90000	2017	Gasoline	OTHER/UNK	Yes	1.0
4	10/1/2018	90000	<2006	Diesel and Diesel Hybrid	OTHER/UNK	No	55.0

Date and Time

```
In [6]: df_transport.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 999 entries, 0 to 998
Data columns (total 7 columns):
Date          997 non-null object
Zip Code      997 non-null object
Model Year    997 non-null object
Fuel          996 non-null object
Make          996 non-null object
Light_Duty    996 non-null object
Vehicles      996 non-null float64
dtypes: float64(1), object(6)
memory usage: 54.8+ KB
```

Convert Date and Time

Convert the Date Feature Column to a Datetime Format

The date column is indeed shown as a string object. We can convert it to the datetime datatype with the to_datetime() function in Pandas.

In [19]:
```
# TODO 2a
df_transport['Date'] =  pd.to_datetime(df_transport['Date'],
                                       format='%m/%d/%Y')
```

In [20]:
```
# TODO 2b
df_transport.info() # Date is now converted.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 999 entries, 0 to 998
Data columns (total 7 columns):
Date           999 non-null datetime64[ns]
Zip Code       999 non-null object
Model Year     999 non-null object
Fuel           999 non-null object
Make           999 non-null object
Light_Duty     999 non-null object
Vehicles       999 non-null float64
dtypes: datetime64[ns](1), float64(1), object(5)
memory usage: 54.8+ KB
```

Parse Date

Let's parse Date into three columns, e.g. year, month, and day.

In [21]:
```
df_transport['year'] = df_transport['Date'].dt.year
df_transport['month'] = df_transport['Date'].dt.month
df_transport['day'] = df_transport['Date'].dt.day
#df['hour'] = df['date'].dt.hour - you could use this if your
#df['minute'] = df['date'].dt.minute - you could use this if y
df_transport.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 999 entries, 0 to 998
Data columns (total 10 columns):
Date          999 non-null datetime64[ns]
Zip Code      999 non-null object
Model Year    999 non-null object
Fuel          999 non-null object
Make          999 non-null object
Light_Duty    999 non-null object
Vehicles      999 non-null float64
year          999 non-null int64
month         999 non-null int64
day           999 non-null int64
dtypes: datetime64[ns](1), float64(1), int64(3), object(5)
memory usage: 78.2+ KB
```

Unwanted Characters - Model Year

Let's investigate a bit more of our data by using the .groupby() function.

In [9]:
```
grouped_data = df_transport.groupby(['Zip Code','Model Year','Fuel','Make','Light_Duty','Vehicles'])
df_transport.groupby('Fuel').first() # Get the first entry for each month.
```

Out[9]:

	Date	Zip Code	Model Year	Make	Light_Duty	Vehicles
Fuel						
Battery Electric	10/1/2018	90000	<2006	OTHER/UNK	No	4.0
Diesel and Diesel Hybrid	10/1/2018	90000	<2006	OTHER/UNK	No	55.0
Flex-Fuel	10/14/2018	90001	2007	Type_A	Yes	78.0
Gasoline	10/1/2018	90000	2006	OTHER/UNK	Yes	1.0
Hybrid Gasoline	10/24/2018	90001	2009	OTHER/UNK	Yes	18.0
Natural Gas	10/25/2018	90001	2009	OTHER/UNK	No	2.0
Other	10/8/2018	90000	<2006	OTHER/UNK	Yes	6.0
Plug-in Hybrid	11/2/2018	90001	2012	OTHER/UNK	Yes	1.0

Categorical Data - “Yes/No”

[5]:
```
df_transport = pd.read_csv('../data/transport/untidy_vehicle_data.csv')
df_transport.head() # Output the first five rows.
```

[5]:

	Date	Zip Code	Model Year	Fuel	Make	Light_Duty	Vehicles
0	10/1/2018	90000	2006	Gasoline	OTHER/UNK	NaN	1.0
1	10/1/2018	NaN	2014	Gasoline	NaN	Yes	1.0
2	NaN	90000	NaN	Gasoline	OTHER/UNK	Yes	NaN
3	10/1/2018	90000	2017	Gasoline	OTHER/UNK	Yes	1.0
4	10/1/2018	90000	<2006	Diesel and Diesel Hybrid	OTHER/UNK	No	55.0

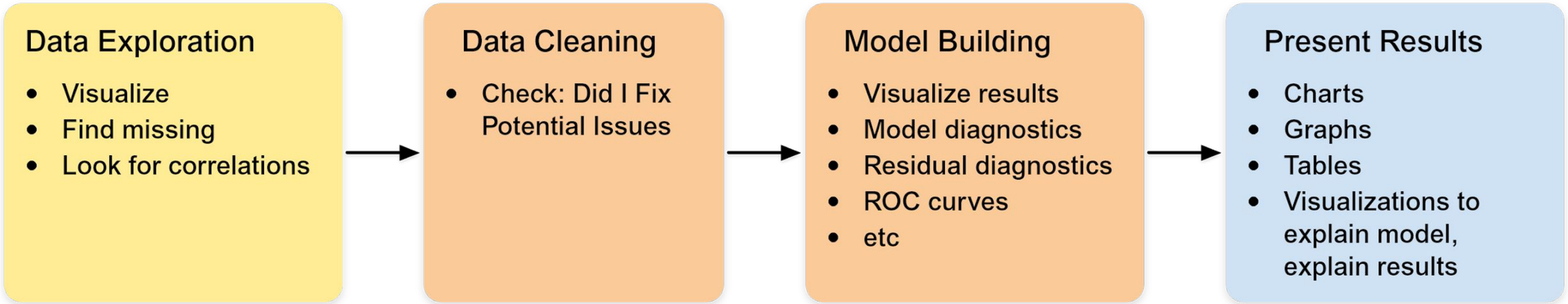
Categorical features

Ocean_Proximity
INLAND
NEAR BAY
OCEAN

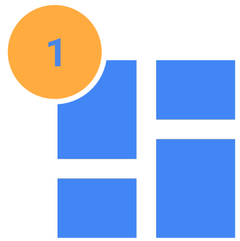
One-Hot Encoding

Ocean_Proximity=INLAND	Ocean_Proximity=NEAR BAY	Ocean_Proximity=OCEAN
1	0	0
0	1	0
0	0	1

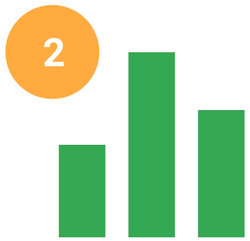
Data Quality



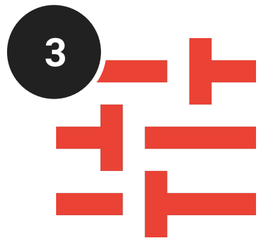
Machine learning is a way to use standard **algorithms** to derive **predictive insights** from **data** and make **repeated decisions**.



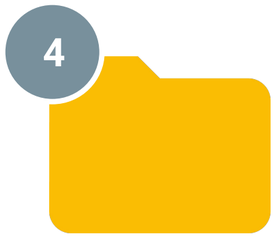
Algorithm



Data

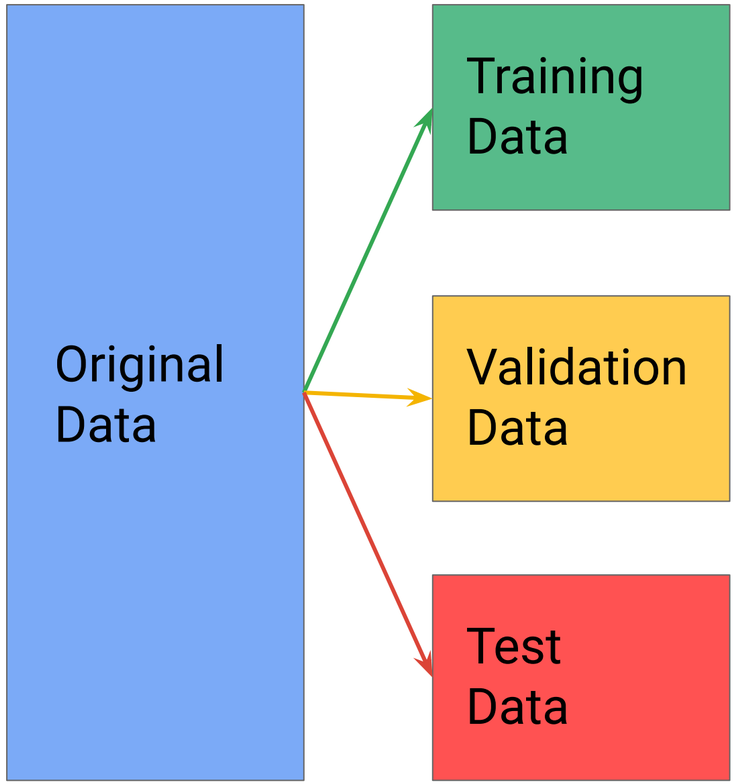


Predictive insight

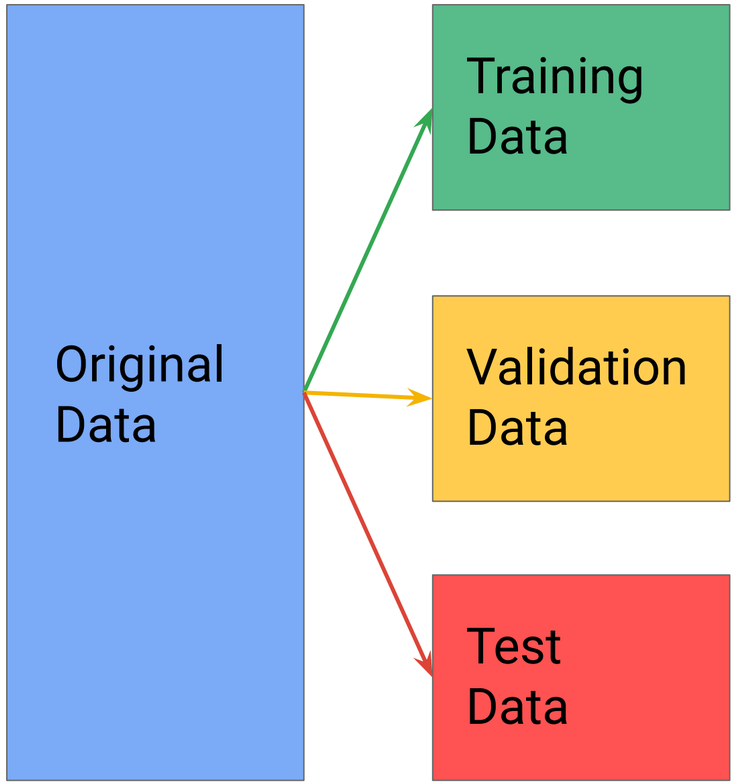


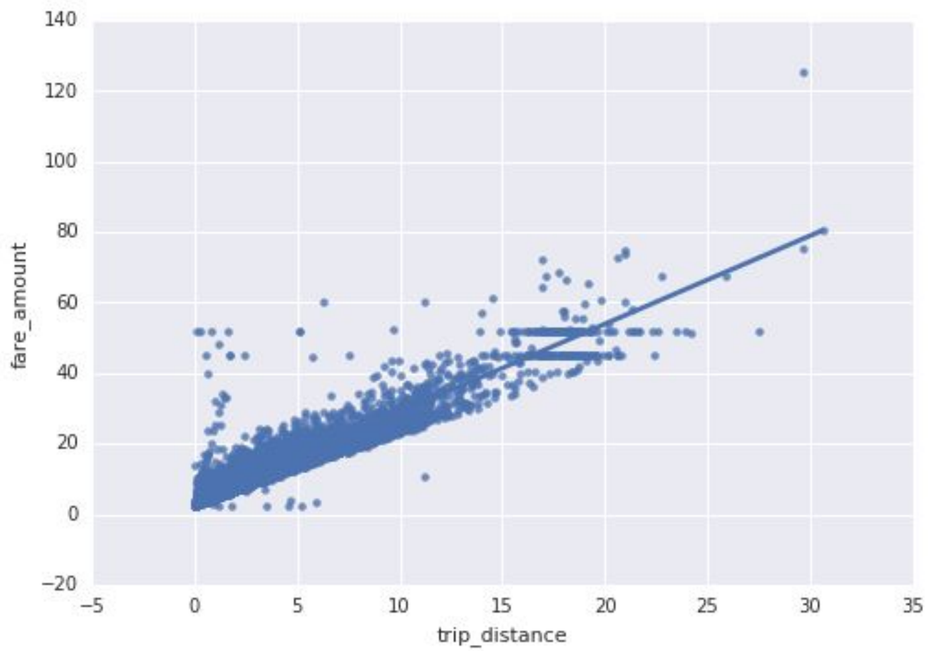
Decision

The Importance of Data Quality



The Importance of Data Quality





Launching into ML:
Exploratory Data Analysis

Agenda

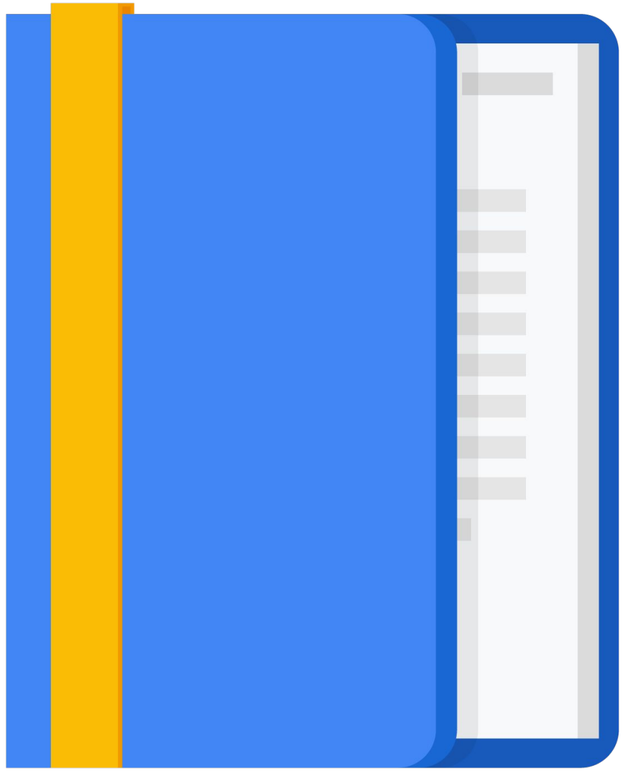
What is Exploratory Data Analysis?

How is EDA used in Machine Learning?

Data Analysis and Visualization

Lab: Improve the Quality of Data

Lab: Explore the data using Python and BigQuery

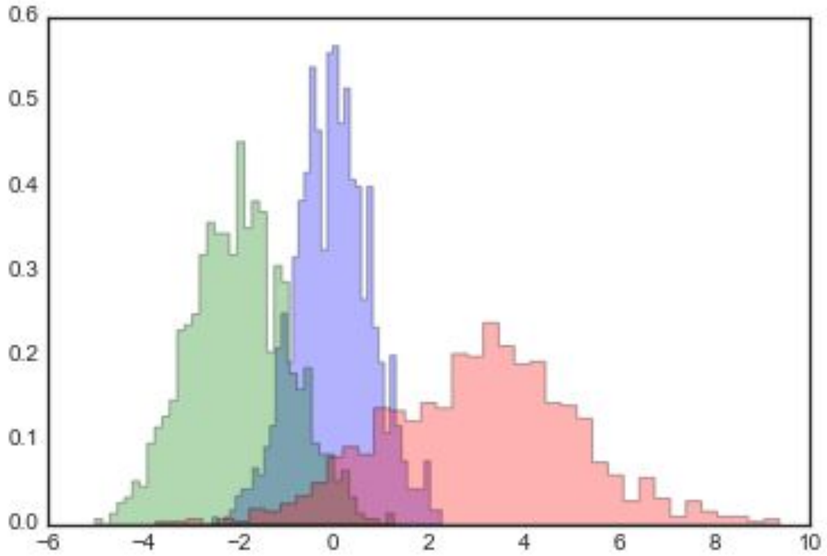


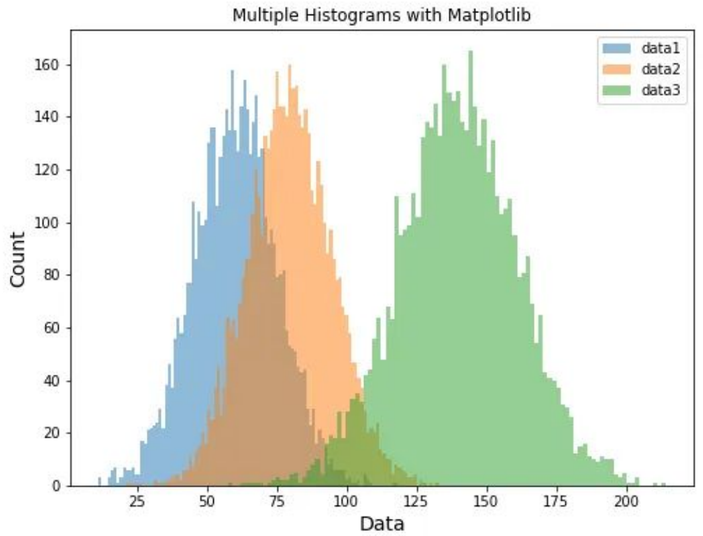
Exploratory Data Analysis

What is Exploratory Data Analysis?

How is EDA used in Machine Learning?

Data Analysis and Visualization.





Exploratory Data Analysis

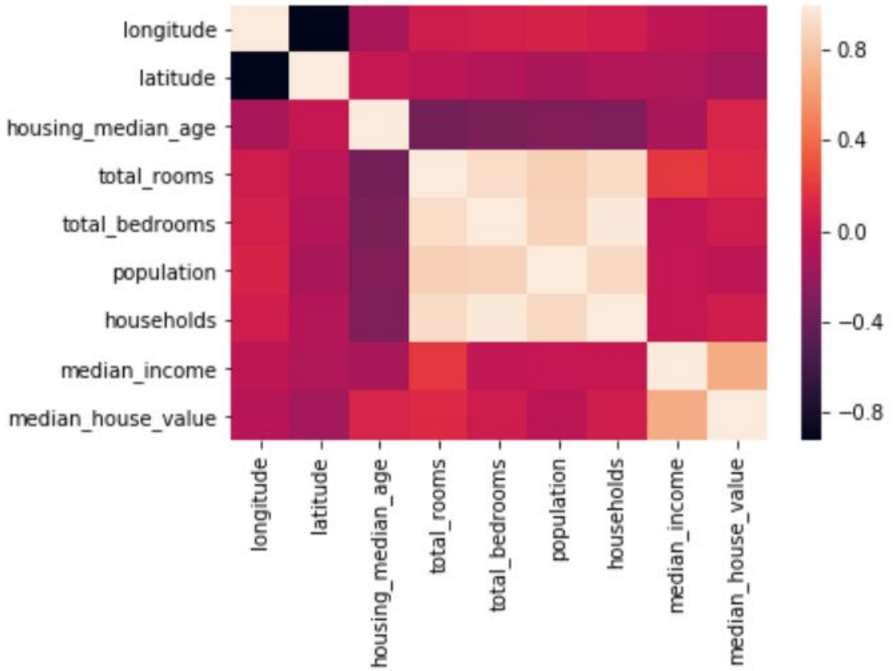
What is Exploratory Data Analysis?

How is EDA used in Machine Learning?

Data Analysis and Visualization.

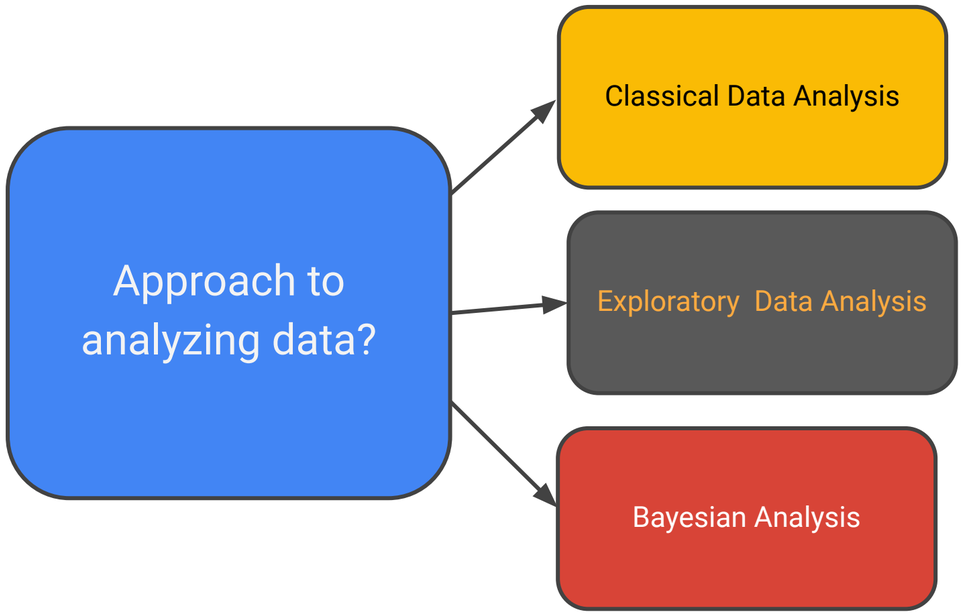
```
[12]: sns.heatmap(df_USAhousing.corr())
```

```
[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7f25ba7c1dd8>
```



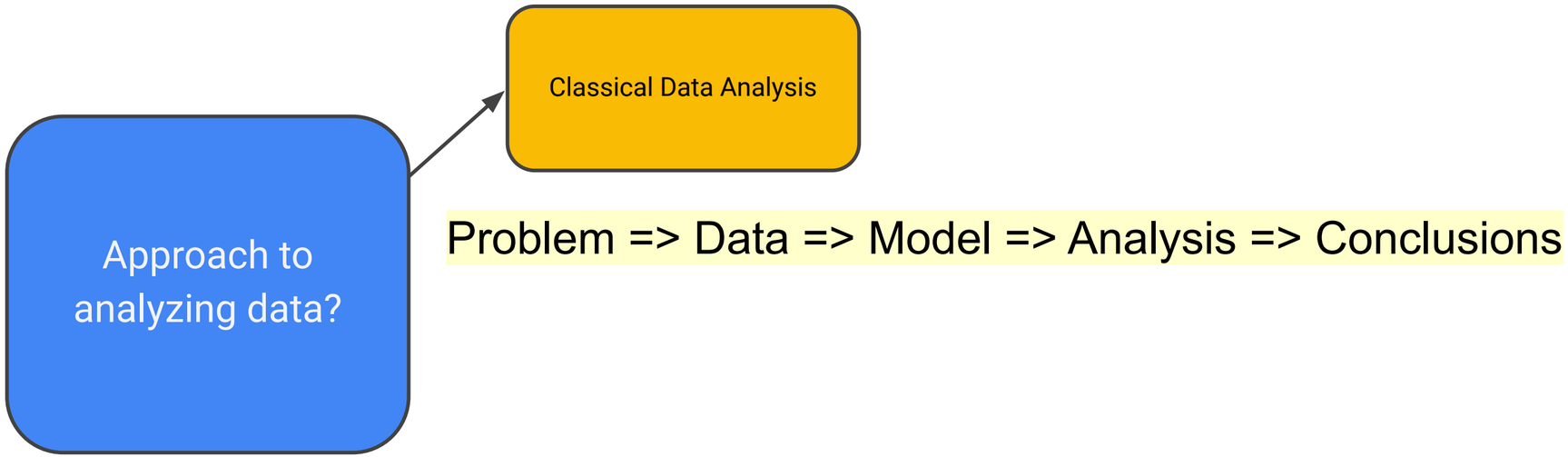
CDA vs. EDA vs. Bayesian?

What other approaches exist and how does Exploratory Data Analysis differ from these other approaches?



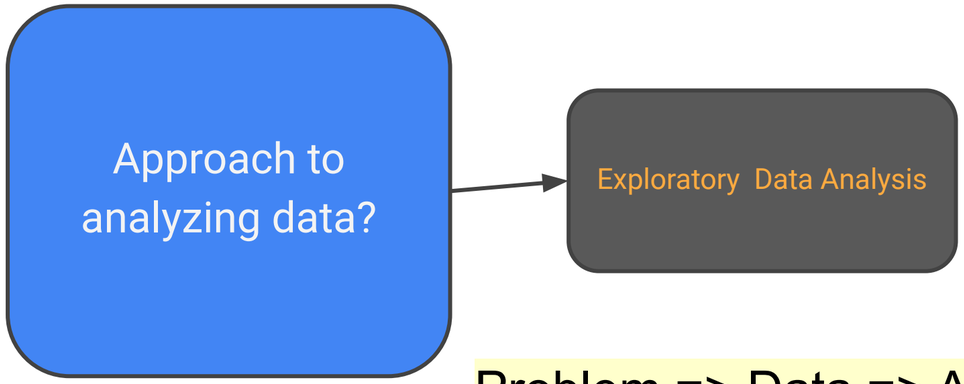
CDA vs. EDA vs. Bayesian?

What other approaches exist and how does Exploratory Data Analysis differ from these other approaches?



CDA vs. EDA vs. Bayesian?

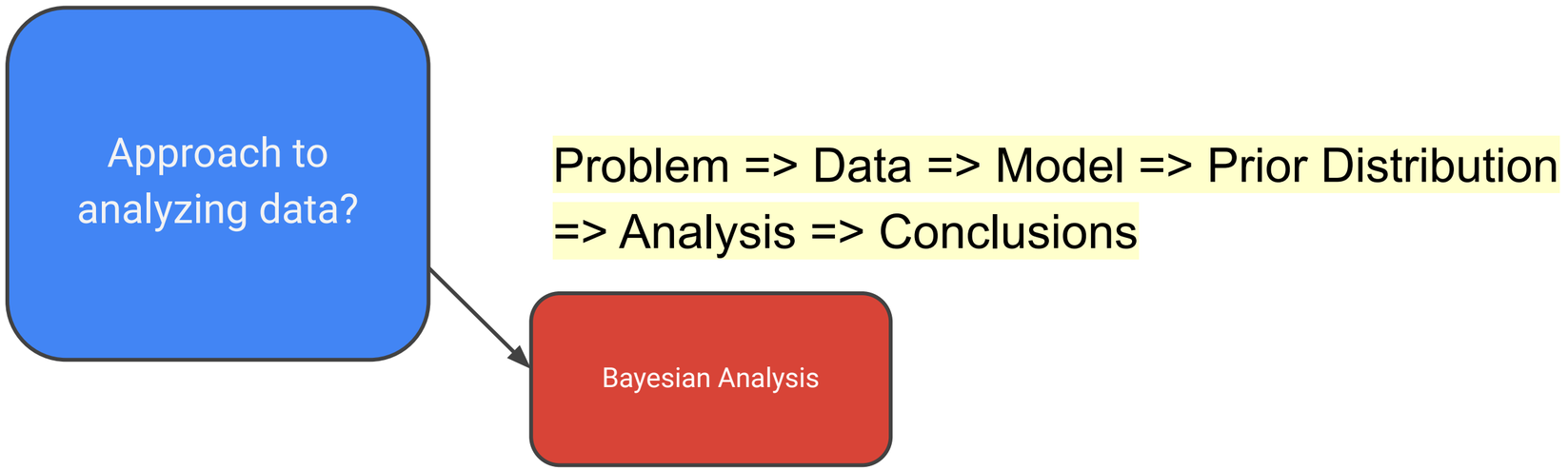
What other approaches exist and how does Exploratory Data Analysis differ from these other approaches?



Problem => Data => Analysis => Model => Conclusions

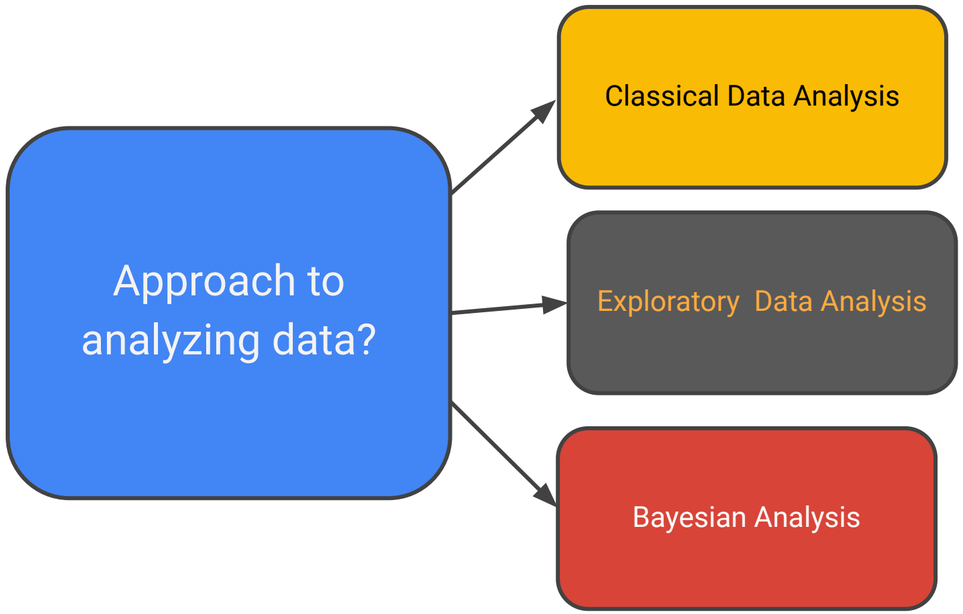
CDA vs. EDA vs. Bayesian?

What other approaches exist and how does Exploratory Data Analysis differ from these other approaches?



CDA vs. EDA vs. Bayesian?

What other approaches exist and how does Exploratory Data Analysis differ from these other approaches?



Agenda

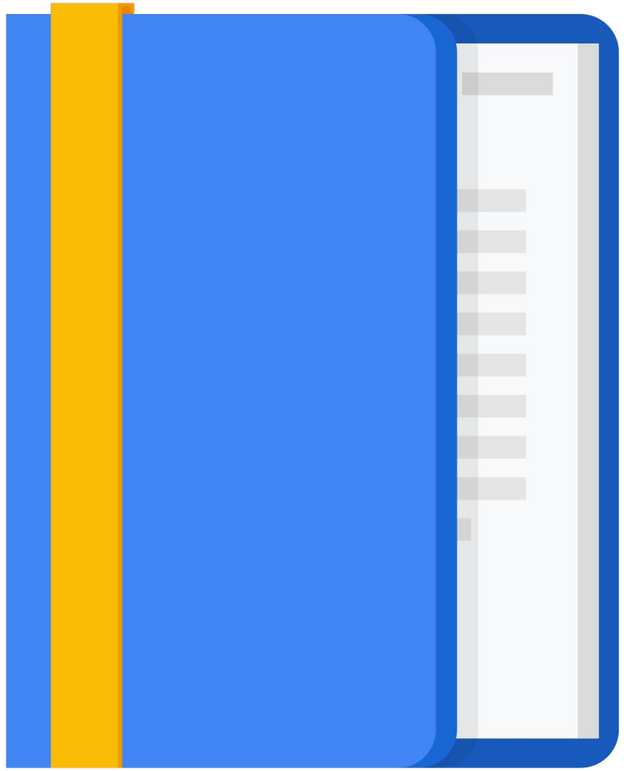
What is Exploratory Data Analysis?

How is EDA used in Machine Learning?

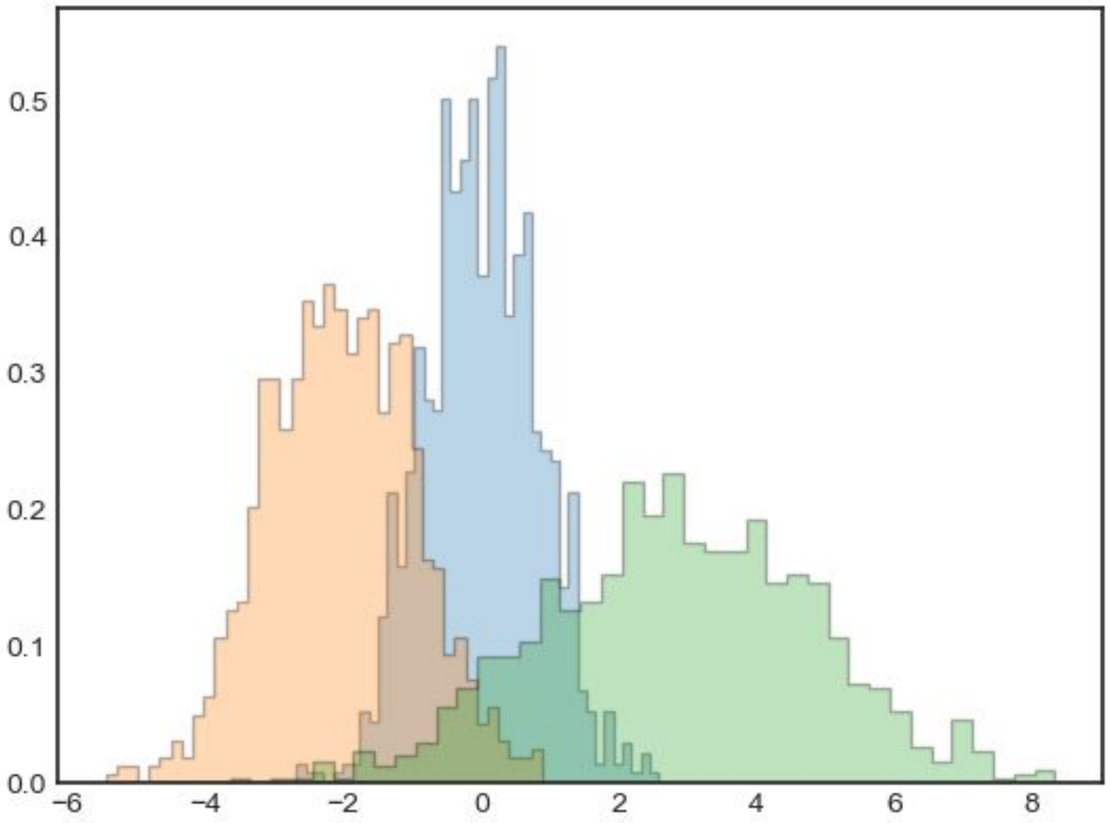
Data Analysis and Visualization

Lab: Improve the Quality of Data

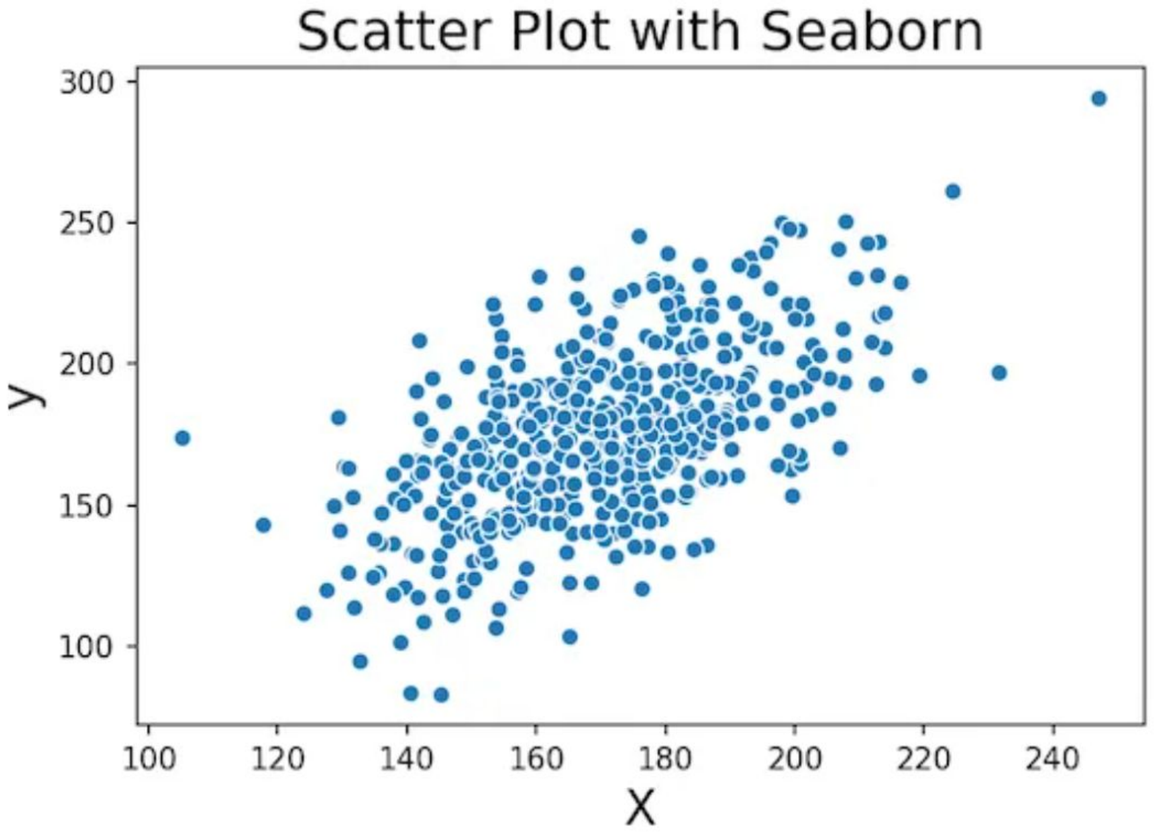
Lab: Explore the data using Python and BigQuery



How is EDA used in Machine Learning?

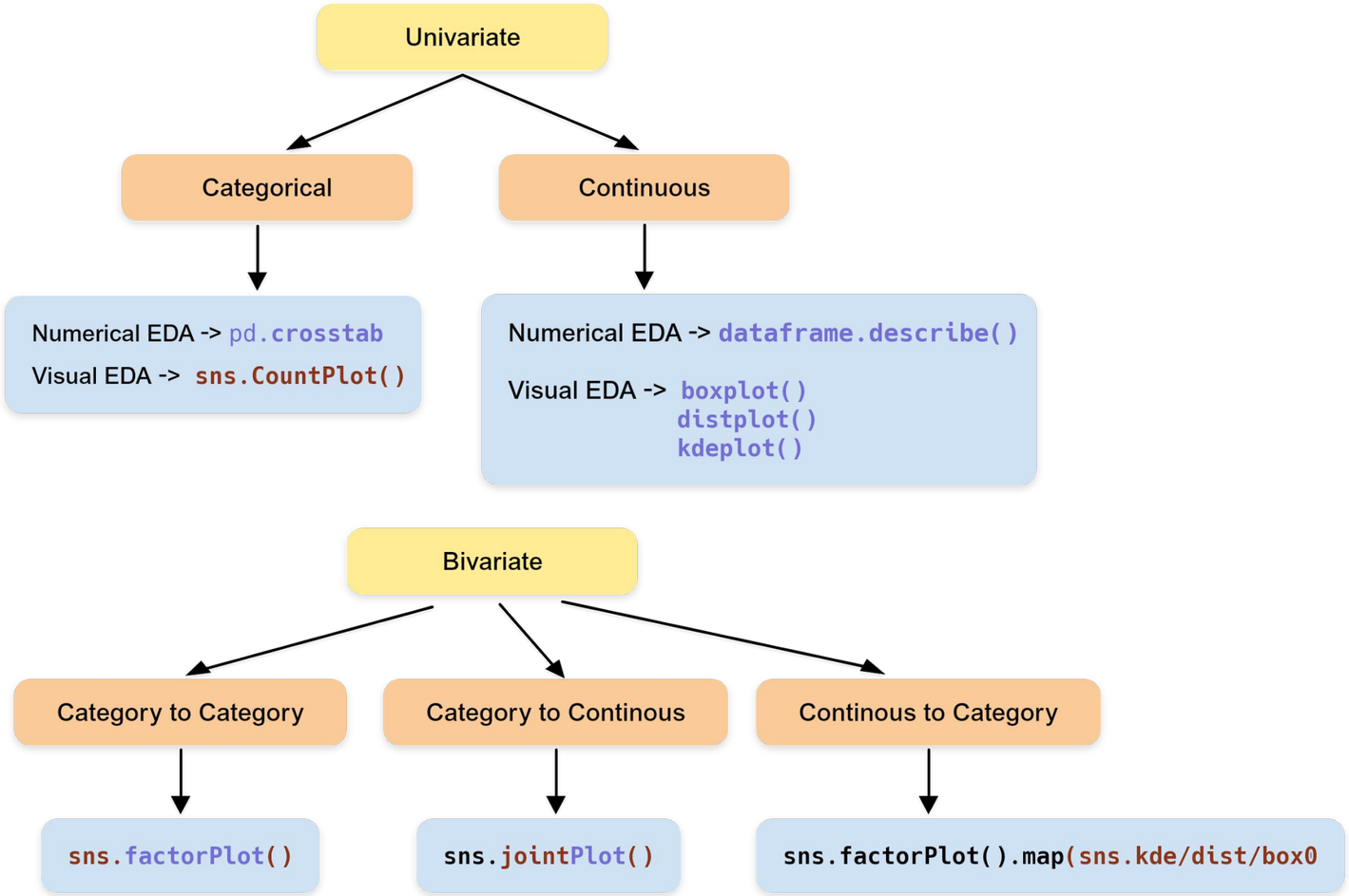


How is EDA used in Machine Learning?

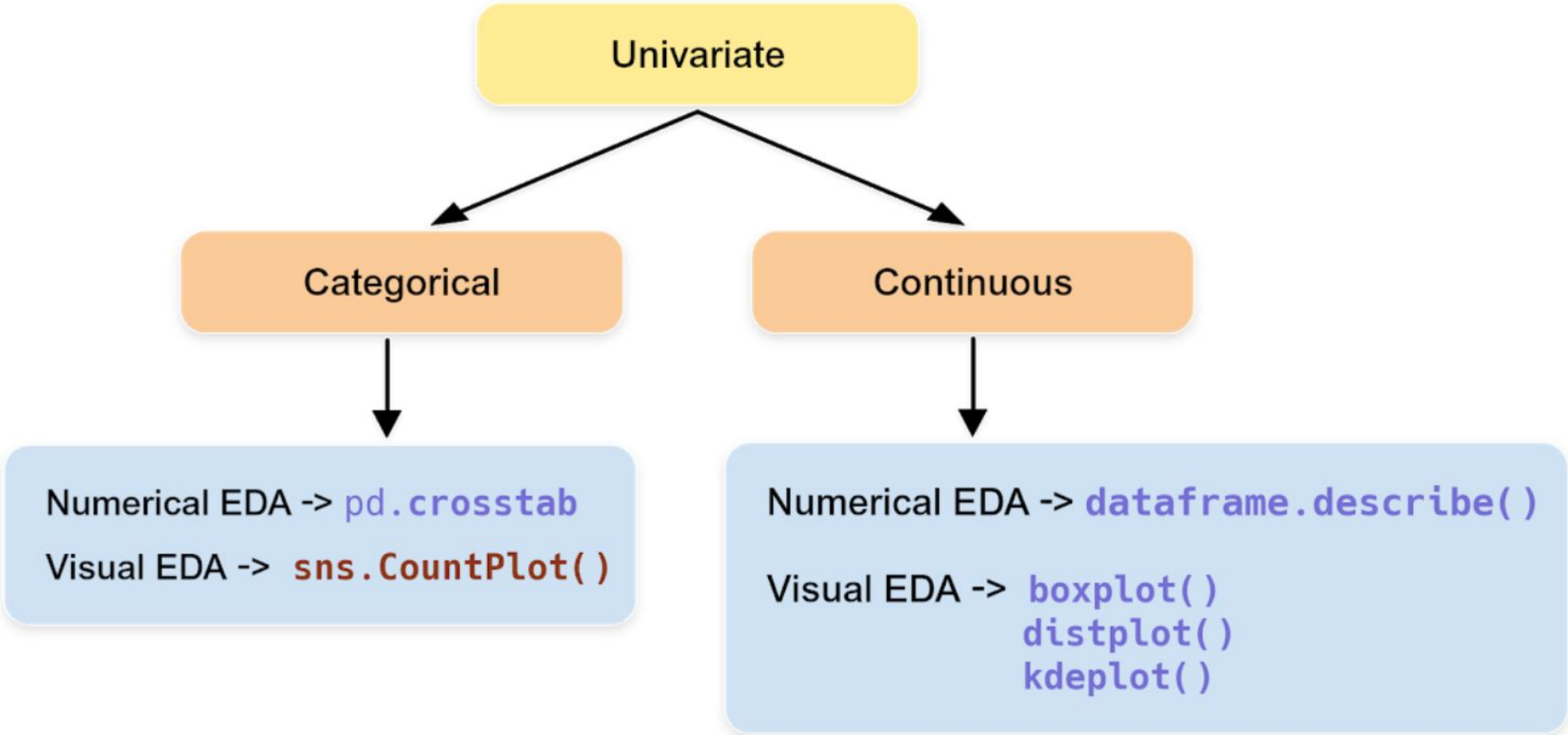


Scatter Plot Seaborn

Exploratory Data Analysis



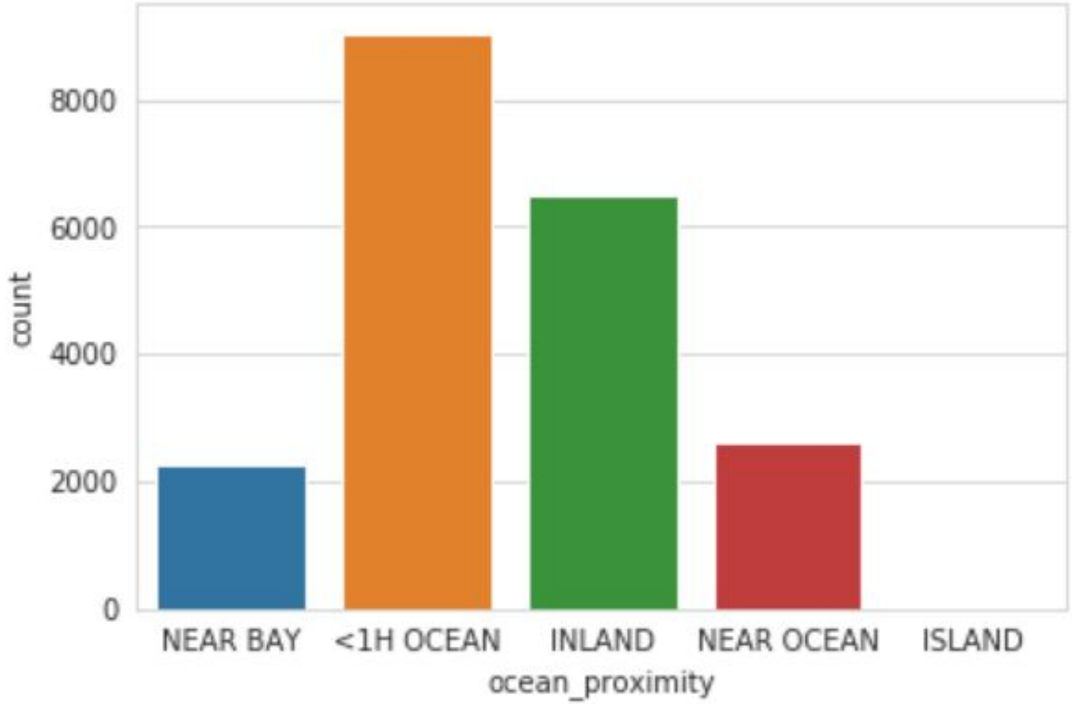
Exploratory Data Analysis



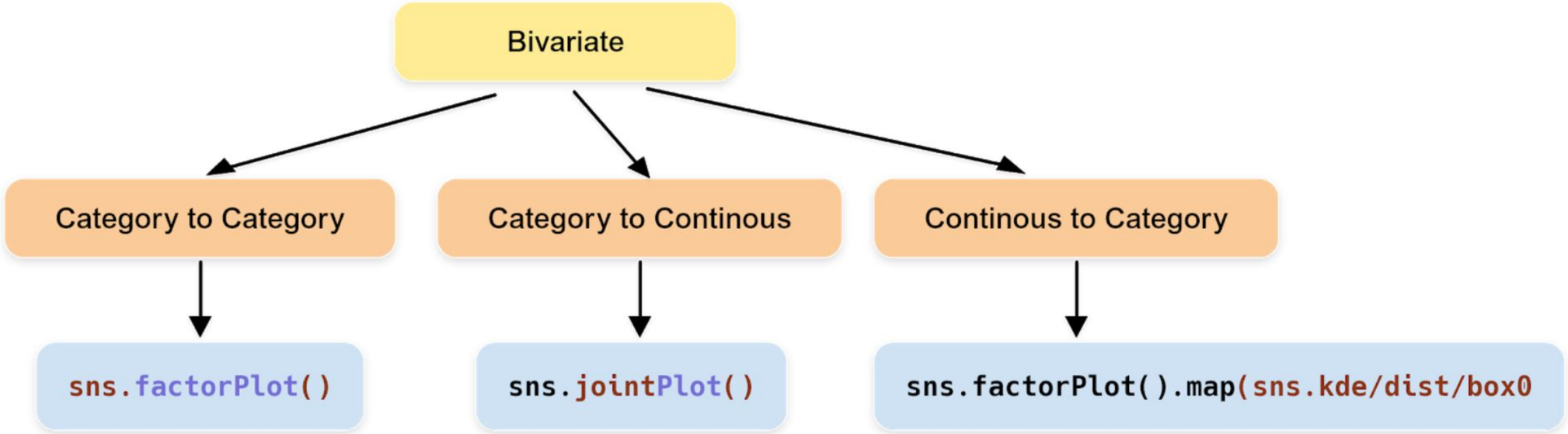
Univariate Data

```
[17]: sns.countplot(x = 'ocean_proximity', data=df_USAhousing)
```

```
[17]: <matplotlib.axes._subplots.AxesSubplot at 0x7f25ba4ef400>
```

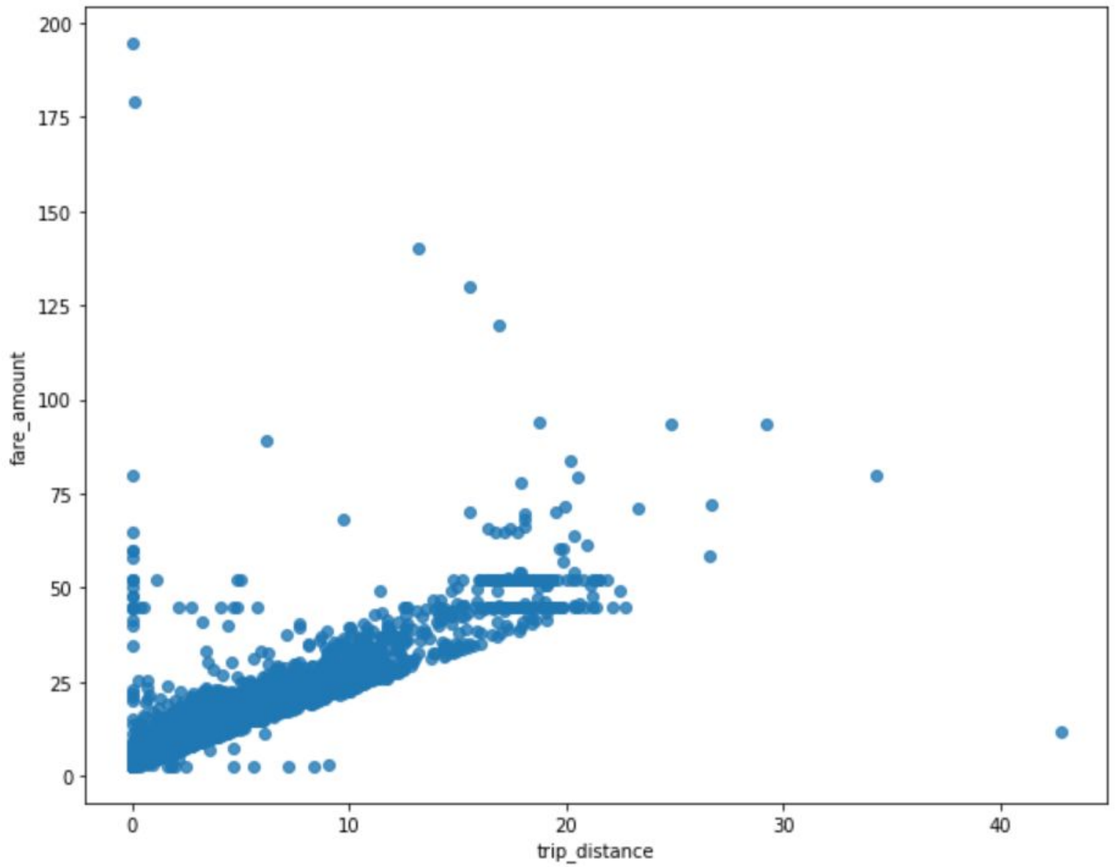


Exploratory Data Analysis



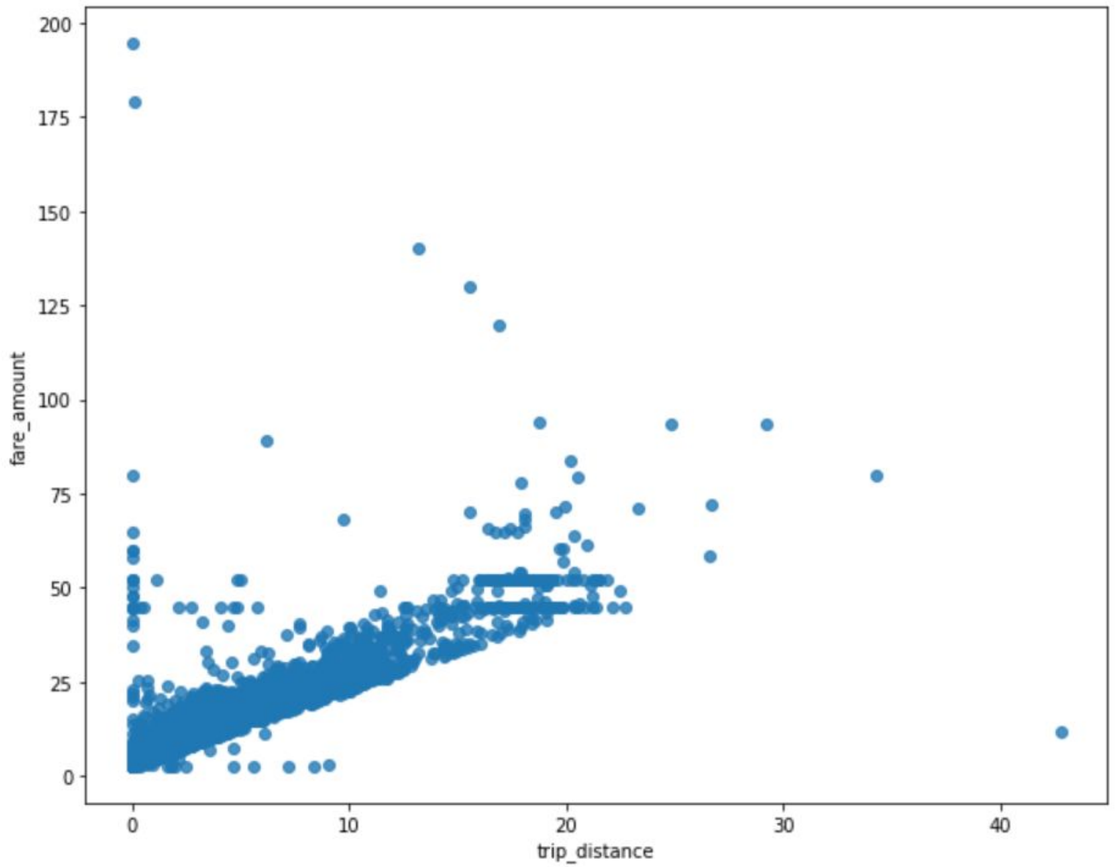
Bivariate Data

```
# TODO 2
ax = sns.regplot(
    x="trip_distance", y="fare_amount",
    fit_reg=False, ci=None, truncate=True, data=trips)
ax.figure.set_size_inches(10, 8)
```



Bivariate Data

```
# TODO 2
ax = sns.regplot(
    x="trip_distance", y="fare_amount",
    fit_reg=False, ci=None, truncate=True, data=trips)
ax.figure.set_size_inches(10, 8)
```



Agenda

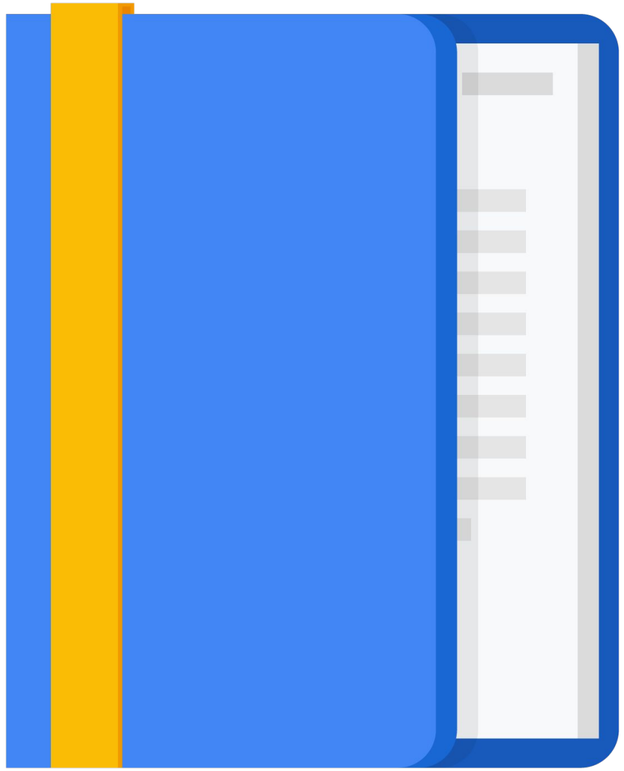
What is Exploratory Data Analysis?

How is EDA used in Machine Learning?

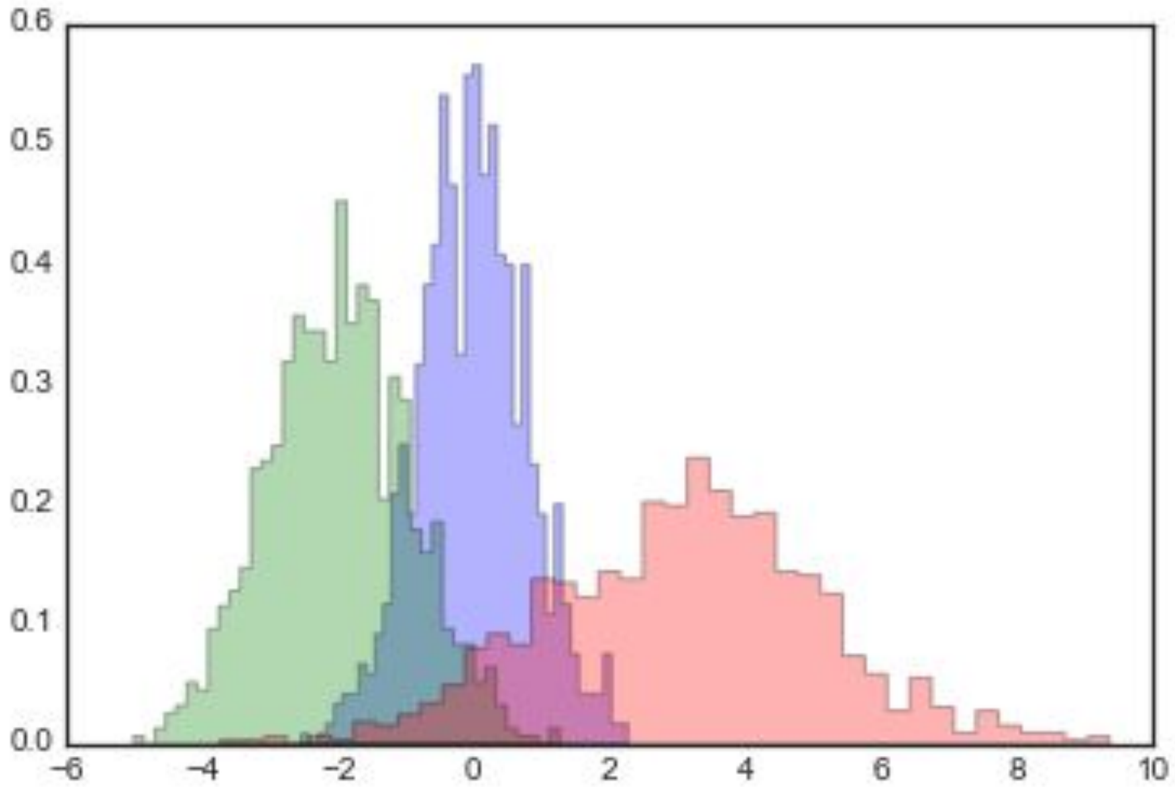
Data Analysis and Visualization

Lab: Improve the Quality of Data

Lab: Explore the data using Python and BigQuery



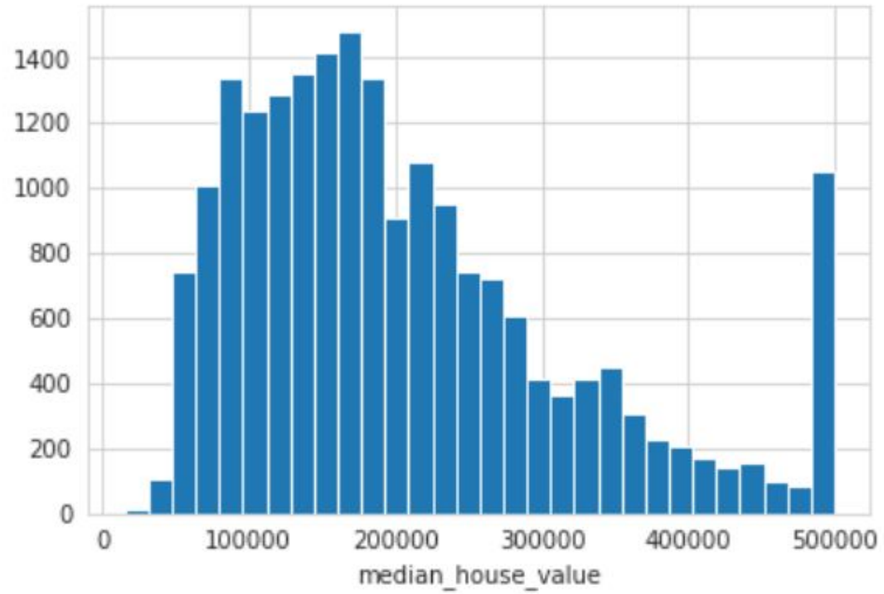
Data analysis and visualization



Histogram

[14]:
```
sns.set_style('whitegrid')
df_USAhousing['median_house_value'].hist(bins=30)
plt.xlabel('median_house_value')
```

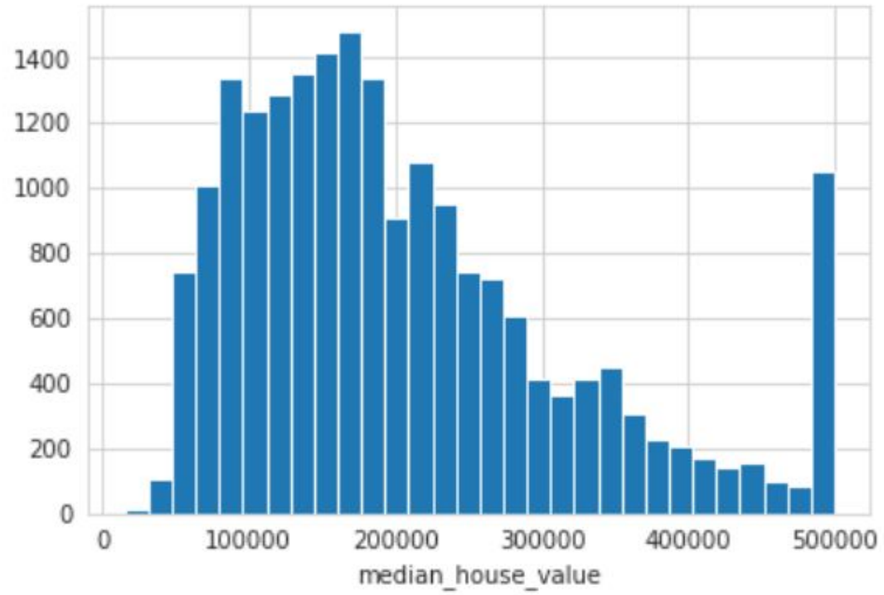
[14]:
```
Text(0.5, 0, 'median_house_value')
```



Histogram

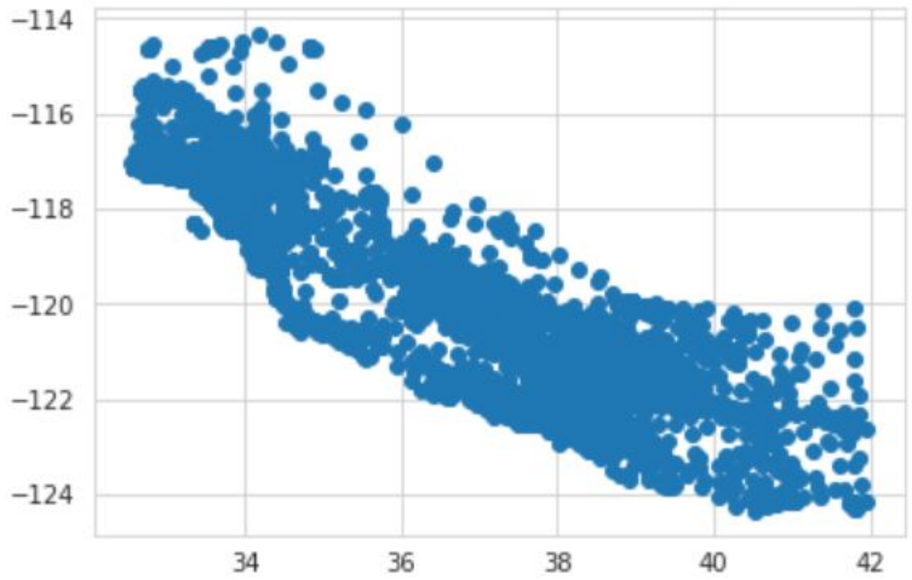
[14]:
```
sns.set_style('whitegrid')
df_USAhousing['median_house_value'].hist(bins=30)
plt.xlabel('median_house_value')
```

[14]:
```
Text(0.5, 0, 'median_house_value')
```



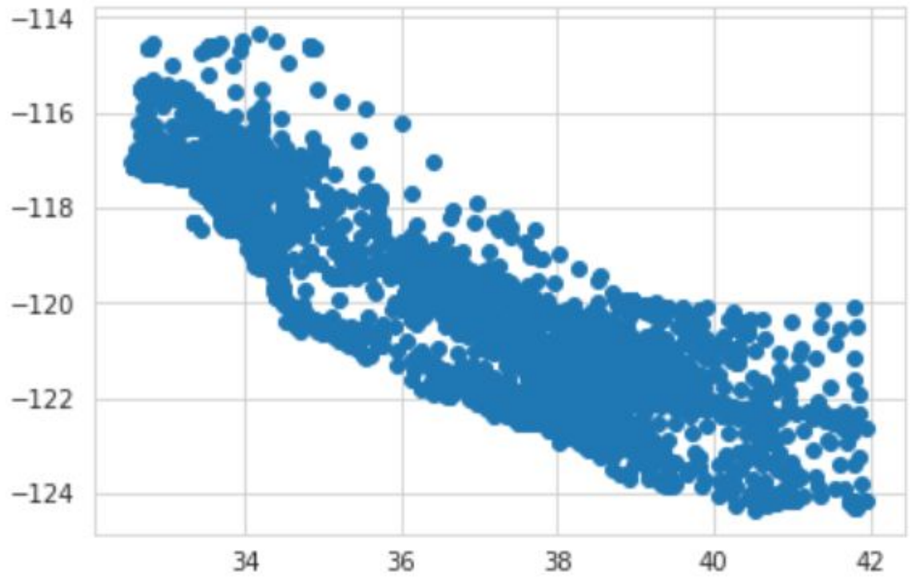
Scatter Plot

```
x = df_USAhousing['latitude']
y = df_USAhousing['longitude']

plt.scatter(x, y)
plt.show()
```



Scatter Plot

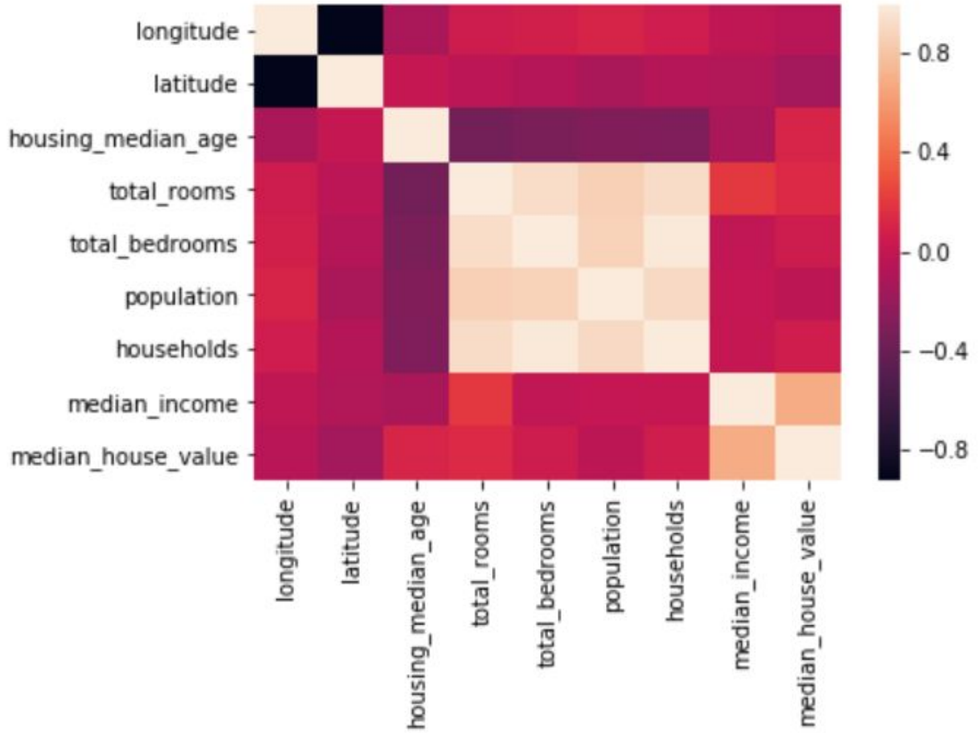
```
x = df_USAhousing['latitude']
y = df_USAhousing['longitude']

plt.scatter(x, y)
plt.show()
```



Correlations

```
[12]: sns.heatmap(df_USAhousing.corr())
```

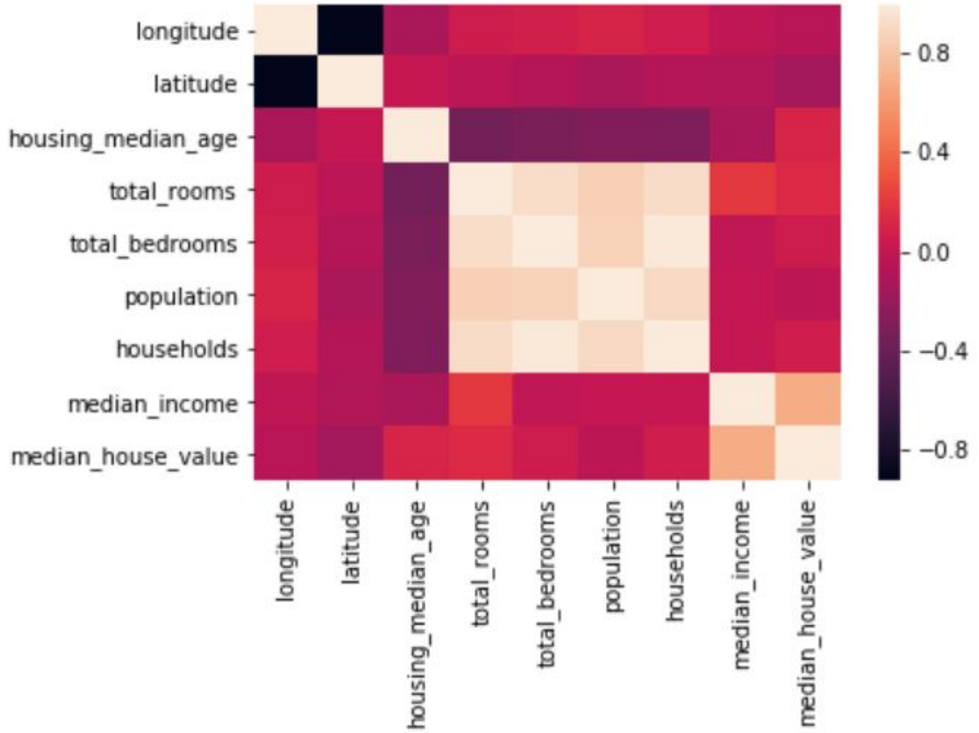
```
[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7f25ba7c1dd8>
```



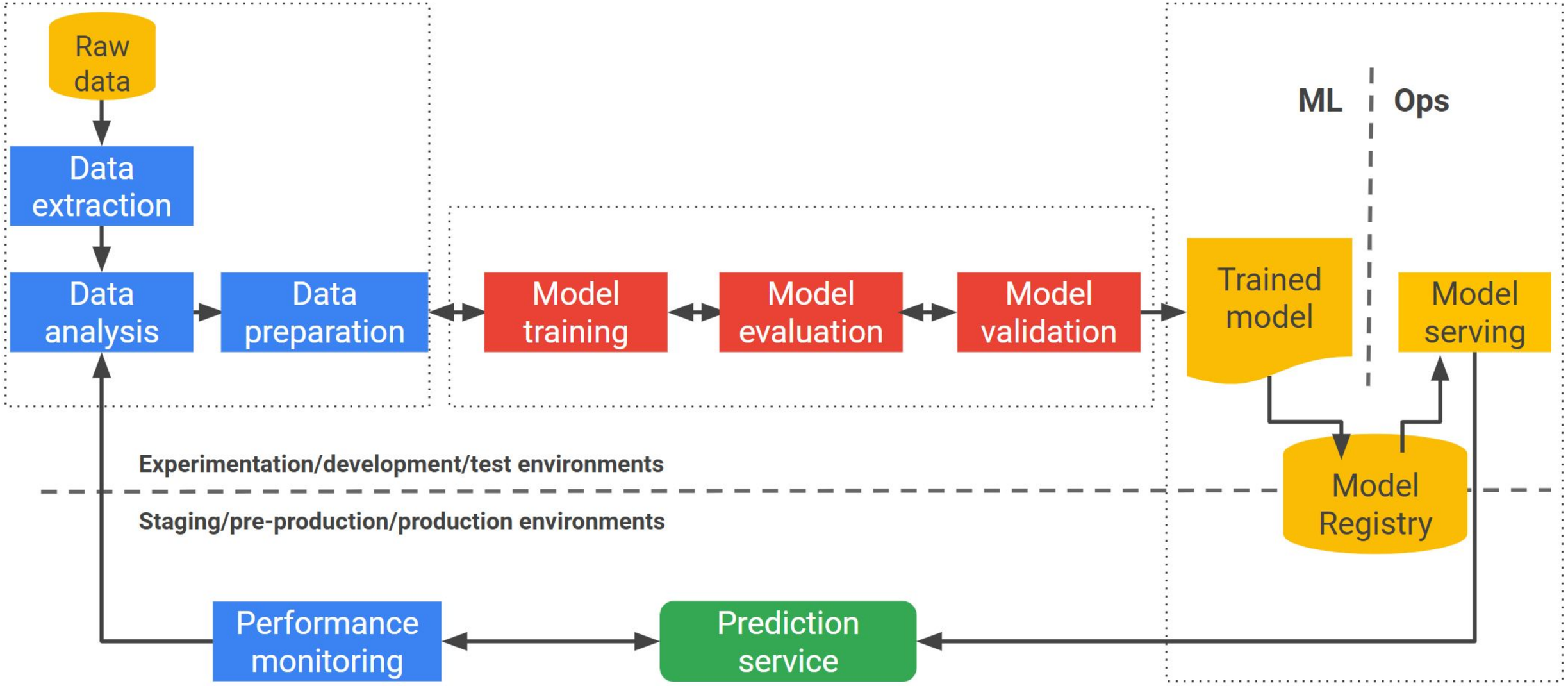
Correlations - Multivariate

```
[12]: sns.heatmap(df_USAhousing.corr())
```

```
[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7f25ba7c1dd8>
```



An ML Pipeline Recap



Agenda

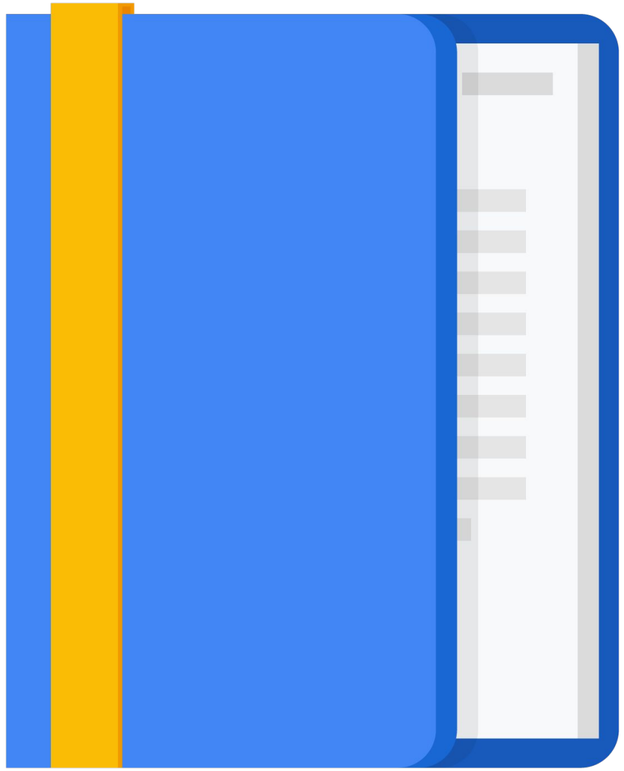
What is Exploratory Data Analysis?

How is EDA used in Machine Learning?

Data Analysis and Visualization

Lab: Improve the Quality of Data

Lab: Explore the data using Python and BigQuery



Lab Intro

Improve the Quality of Data

Google Cloud

Agenda

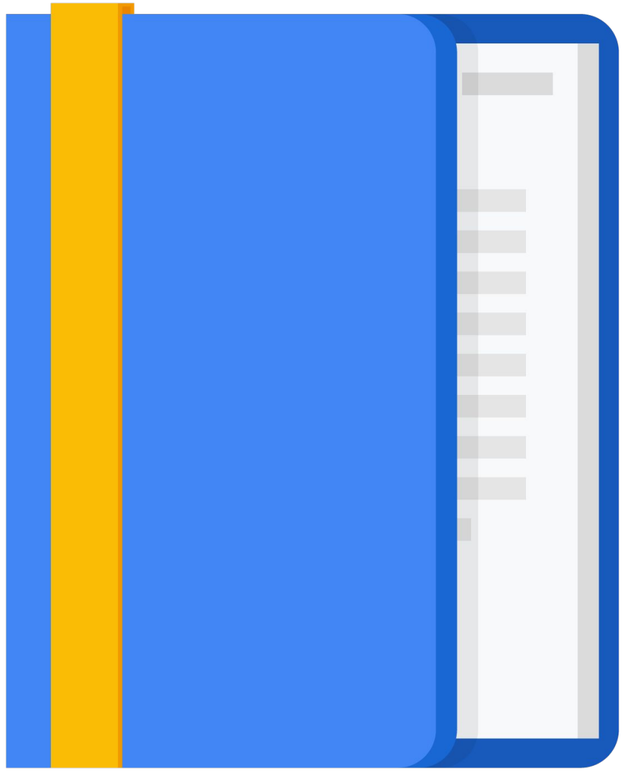
What is Exploratory Data Analysis?

How is EDA used in Machine Learning?

Data Analysis and Visualization

Lab: Improve the Quality of Data

Lab: Explore the data using Python and BigQuery



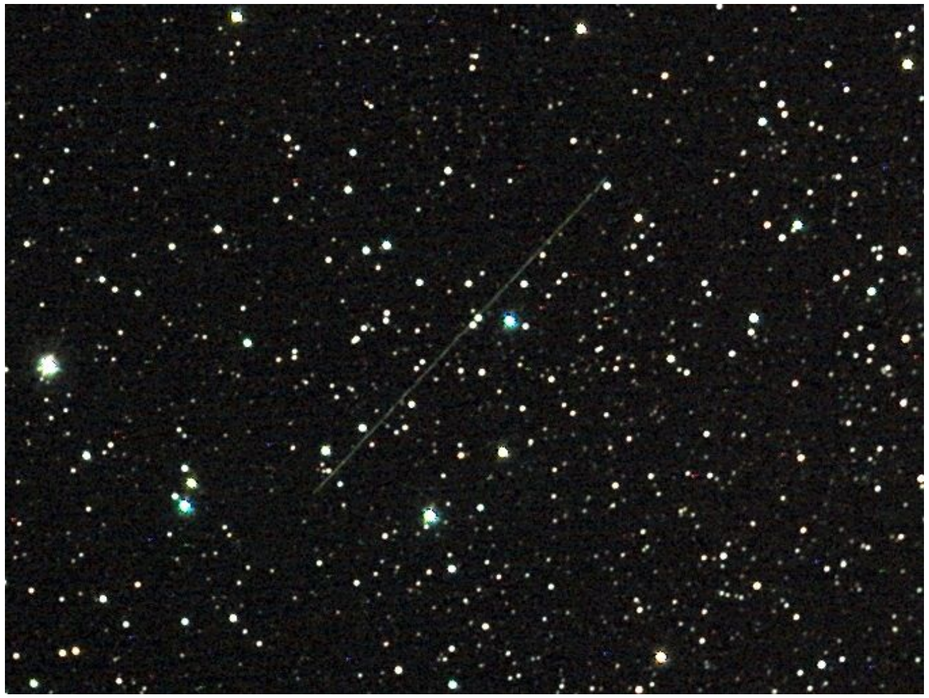
Lab Intro

Explore the Data using Python and BigQuery

Google Cloud

Launching into ML:
ML in Practice

ML in Practice



Agenda

Supervised Learning

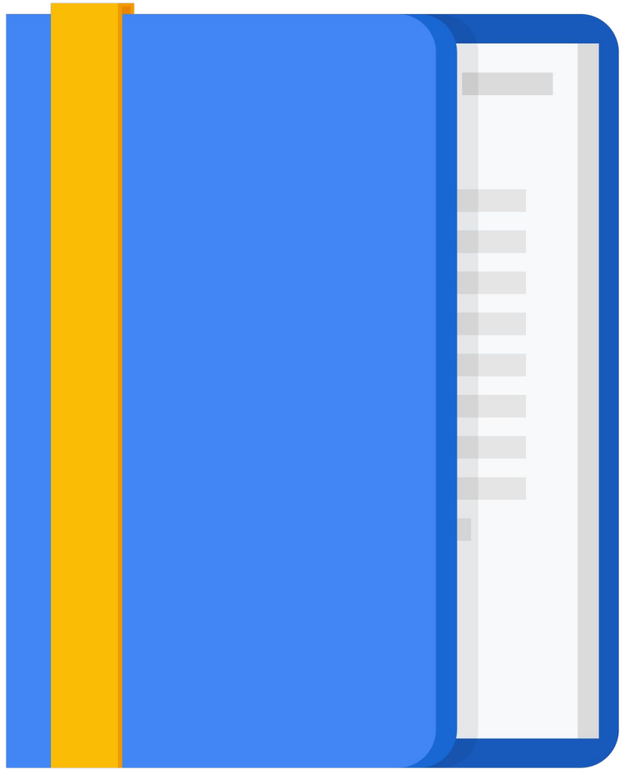
Lab: Introduction to Linear Regression

BigQuery Machine Learning

Lab: Creating BigQuery ML models for Taxifare Prediction

Logistic Regression

Short History of Machine Learning

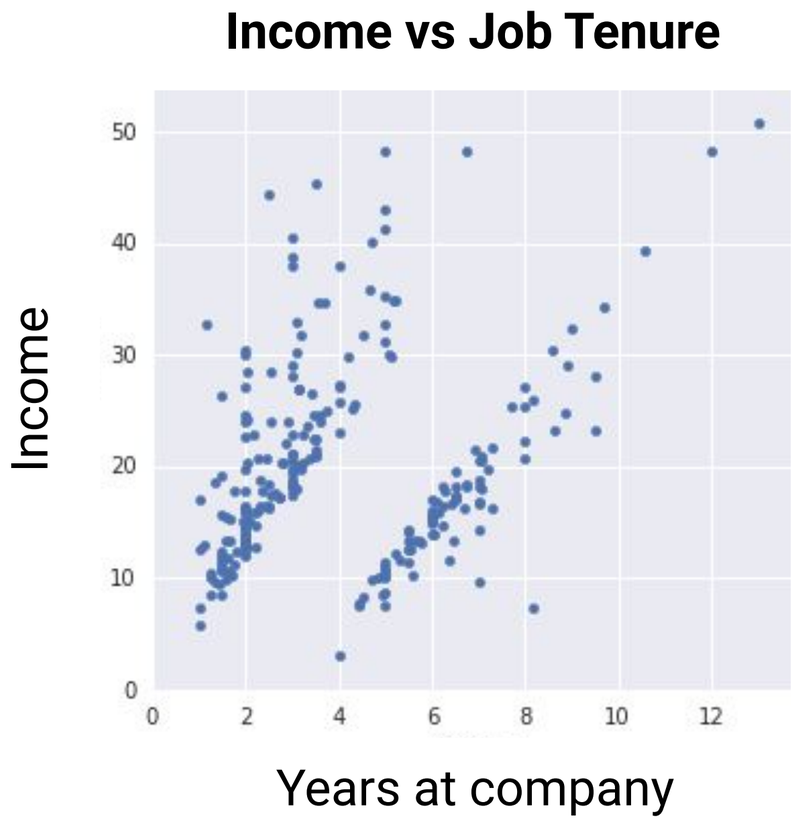


Unsupervised and supervised learning are the two types of ML algorithms

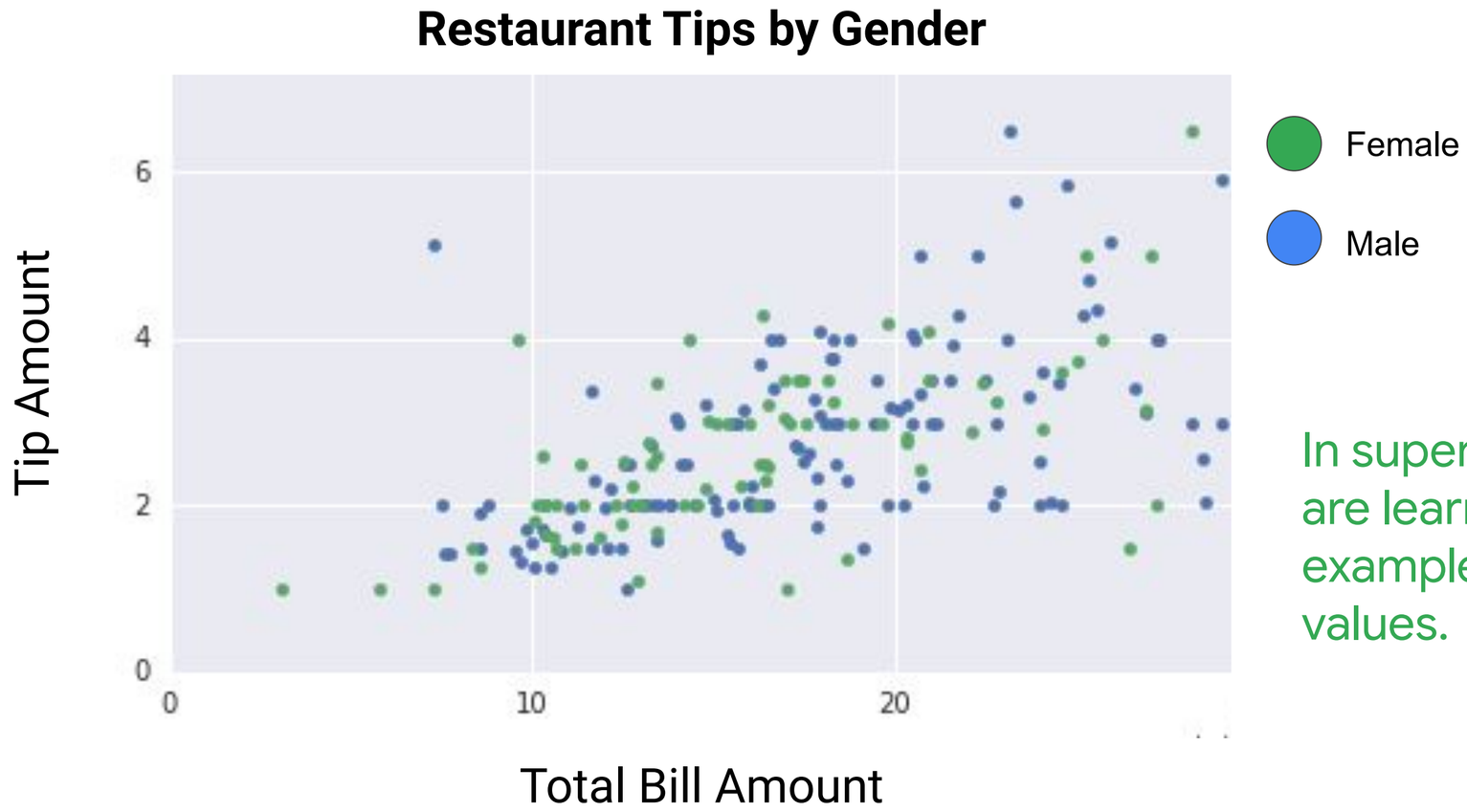
Example Model: Clustering

Is this employee on the “fast-track” or not?

In unsupervised learning, data is not labeled.



Supervised learning implies the data is already labeled



In supervised learning we are learning from past examples to predict future values.

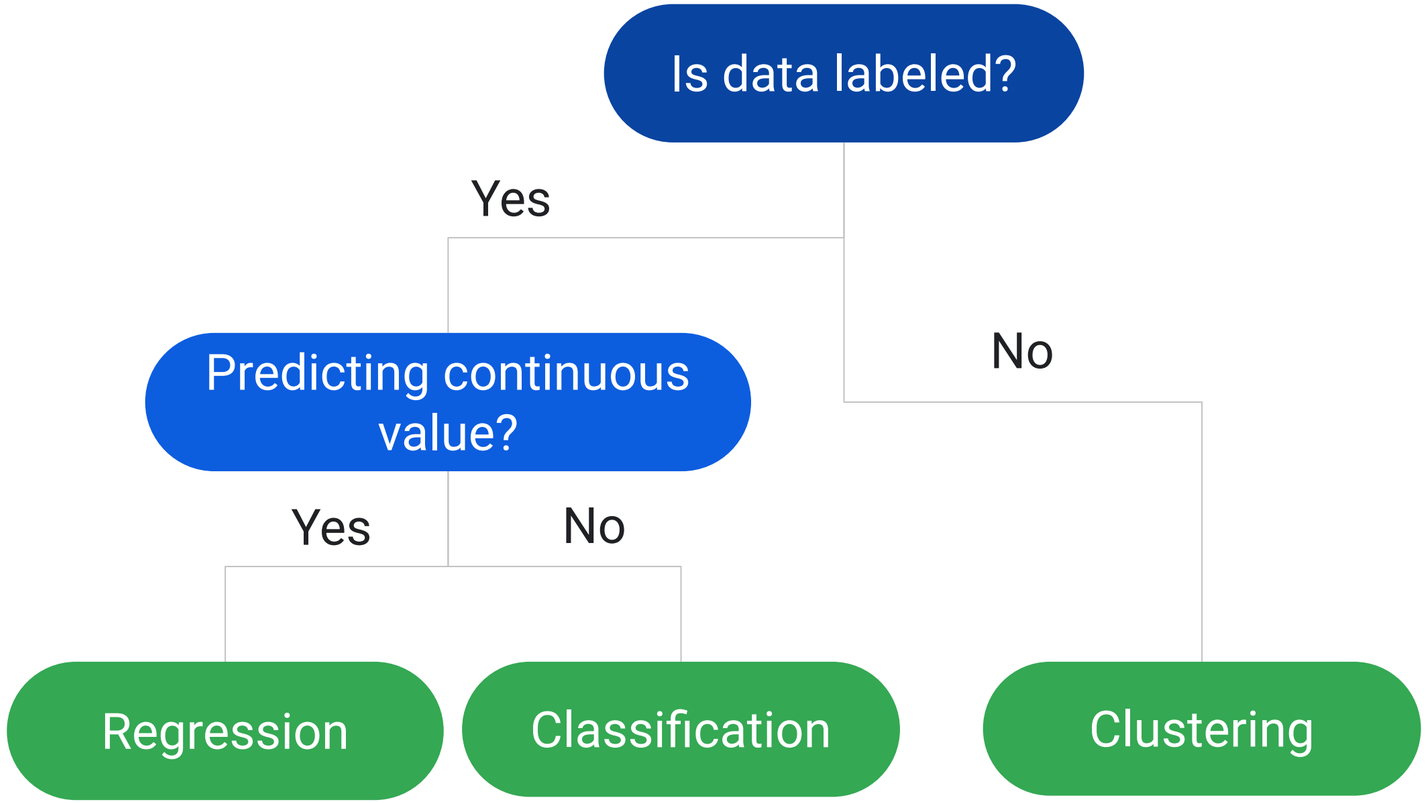
Regression and classification are supervised ML model types

	total_bill	tip	sex	smoker	day	time
2	16.99	1.01	Female	No	Sun	Dinner
3	10.34	1.66	Male	No	Sun	Dinner
4	21.01	3.5	Male	No	Sun	Dinner
5	23.68	3.31	Male	No	Sun	Dinner
6	24.59	3.61	Female	No	Sun	Dinner
7	25.29	4.71	Male	No	Sun	Dinner
8	8.77	2	Male	No	Sun	Dinner
9	26.88	3.12	Male	No	Sun	Dinner

Option 1
Regression Model
Predict the tip amount

Option 2
Classification Model
Predict the sex of the customer

The type of ML problem depends on whether or not you have labeled data and what you are interested in predicting



Quiz: Supervised learning

Imagine you are in banking and you are creating an ML model for detecting if transactions are fraudulent or not. Is this classification or regression and why?

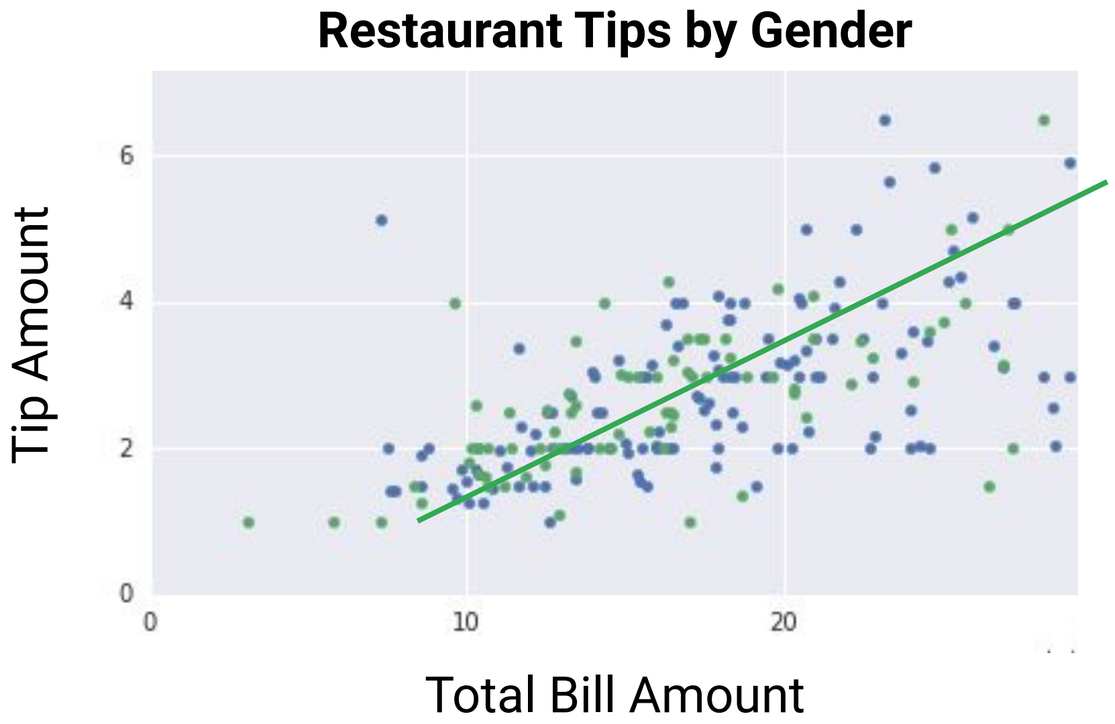
A. Regression, categorical label
B. Regression, continuous label
C. Classification, categorical label
D. Classification, continuous label

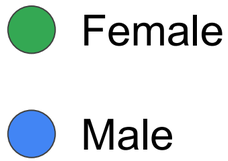
Quiz: Supervised learning

Imagine you are in banking and you are creating an ML model for detecting if transactions are fraudulent or not. Is this classification or regression and why?

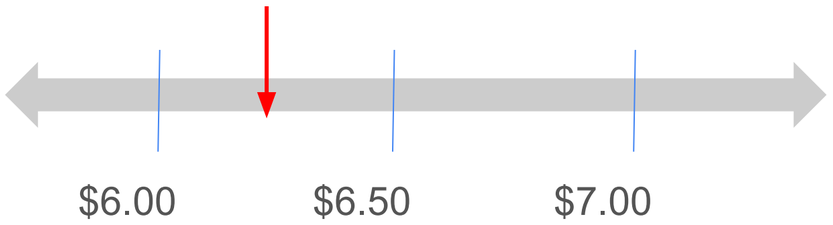
A. Regression, categorical label
B. Regression, continuous label
C. Classification, categorical label
D. Classification, continuous label

Use regression for predicting continuous label values

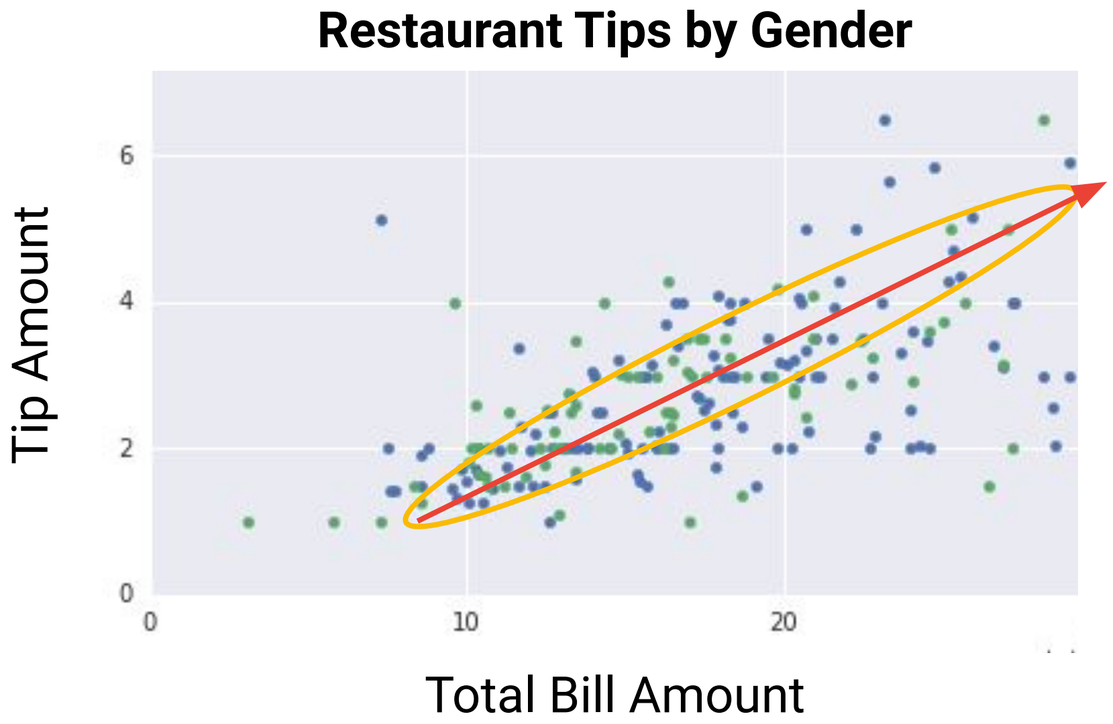




Regression Model (linear) predicts tip amount of $6.25.



Use classification for predicting categorical label values



Female

Male

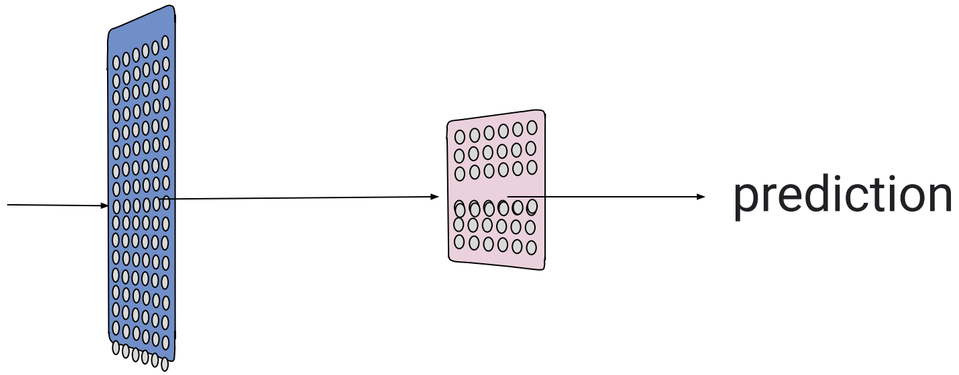
Classification Model (binary) predicts category of Female.

A data warehouse can be a source of structured data training examples for your ML model

```
SELECT
  gestation_weeks,
  mother_age,
  cigarette_use,
  alcohol_use,
  weight_gain_pounds
FROM
  `bigquery-public-data.samples.natality`
WHERE cigarette_use is not null AND alcoho
```

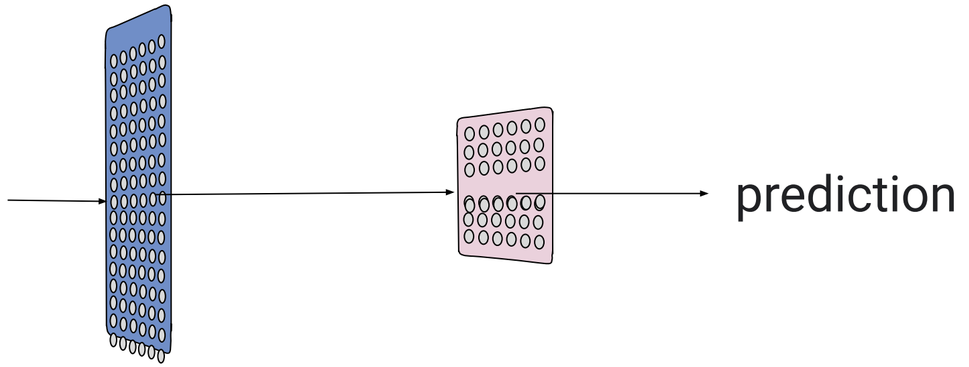
weight	year	mother_age	gestation_weeks	cigarette_use	alcohol_use
7.86	2003	25	39	false	false
7.5	2003	21	39	false	false
8.06	2004	29	40	false	false
7.56	2004	38	37	false	false
7.06	2003	22	38	false	false

Data on births is sourced from our BigQuery Data Warehouse using SQL.



Since baby weight is a continuous value, use regression to predict

weight	year	mother_age	gestation_weeks	cigarette_use	alcohol_use
7.86	2003	25	39	false	false
7.5	2003	21	39	false	false
8.06	2004	29	40	false	false
7.56	2004	38	37	false	false
7.06	2003	22	38	false	false



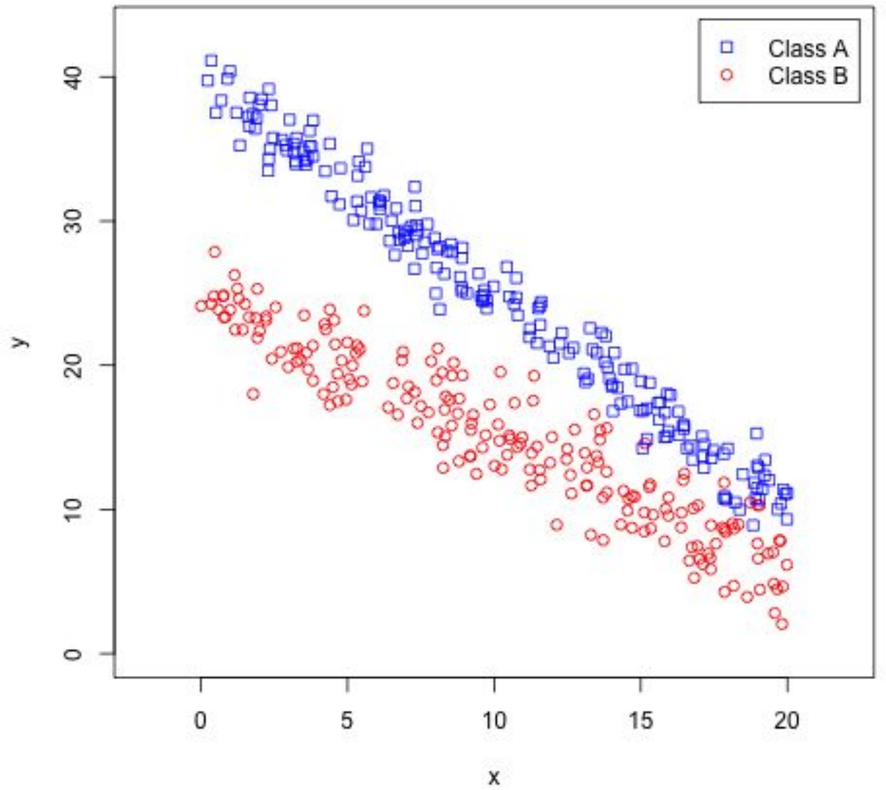
Weight is stored as a floating point number, representing a continuous (real) value.

Regression DNN Model

Quiz: Regression/Classification

Is this dataset a good candidate for linear regression and/or linear classification?

A. Linear classification
B. Both
C. None of the above



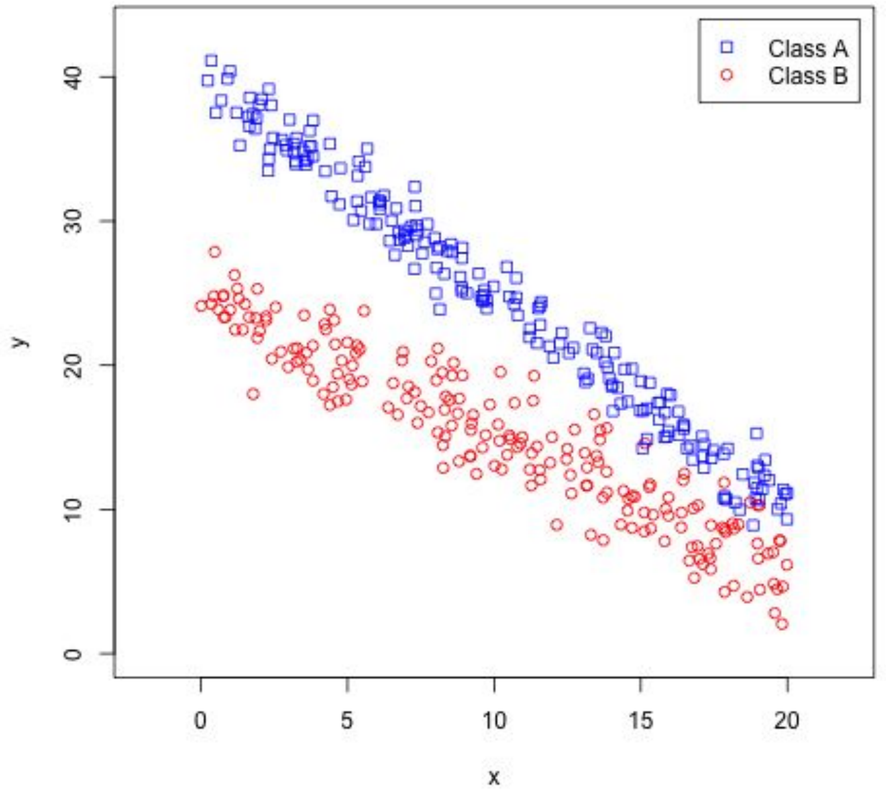
Quiz: Regression/Classification

Is this dataset a good candidate for linear regression and/or linear classification?

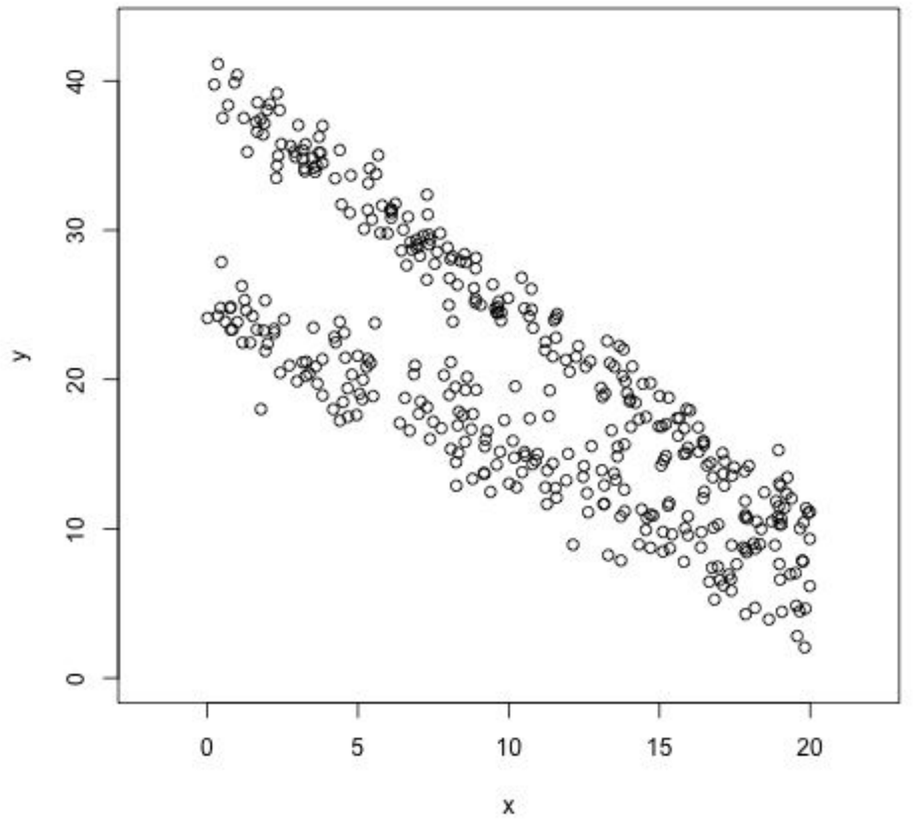
A. Linear classification

B. Both

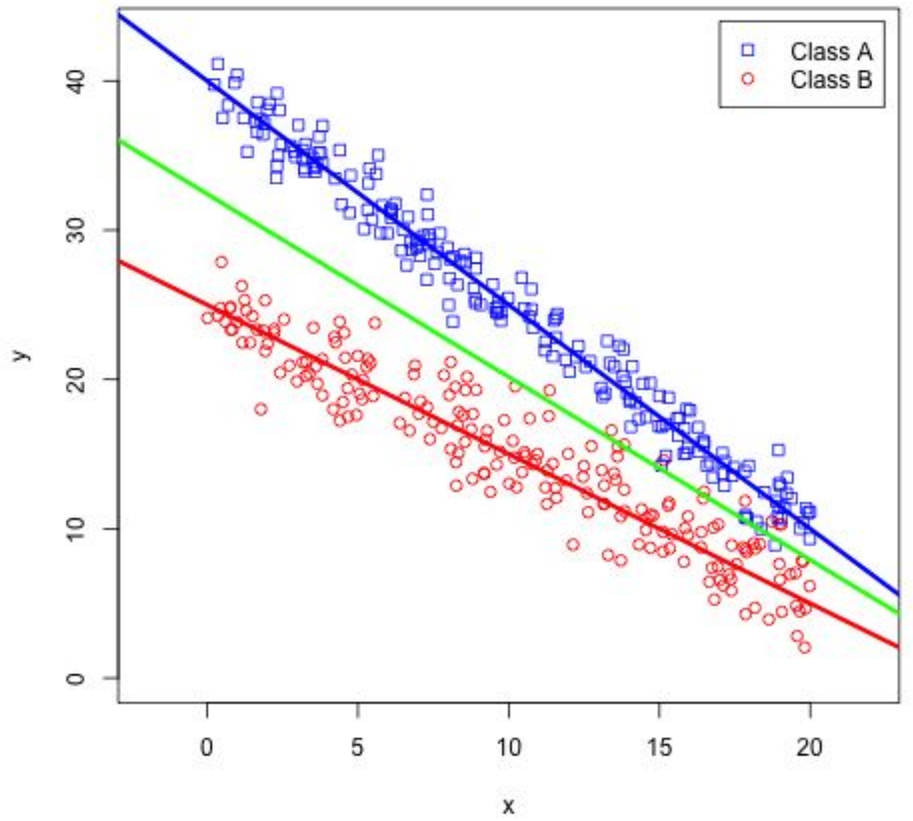
C. None of the above



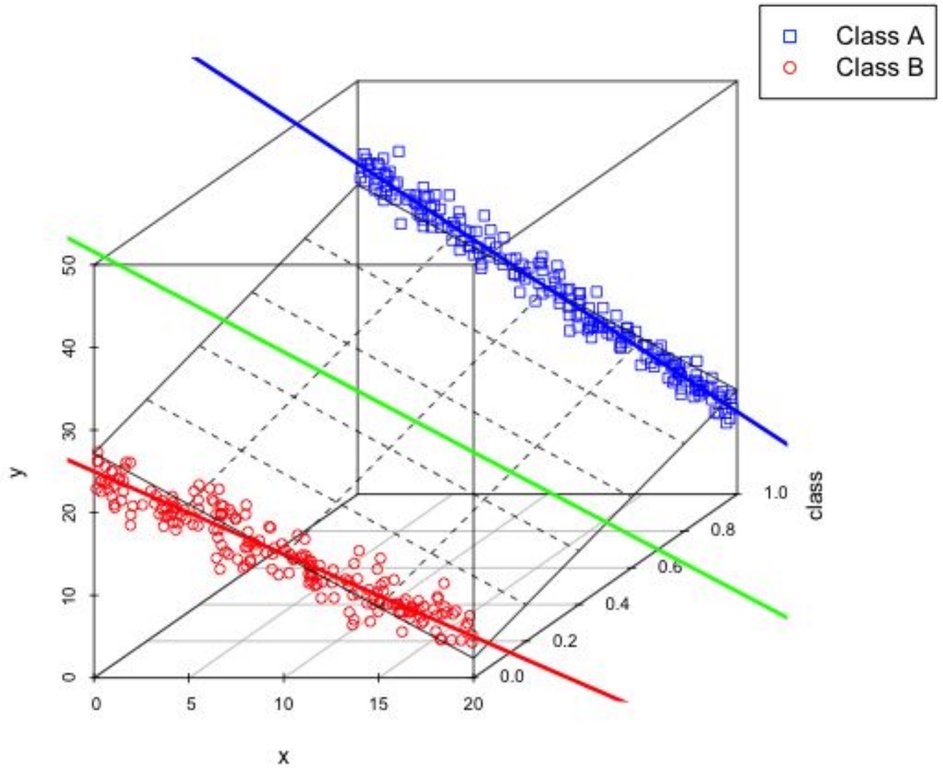
Is this dataset a good candidate for linear regression and/or linear classification?



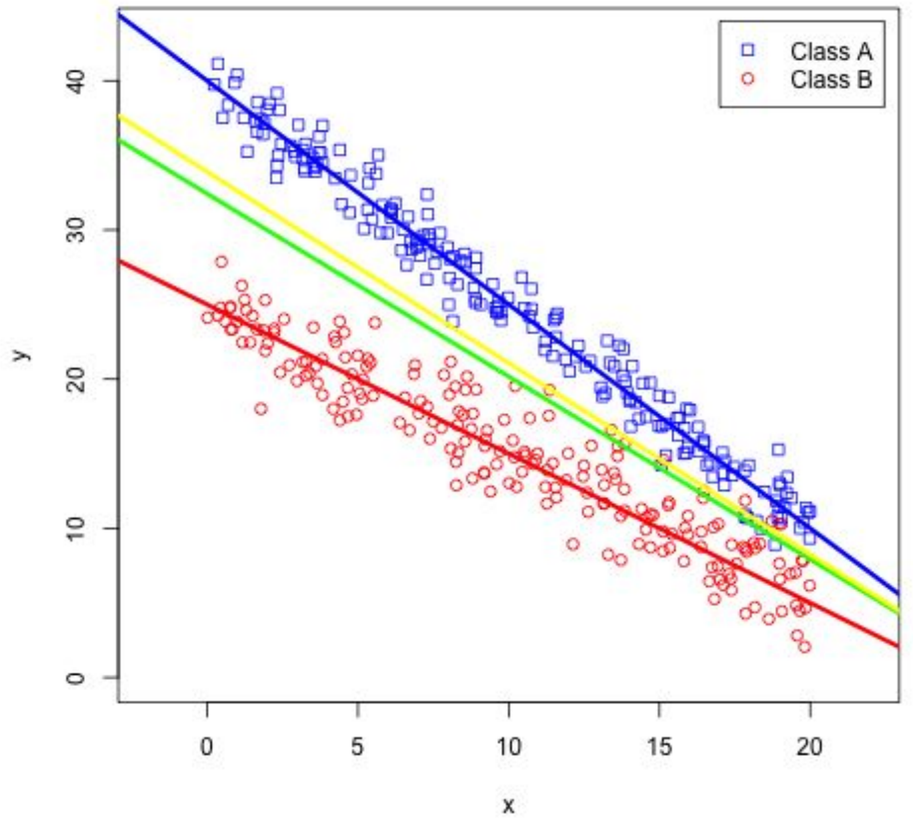
Is this dataset a good candidate for linear regression and/or linear classification?



Is this dataset a good candidate for linear regression and/or linear classification?



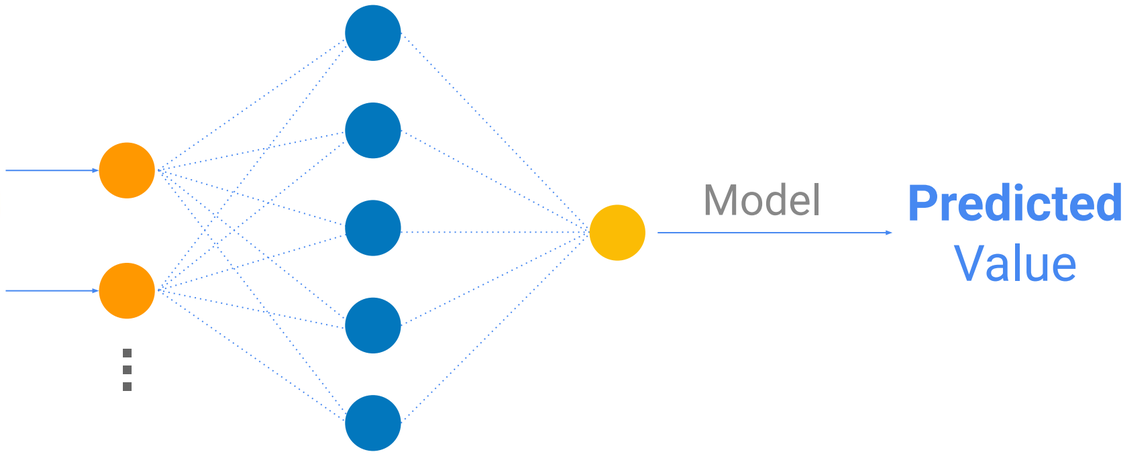
Is this dataset a good candidate for linear regression and/or linear classification?



The point of ML is to make predictions

	mintemp	maxtemp	rain	day_1	day_2	day_3	day_4	day_5	day_6	day_7
104	28.9	37.9	0.01	0	0	0	1	0	0	0
9	32.0	43.0	0.00	0	1	0	0	0	0	0
114	35.1	48.0	0.00	1	0	0	0	0	0	0
11	30.0	37.9	0.00	0	0	0	0	0	0	0
316	19.0	39.9	0.05	0	0	1	0	0	0	0

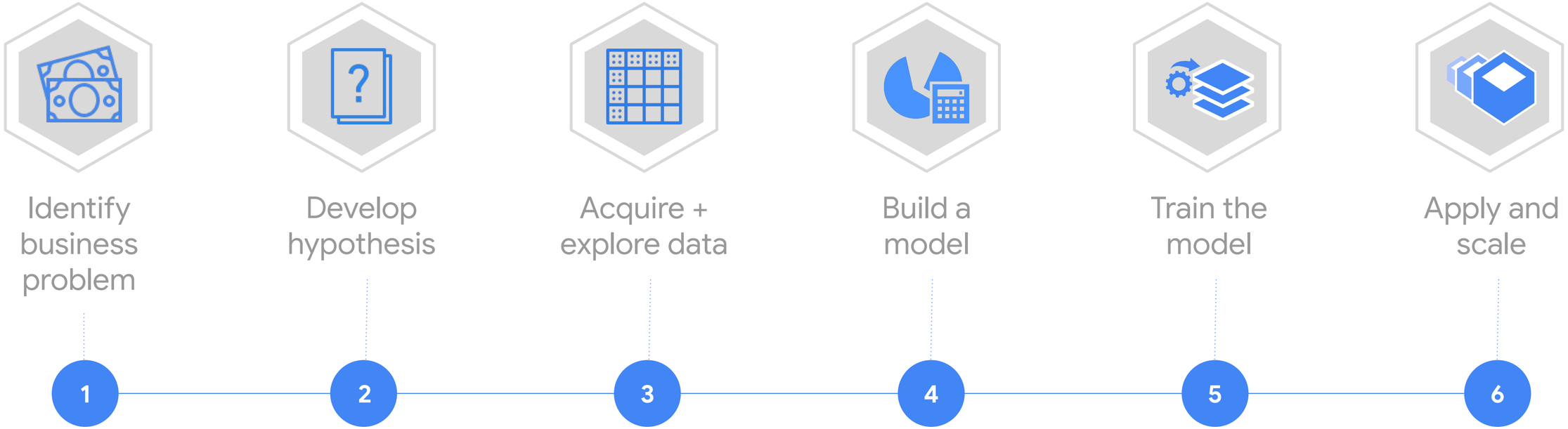
Input Feature



Neural network image by Dake, Mysid [CC BY 1.0], via Wikimedia Commons

Google Cloud

To build a machine learning model



Google Cloud

Agenda

Supervised Learning

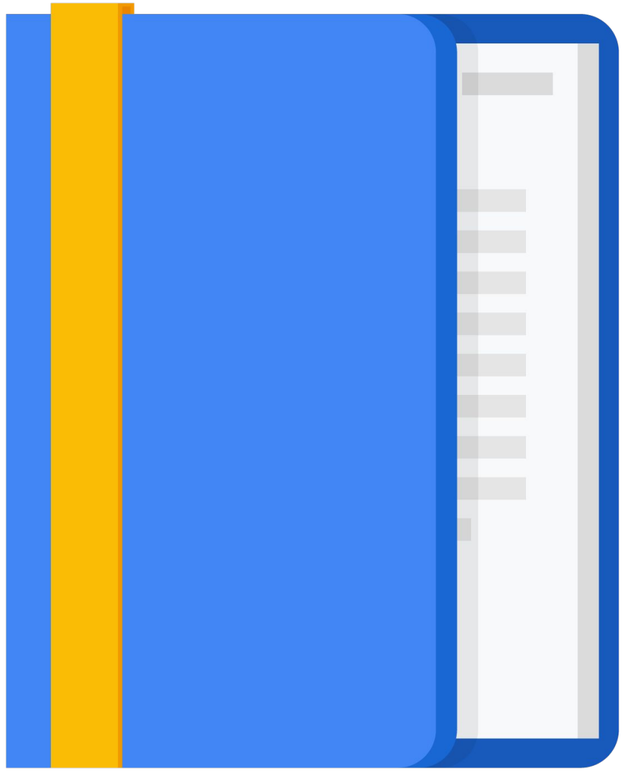
Lab: Introduction to Linear Regression

BigQuery Machine Learning

Lab: Creating BigQuery ML models for Taxifare Prediction

Logistic Regression

Short History of Machine Learning



Lab Intro

Introduction to Linear Regression

Google Cloud

Agenda

Supervised Learning

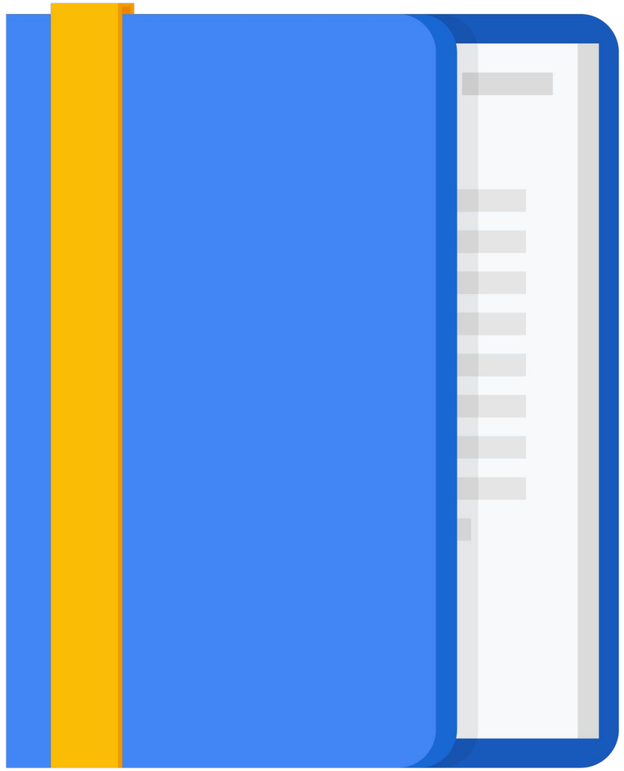
Lab: Introduction to Linear Regression

BigQuery Machine Learning

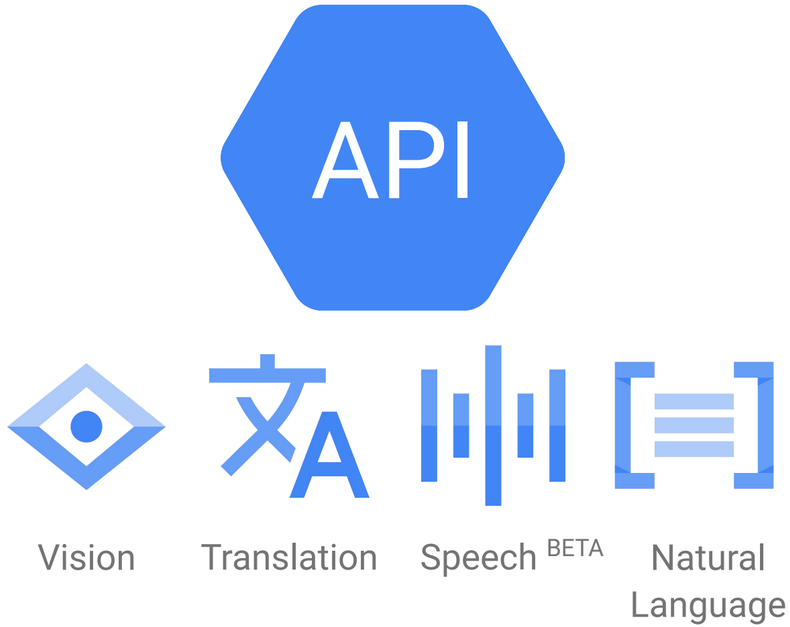
Lab: Creating BigQuery ML models for Taxifare Prediction

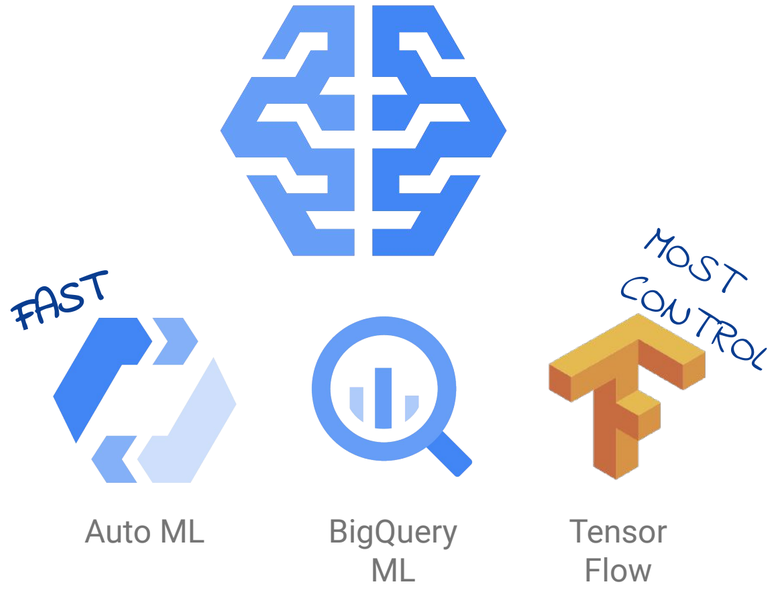
Logistic Regression

Short History of Machine Learning



BigQuery ML is a way to build custom models





Pre-trained models

very common to enrich your data with pre-trained models, to take advantage of unstructured data

Build your own model

Developers
Data Analysts
ML engineers

Google Cloud

Working with BigQuery ML

1. Write SQL query to extract training data from BigQuery
2. Create a model, specifying model type
3. Evaluate model and verify that it meets requirements
4. Predict using model on data extracted from BigQuery

Where was this article published?

1. Techcrunch
2. GitHub
3. NY Times

Unlikely Partnership in House Gives Lawmakers Hope for Border Deal

Representatives Nita M. Lowey and Kay Granger are the first women to lead the House Appropriations Committee. Their bond gives lawmakers optimism for the work to come.

By EMILY COCHRANE

Fitbit's newest fitness tracker is just for employees and health insurance members

1 hour ago **Jon Russell**

Fitbit has a new fitness tracker, but it's one that you can't buy in stores. The company quietly uncorked the Inspire on Friday, releasing its first product that is available only to co...

Downloading the Android Studio Project Folder

FTC Engineering edited this page on Sep 19, 2017 · 1 revision

Downloading the Android Studio Project Folder

Google Cloud

SQL query to extract data

1

```
SELECT
  url, title
FROM
  `bigquery-public-data.hacker_news.stories`
WHERE
  LENGTH(title) > 10
  AND LENGTH(url) > 0
LIMIT 10
```

url	title
http://www.bbc.co.uk/news/business-27732743	Vodafone reveals direct government wiretaps
https://www.kickstarter.com/projects/appdocu/a...	Doc – App: The Human Story
http://www.starwebworld.com/android-jelly-bean...	Android Jelly Bean: Streaming Audio Through th...
http://www.myplanetdigital.com/digital_strateg...	Why Canadian Tech Entrepreneurs Need to Man/Wo...
http://startupislandconference.com/index.html	StartupConference June 13. - 16. 2013, HVAR Cr...
http://kopimism.org/	Kopimism Hactivism Meetup Tomorrow (Sunday) in...
http://unearthedgadget.com/xbox-live-gold-2/14...	Xbox Live Gold Membership Is It Really Worth -...
https://evertale.com	Evertale changes the way people remember
http://www.racketboy.com/retro/commodore-amiga...	Commodore Amiga: A Beginner's Guide
http://www.extremetech.com/extreme/156393-cold...	Cold fusion reactor "independently verified"

Use regex to get source + train on words of title

1

txtclass_words — LINK SHARING

```
WITH extracted AS (
SELECT source, REGEXP_REPLACE(LOWER(REGEXP_REPLACE(title, '[^a-zA-Z0-9 $.-]', ' ')), "  "
FROM
  (SELECT
    ARRAY_REVERSE(SPLIT(REGEXP_EXTRACT(url, '.*://(.[^/]+)/'), '.'))[OFFSET(1)] AS source
    title
  FROM
    `bigquery-public-data.hacker_news.stories`
  WHERE
    REGEXP_CONTAINS(REGEXP_EXTRACT(url, '.*://(.[^/]+)/'), '.com$')
    AND LENGTH(title) > 10
  )
)
, ds AS (
SELECT ARRAY_CONCAT(SPLIT(title, " "), ['NULL', 'NULL', 'NULL', 'NULL', 'NULL']) AS words
extracted
WHERE (source = 'github' OR source = 'nytimes' OR source = 'techcrunch')
)
SELECT
source,
words[OFFSET(0)] AS word1,
words[OFFSET(1)] AS word2,
words[OFFSET(2)] AS word3,
words[OFFSET(3)] AS word4,
words[OFFSET(4)] AS word5
FROM ds
```

Run | Save query | Save view | More — This query will process 204.

Query results — SAVE RESULTS — EXPLORE IN DATA STUDIO

37293	nytimes	the	socratic	shrink	NULL	NULL
37294	nytimes	still	stuck	in	a	climate
37295	nytimes	as	unlimited	data	plans	are
37296	nytimes	disney	s	neuroscience	advertising	lab

Create classification model

2

```
CREATE OR REPLACE MODEL advdata.txtclass
OPTIONS(model_type='logistic_reg', input_label_cols=['source'])
AS

WITH extracted AS (
...
)
, ds AS (
SELECT ARRAY_CONCAT(SPLIT(title, " "), ['NULL', 'NULL', 'NULL',
'NULL', 'NULL']) AS words, source FROM extracted
WHERE (source = 'github' OR source = 'nytimes' OR source =
'techcrunch')
)
SELECT
source,
words[OFFSET(0)] AS word1,
words[OFFSET(1)] AS word2,
words[OFFSET(2)] AS word3,
words[OFFSET(3)] AS word4,
words[OFFSET(4)] AS word5
FROM ds
```

Query to extract training data

Evaluate model

```
SELECT * FROM ML.EVALUATE(MODEL advdata.txtclass)
```

3

precision	recall	accuracy	f1_score	log_loss	roc_auc
0.783	0.783	0.79	0.783	0.858	0.918

(BigQuery ML splits the training data and reports evaluation statistics on the held-out set)

Actual labels	Predicted labels: github	nytimes	techcrunch	% samples
github	88.8%	5.29%	5.9%	37.83%
nytimes	6.34%	70.92%	22.74%	31.26%
techcrunch	5.54%	19.35%	75.11%	30.9%

Predict using trained model

4

```
SELECT * FROM ML.PREDICT(MODEL advdata.txtclass,(
     SELECT 'government' AS word1, 'shutdown' AS word2, 'leaves'
AS word3, 'workers' AS word4, 'reeling' AS word5
     UNION ALL SELECT 'unlikely', 'partnership', 'in', 'house',
'gives'
     UNION ALL SELECT 'fitbit', 's', 'fitness', 'tracker', 'is'
     UNION ALL SELECT 'downloading', 'the', 'android', 'studio',
'project'
))
```

Row	predicted_source	word1	word2	word3	word4	word5
1	nytimes	government	shutdown	leaves	workers	reeling
2	nytimes	unlikely	partnership	in	house	gives
3	techcrunch	fitbit	s	fitness	tracker	is
4	techcrunch	downloading	the	android	studio	project

"Batch prediction"

Linear Classifier (Logistic regression)

```
create or replace model models.will_buy_banana_example
options(model_type='logistic_reg', input_label_cols=['banana']) AS

with purchases_data AS (
select
  CAST(user_id AS string) customer_id,
  CAST(zip_code AS string) home_zipcode,
  ['dawn', 'morning', 'afternoon', 'night'][OFFSET(CAST
(TRUNC(order_hour_of_day/6) AS INT64))] AS time_of_day,
  (SELECT 24852 IN (SELECT product_id FROM UNNEST(order_lines))) AS
banana
FROM operations.orders_with_lines
JOIN operations.customers_loyalty
ON user_id = id
)
select * from purchases_data
```

DNN Classifier

```
create or replace model models.will_buy_banana_example
options(model_type='dnn_classifier', hidden_units=[64, 8],
input_label_cols=['banana']) AS

with purchases_data AS (
select
  CAST(user_id AS string) customer_id,
  CAST(zip_code AS string) home_zipcode,
  ['dawn', 'morning', 'afternoon', 'night'][OFFSET(CAST
(TRUNC(order_hour_of_day/6) AS INT64))] AS time_of_day,
  (SELECT 24852 IN (SELECT product_id FROM UNNEST(order_lines))) AS
banana
FROM operations.orders_with_lines
JOIN operations.customers_loyalty
ON user_id = id
)
select * from purchases_data
```

Linear regression

```
CREATE OR REPLACE MODEL ch09edu.bicycle_model
OPTIONS(input_label_cols=['duration'],
        model_type='linear_reg')
AS

SELECT
  duration
  , start_station_name
  , CAST(EXTRACT(dayofweek from start_date) AS STRING)
         as dayofweek
  , CAST(EXTRACT(hour from start_date) AS STRING)
         as hourofday
FROM
  `bigquery-public-data.london_bicycles.cycle_hire`
```

DNN regression

```
CREATE OR REPLACE MODEL ch09edu.bicycle_model
OPTIONS(input_label_cols=['duration'],
        model_type='dnn_regressor', hidden_units=[32, 4])
AS

SELECT
  duration
  , start_station_name
  , CAST(EXTRACT(dayofweek from start_date) AS STRING)
         as dayofweek
  , CAST(EXTRACT(hour from start_date) AS STRING)
         as hourofday
FROM
  `bigquery-public-data.london_bicycles.cycle_hire`
```

Agenda

Supervised Learning

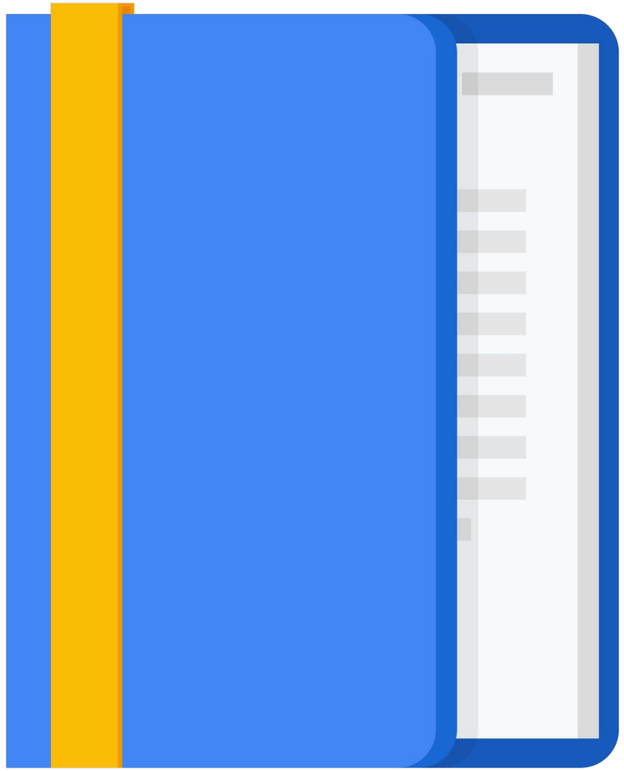
Lab: Introduction to Linear Regression

BigQuery Machine Learning

Lab: Creating BigQuery ML models for Taxifare Prediction

Logistic Regression

Short History of Machine Learning



Lab Intro

Creating BigQuery ML models for Taxifare Prediction

Google Cloud

Agenda

Supervised Learning

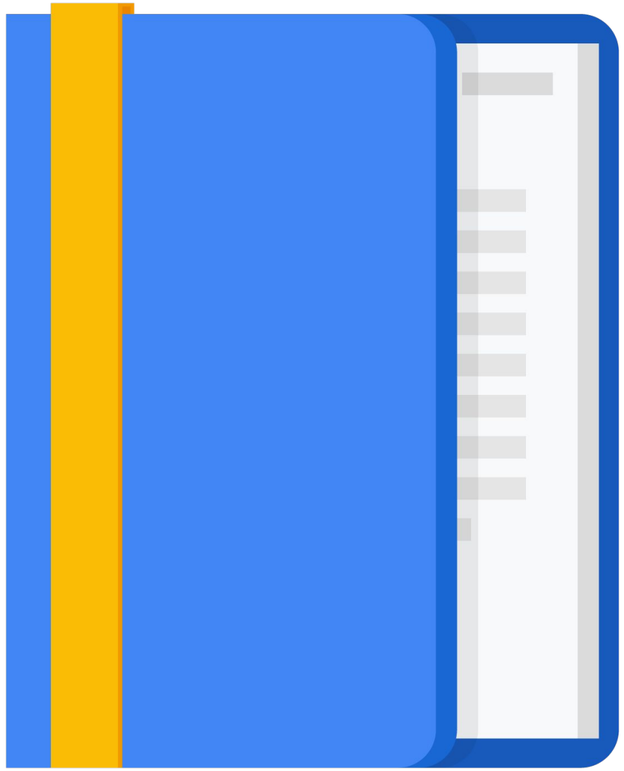
Lab: Introduction to Linear Regression

BigQuery Machine Learning

Lab: Creating BigQuery ML models for Taxifare Prediction

Logistic Regression

Short History of Machine Learning



Suppose you use linear regression to predict coin flips

You might use features like angle of bend, coin mass, etc. What could go wrong?

Suppose you use linear regression to predict coin flips

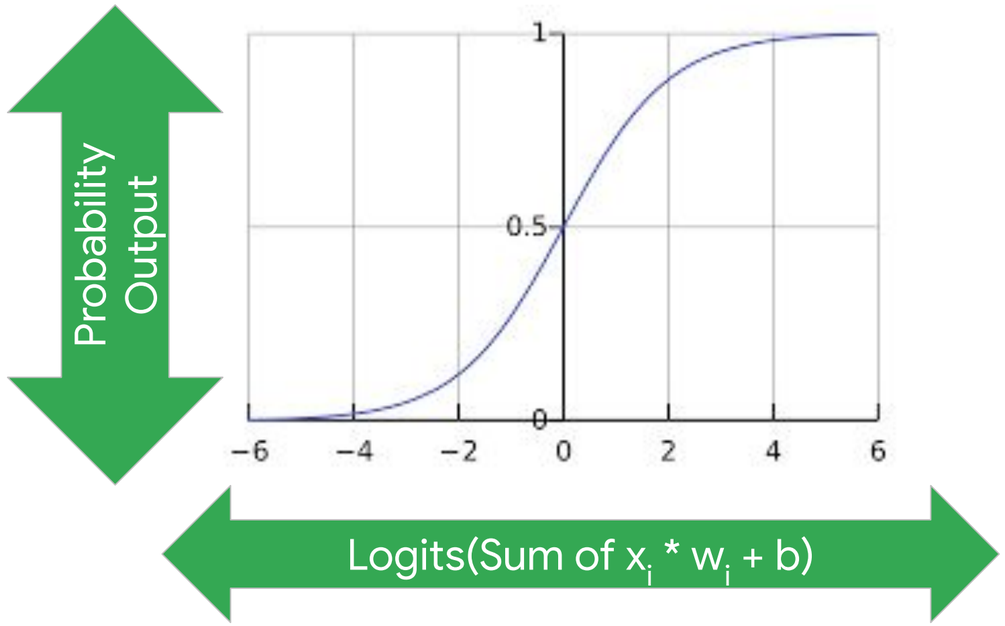
What could go wrong?

Logistic Regression: transform linear regression by a sigmoid activation function

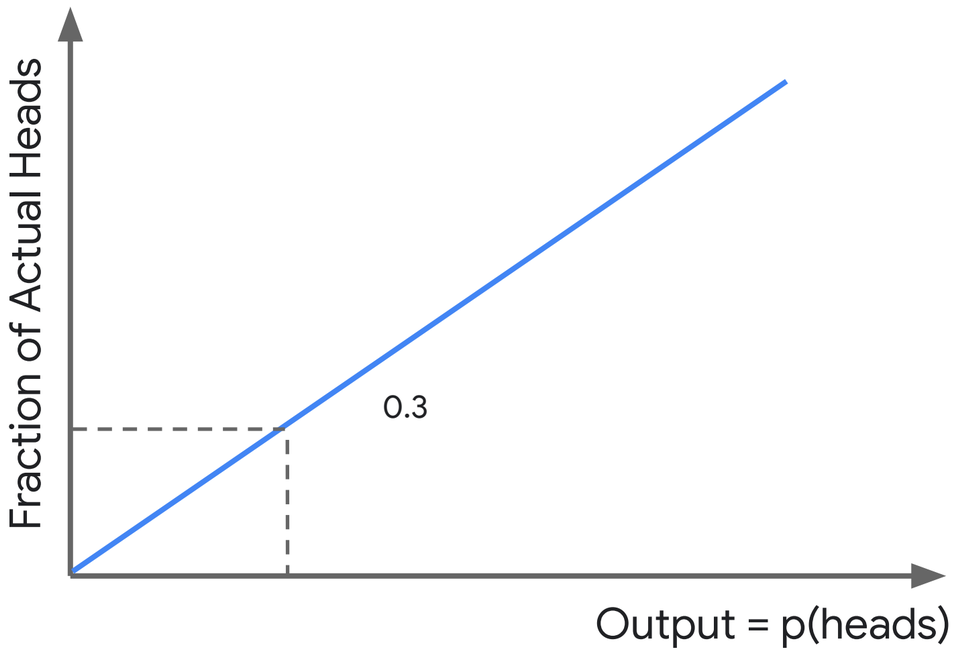
linear model

$$\hat{y} = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x}+b)}}$$

squish through a sigmoid



The output of Logistic Regression is a calibrated probability estimate



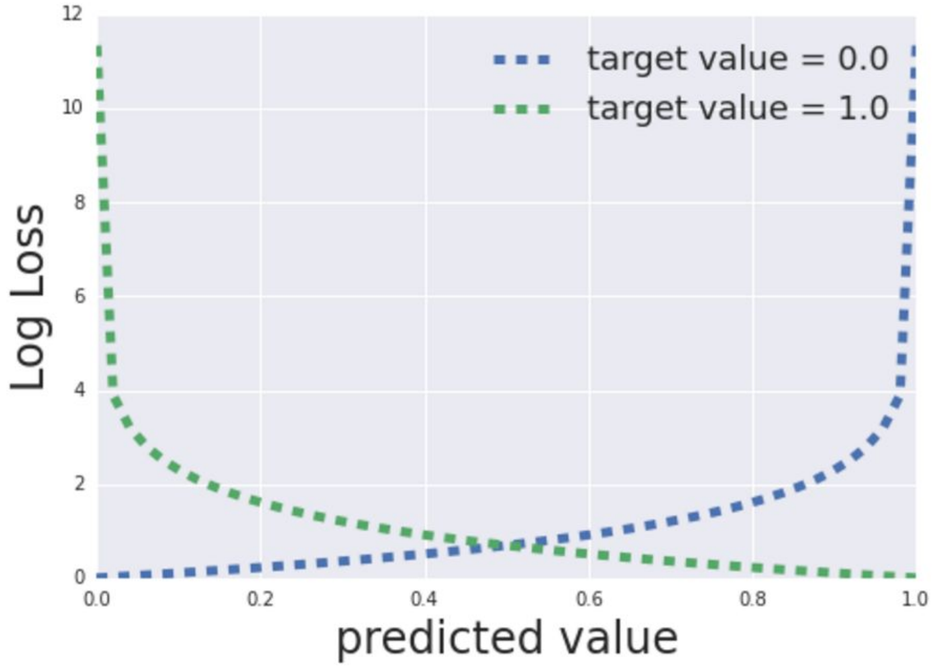
Useful because we can cast binary classification problems into probabilistic problems:

Will customer buy item?

becomes

Predict the probability that customer buys item

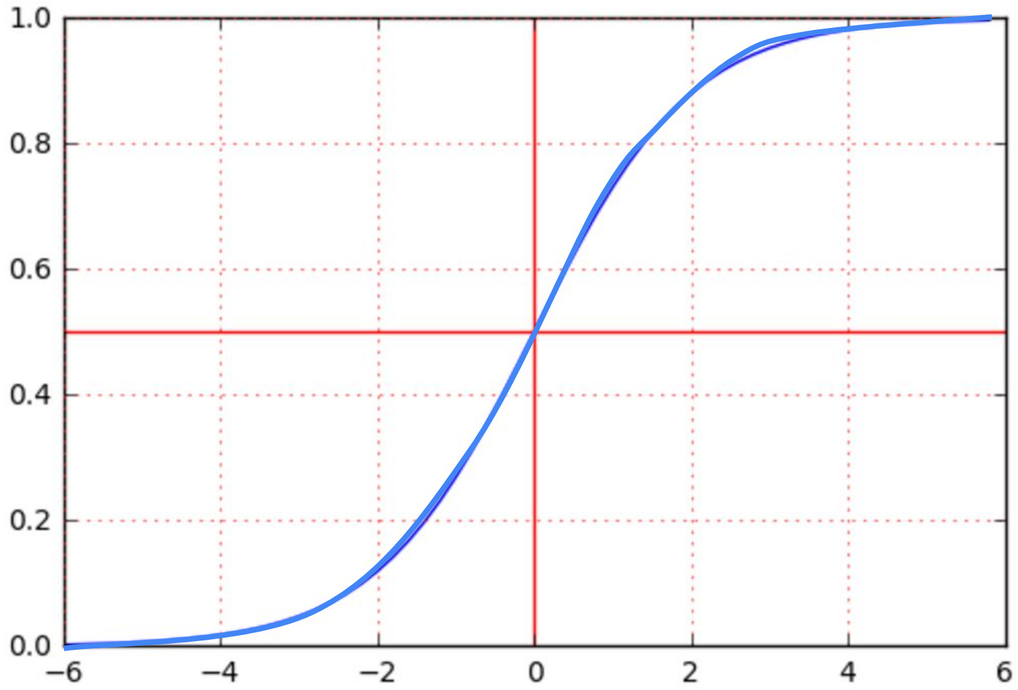
Typically, use cross-entropy (related to Shannon's information theory) as the error metric



Less emphasis on errors where the output is relatively close to the label.

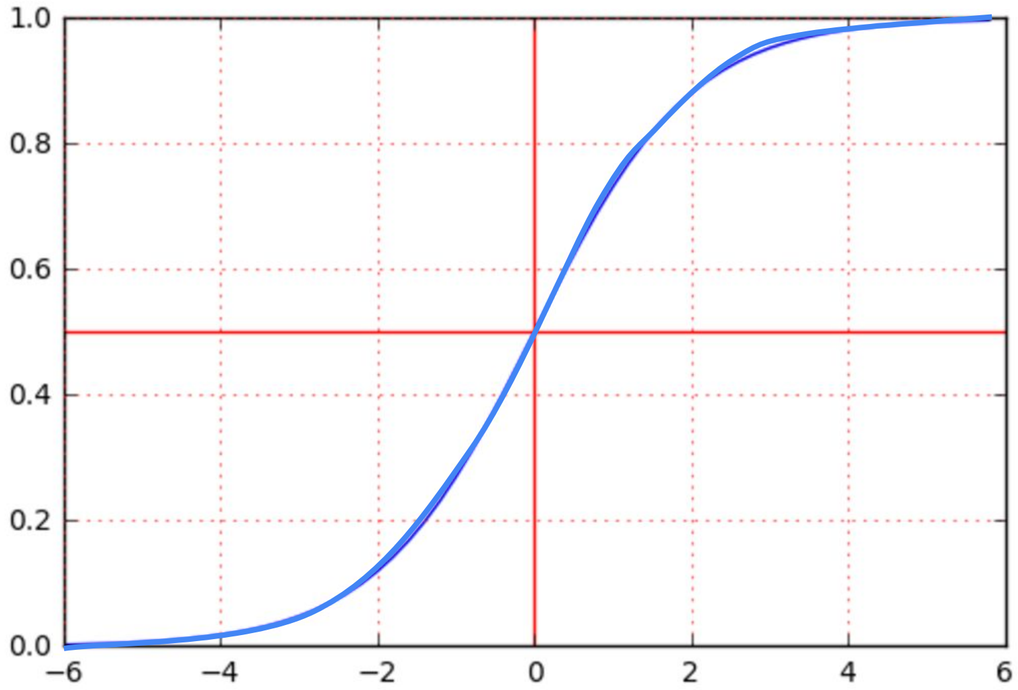
$$LogLoss = \sum_{(\boldsymbol{x},y)\in D} -ylog(\hat{y}) - (1-y)log(1-\hat{y})$$

Regularization is important in logistic regression because driving the loss to zero is difficult and dangerous



Weights will be driven to -inf and +inf the longer we train

Regularization is important in logistic regression because driving the loss to zero is difficult and dangerous



Near the asymptotes, gradient is really small

Weights will be driven to -inf and +inf the longer we train

Logistic Regression Regularization Quiz

Why is it important to add regularization to logistic regression?

A. Helps stops weights being driven to +/- infinity.
B. Helps logits stay away from asymptotes which can halt training
C. Transforms outputs into a calibrated probability estimate
D. Both A & B
E. Both A & C

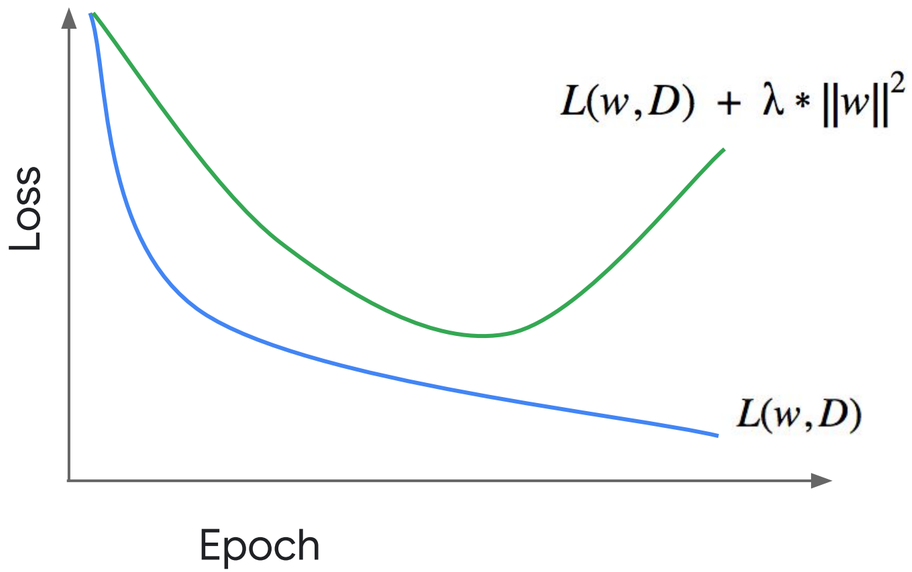
Logistic Regression Regularization Quiz

Why is it important to add regularization to logistic regression?

A. Helps stops weights being driven to +/- infinity.
B. Helps logits stay away from asymptotes which can halt training
C. Transforms outputs into a calibrated probability estimate
D. Both A & B
E. Both A & C

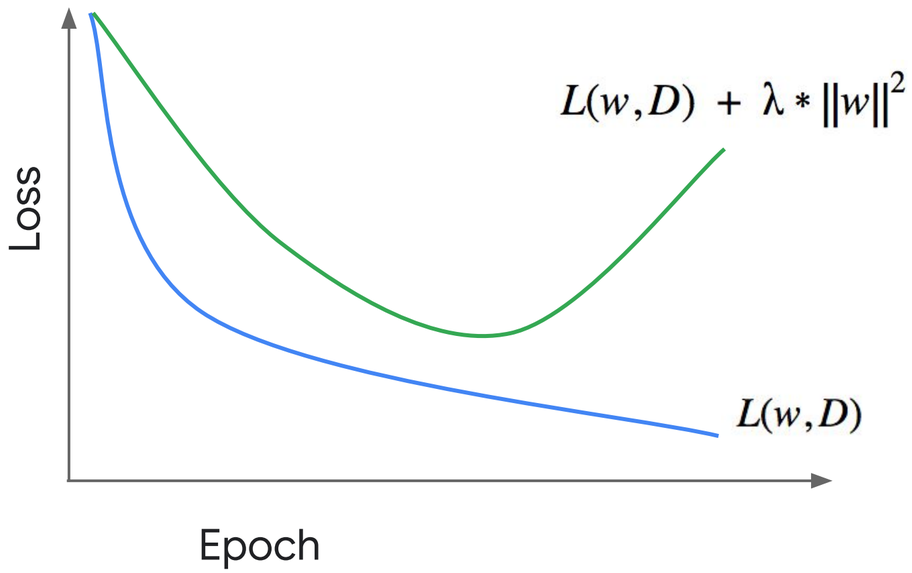
Often we do both regularization and early stopping to counteract overfitting

Regularization

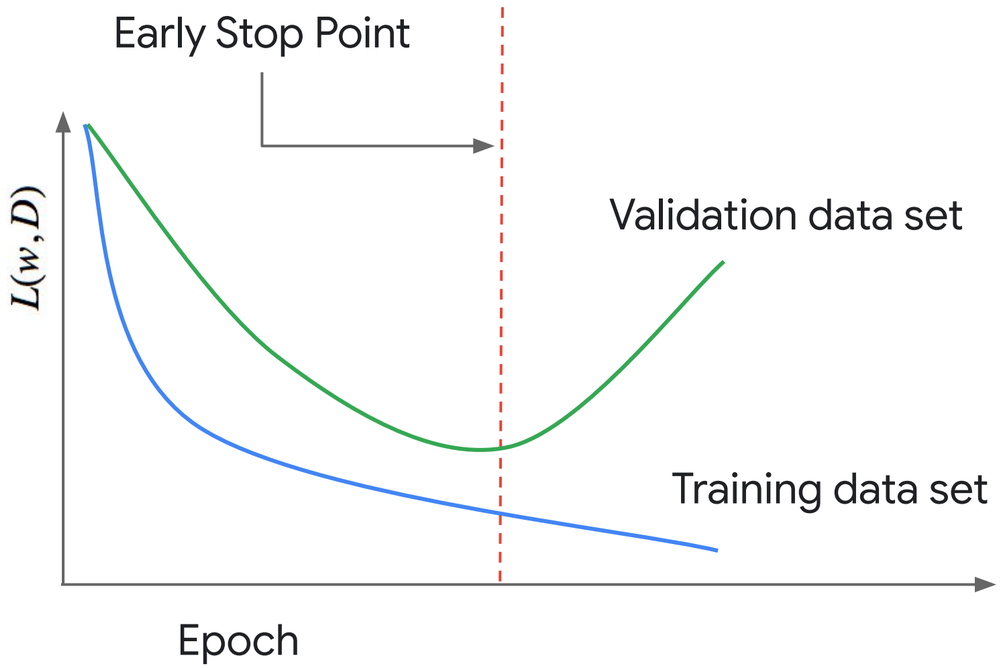


Often we do both regularization and early stopping to counteract overfitting

Regularization



Early stopping



In many real-world problems, the probability is not enough; we need to make a binary decision

Send the mail to spam folder or not?

Approve the loan or not?

Which road should we route the user through?

In many real-world problems, the probability is not enough; we need to make a binary decision

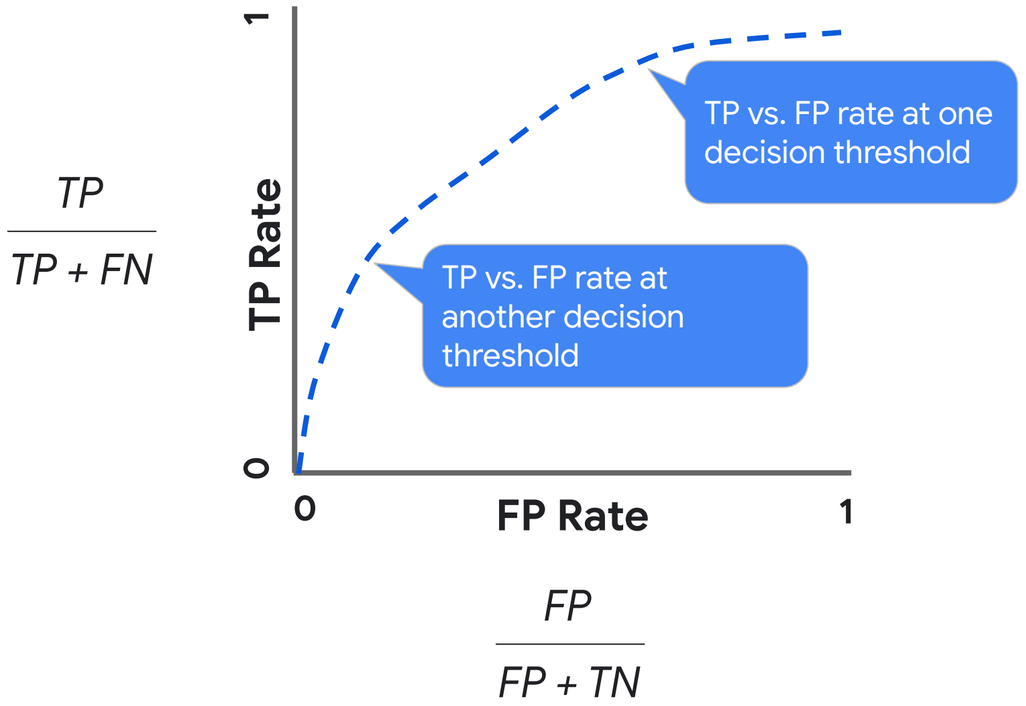
Send the mail to spam folder or not?

Approve the loan or not?

Which road should we route the user through?

Choice of threshold is important and can be tuned

Use the ROC curve to choose the decision threshold based on decision criteria



The Area-Under-Curve (AUC) provides an aggregate measure of performance across all possible classification thresholds

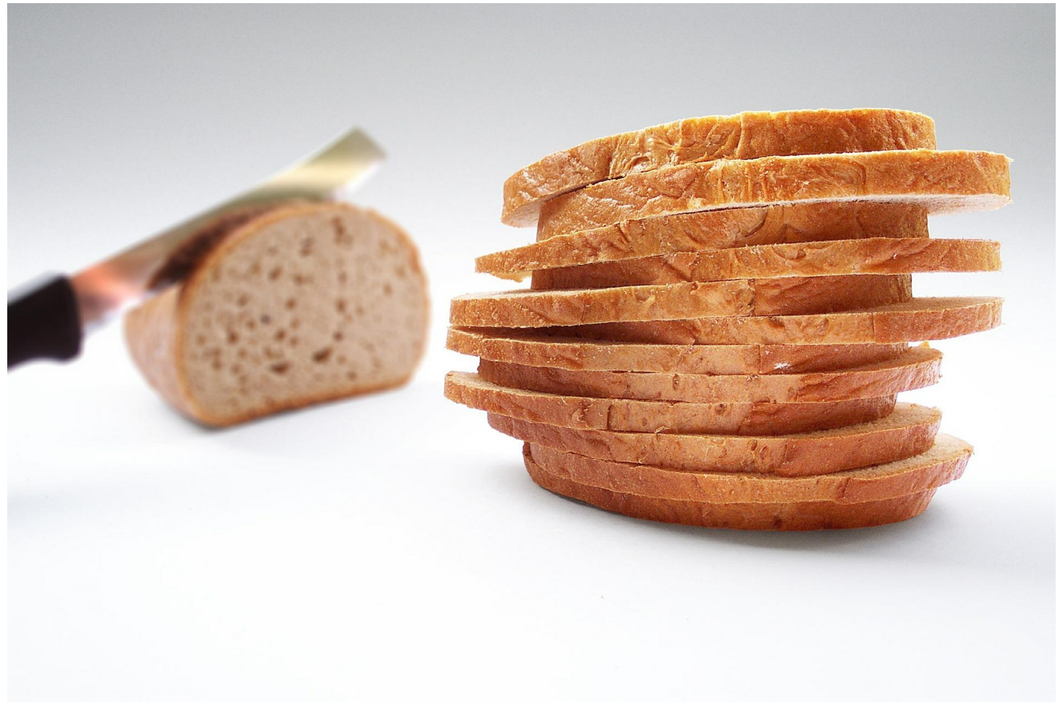
AUC helps you choose between models when you don't know what decision threshold is going to be ultimately used.

"If we pick a random positive and a random negative, what's the probability my model scores them in the correct relative order?"

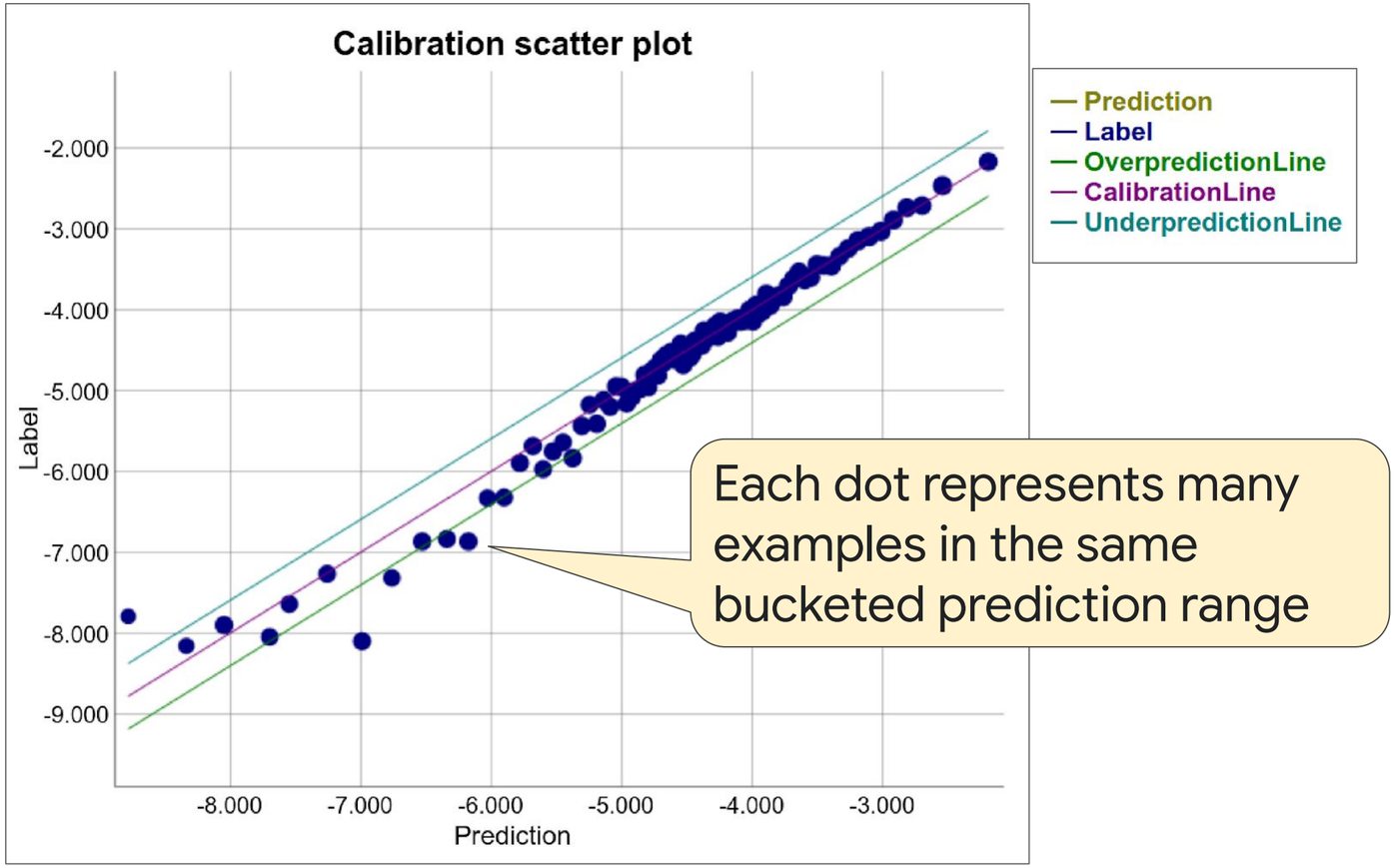
Logistic Regression predictions should be unbiased

average of predictions == average of observations

Look for bias in slices of data, this can guide improvements.



Use calibration plots of bucketed bias to find slices where your model performs poorly



Logistic Regression Quiz

Which of these is important when performing logistic regression?

A. Adding regularization
B. Choosing a tuned threshold
C. Checking for bias
D. All of the above

Logistic Regression Quiz

Which of these is important when performing logistic regression?

A. Adding regularization
B. Choosing a tuned threshold
C. Checking for bias
D. All of the above

Agenda

Supervised Learning

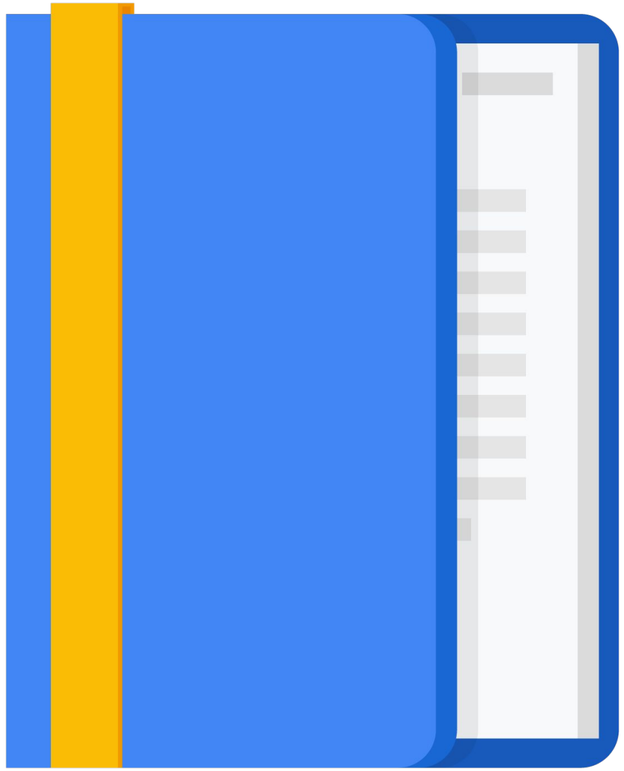
Lab: Introduction to Linear Regression

BigQuery Machine Learning

Lab: Creating BigQuery ML models for Taxifare Prediction

Logistic Regression

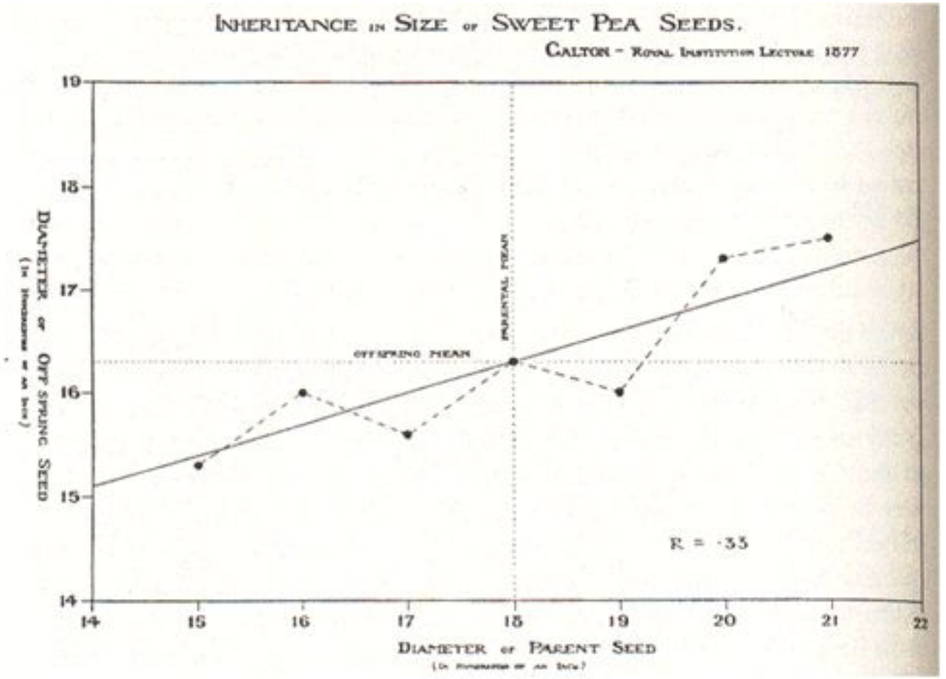
Short History of Machine Learning



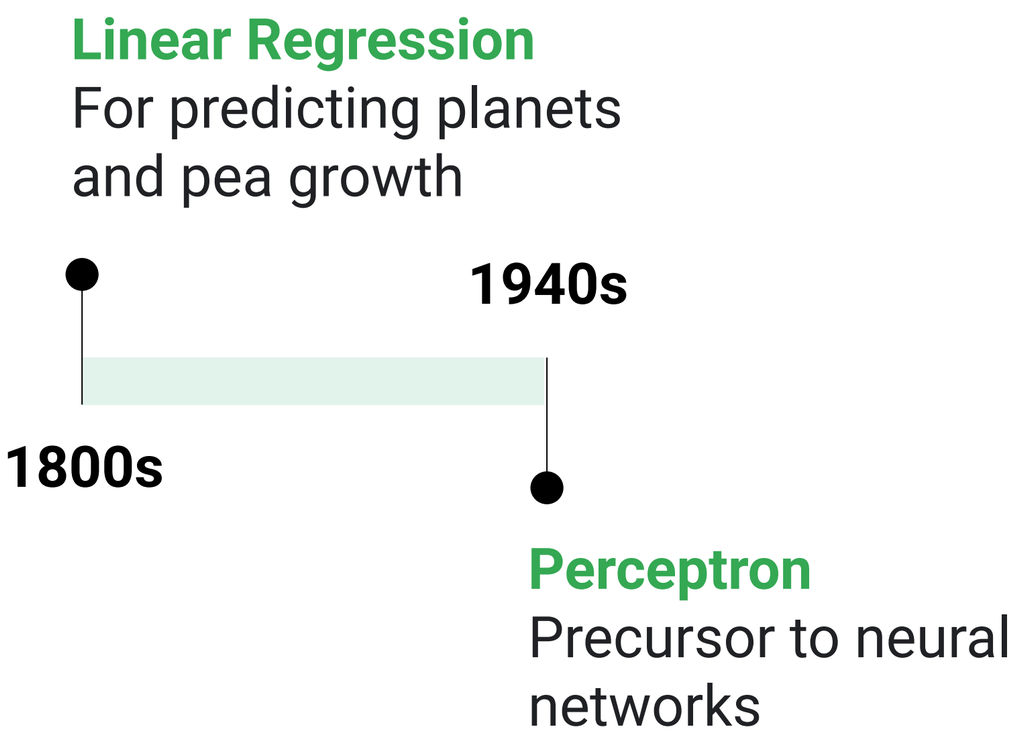
Linear regression was invented when computations were done by hand, but it continues to work well for large datasets

Linear Regression
For predicting planets and pea growth

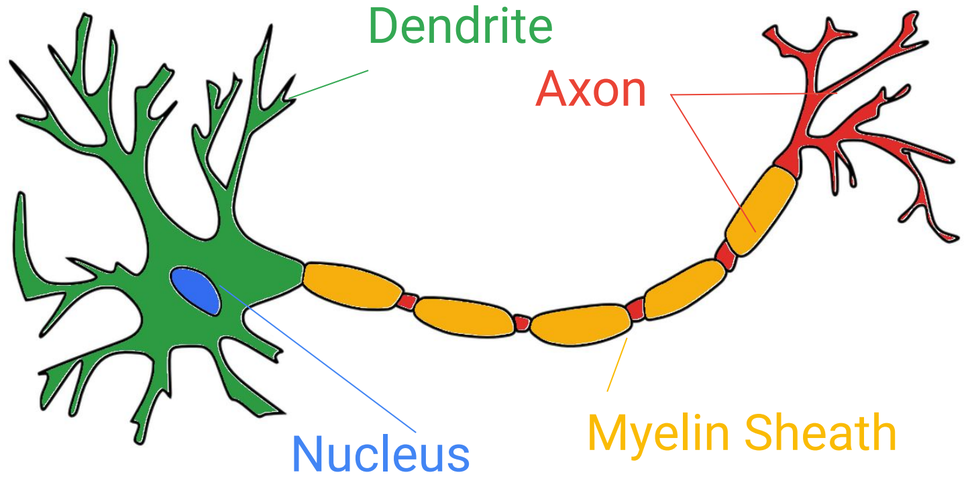
1800s

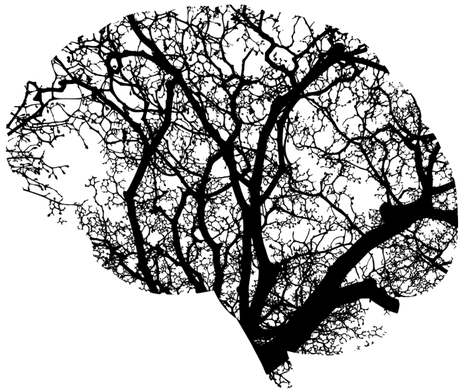


The perceptron was a computational model of a neuron

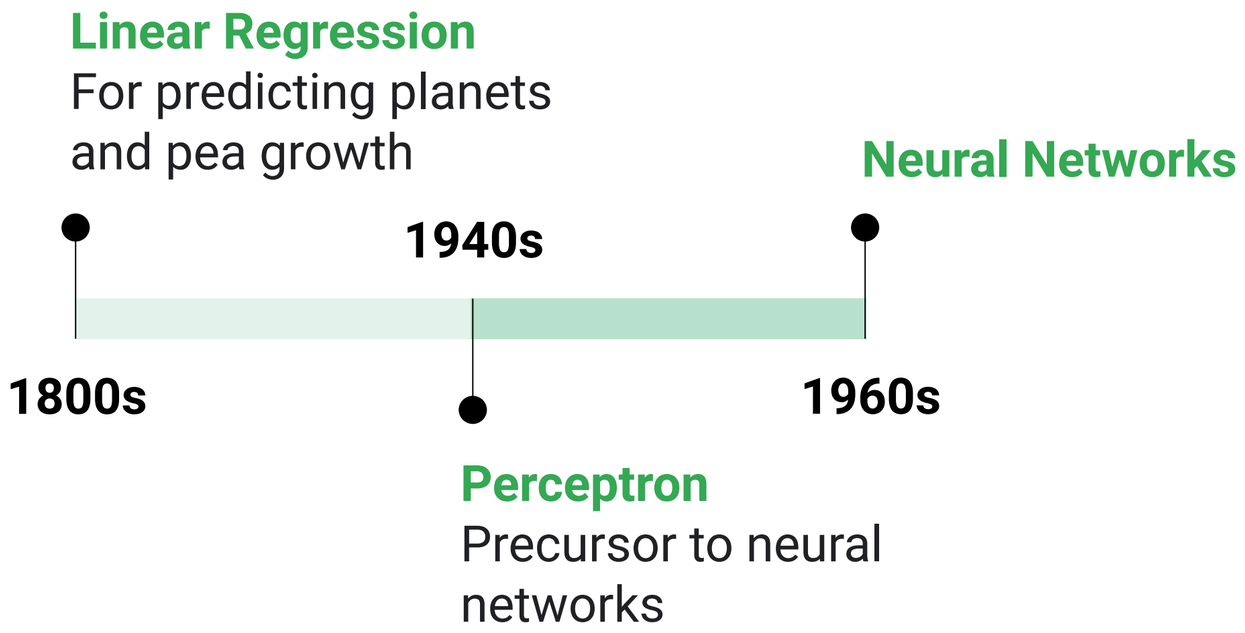


Perceptron motivation: Neurons

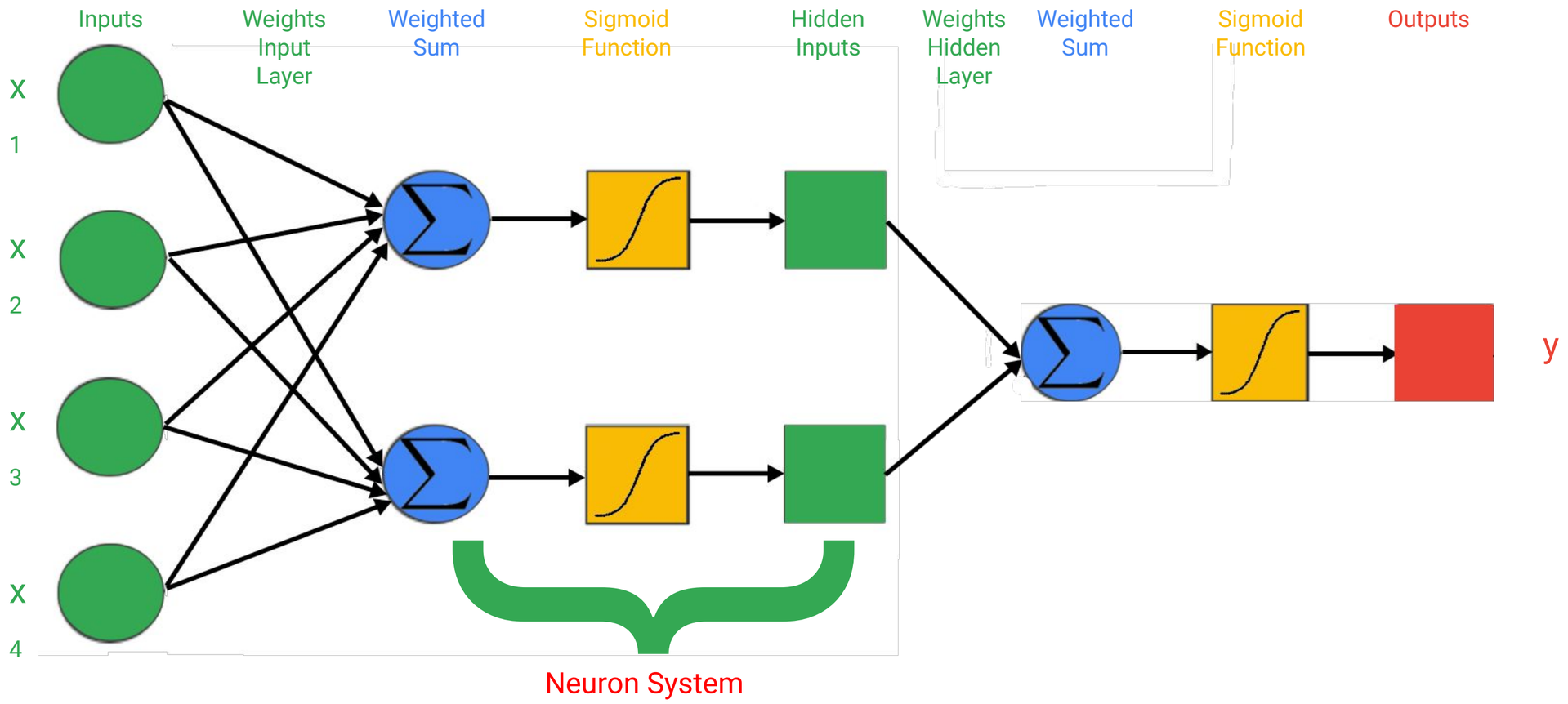




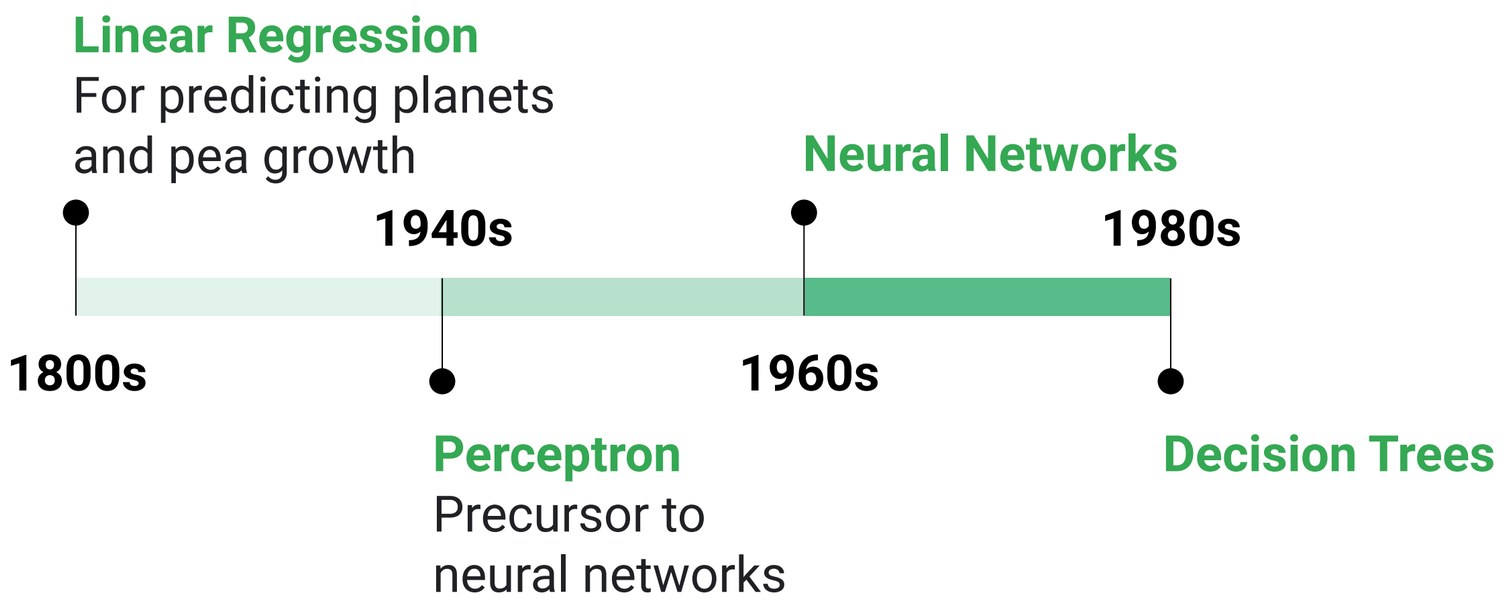
Neural networks combine layers of perceptrons, making them more powerful but also harder to train effectively



Neural networks: Multi-layer perceptron

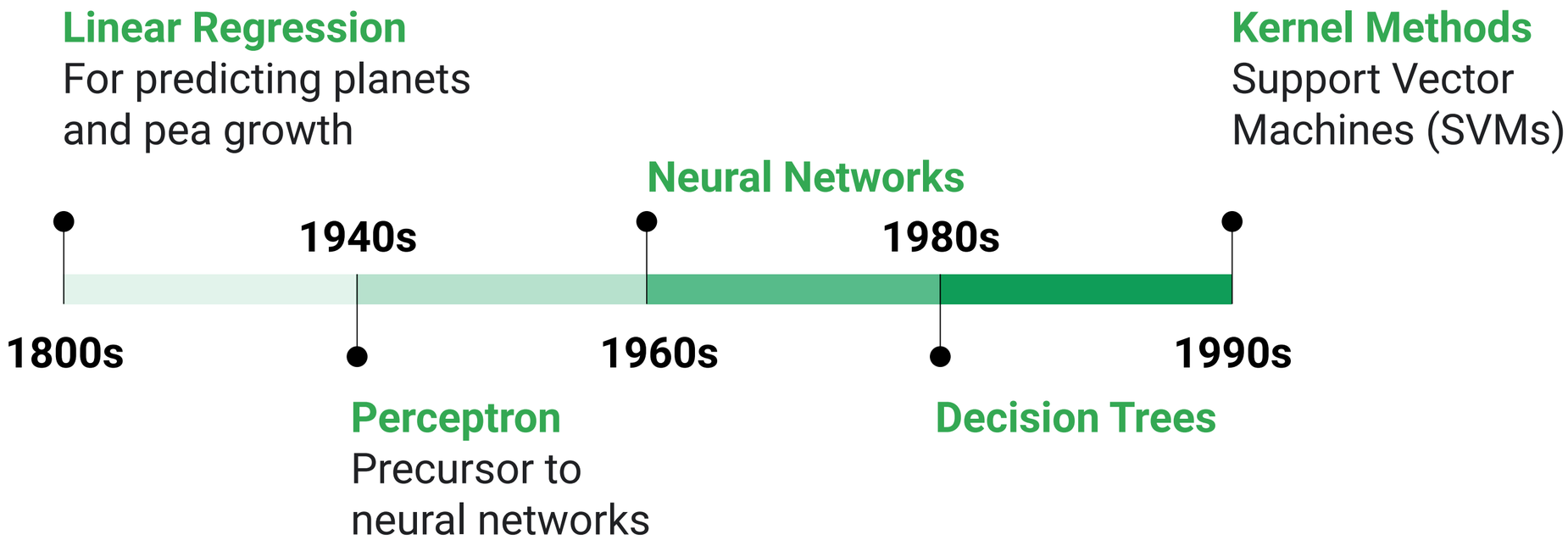


Decision trees build piecewise linear decision boundaries, are easy to train, and are easy for humans to interpret



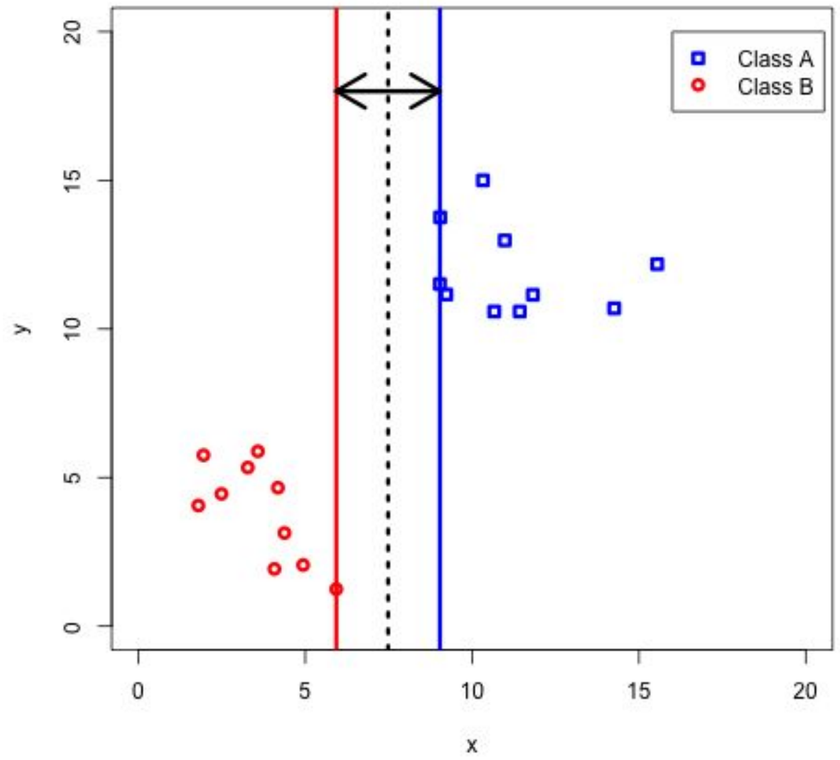
Decision trees and the Titanic

Support vector machines are nonlinear models that build maximum marginal boundaries in hyperspace

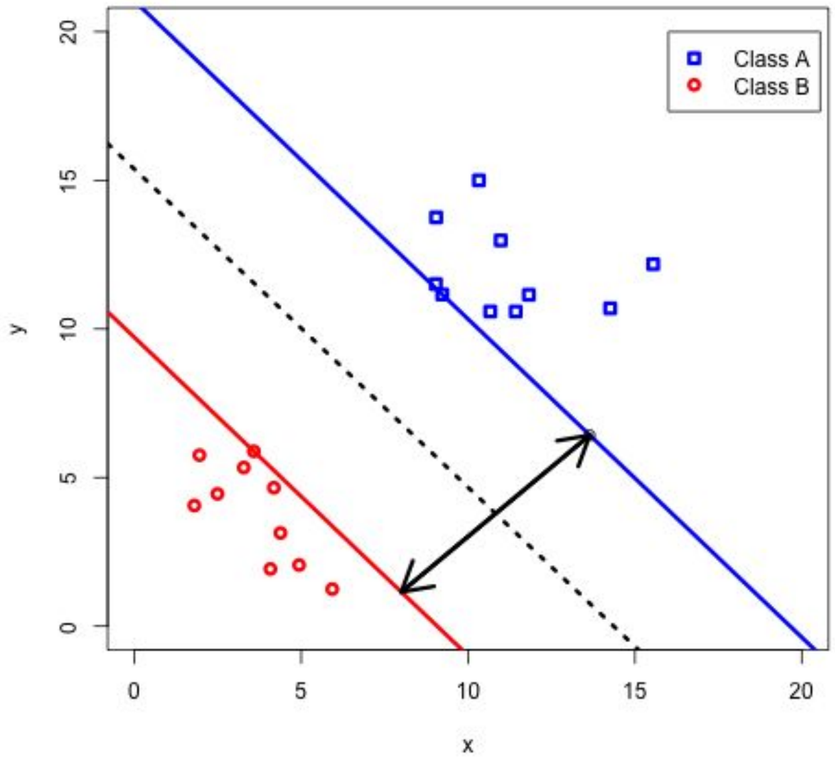


SVMs maximize the margin between two classes

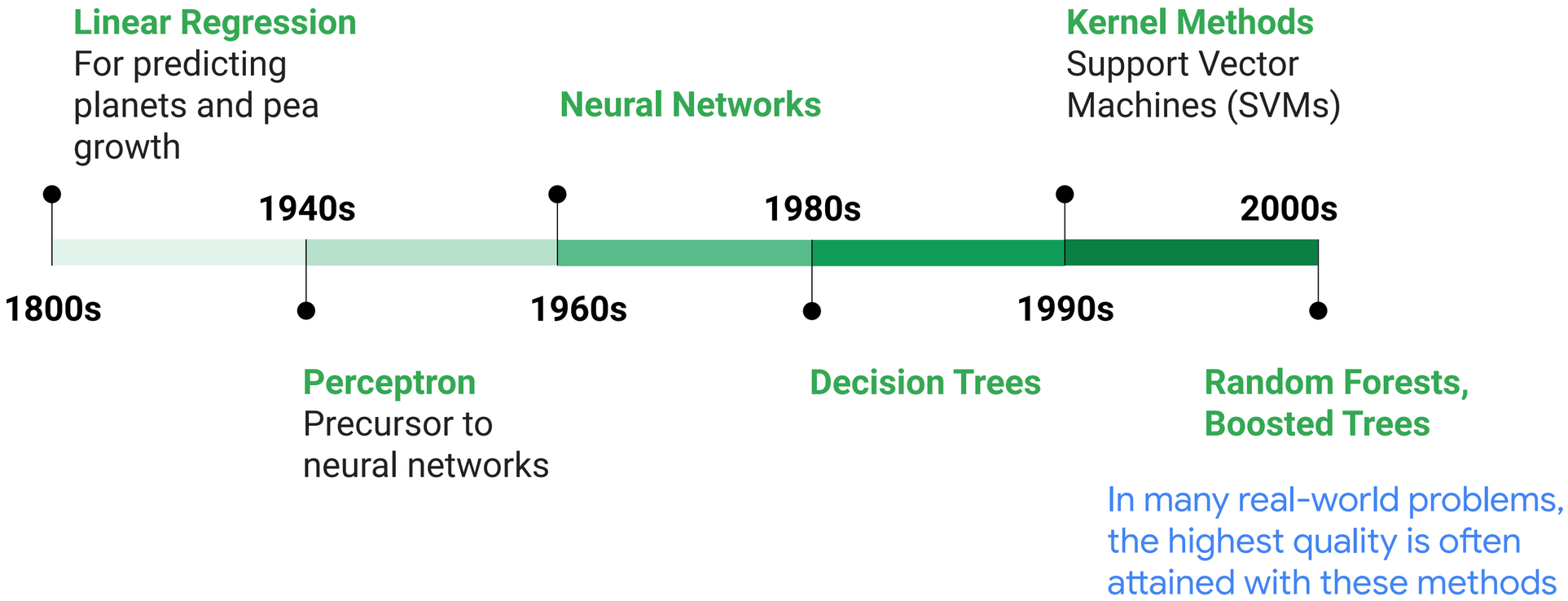
Small margin



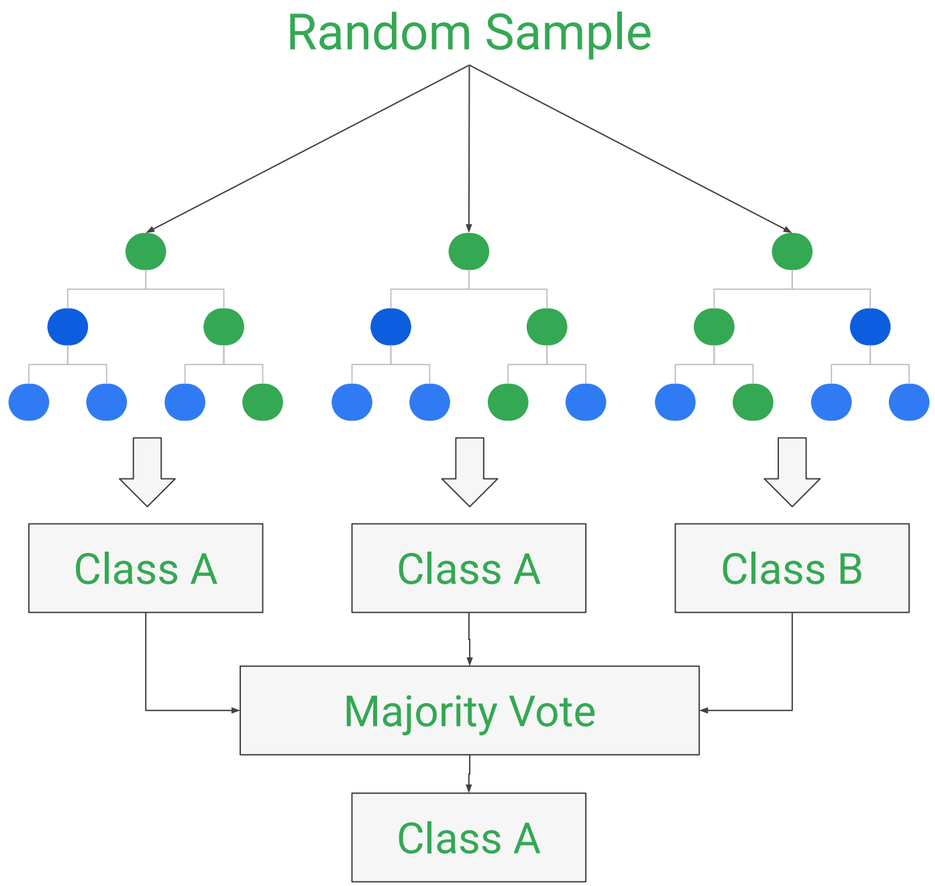
Large margin



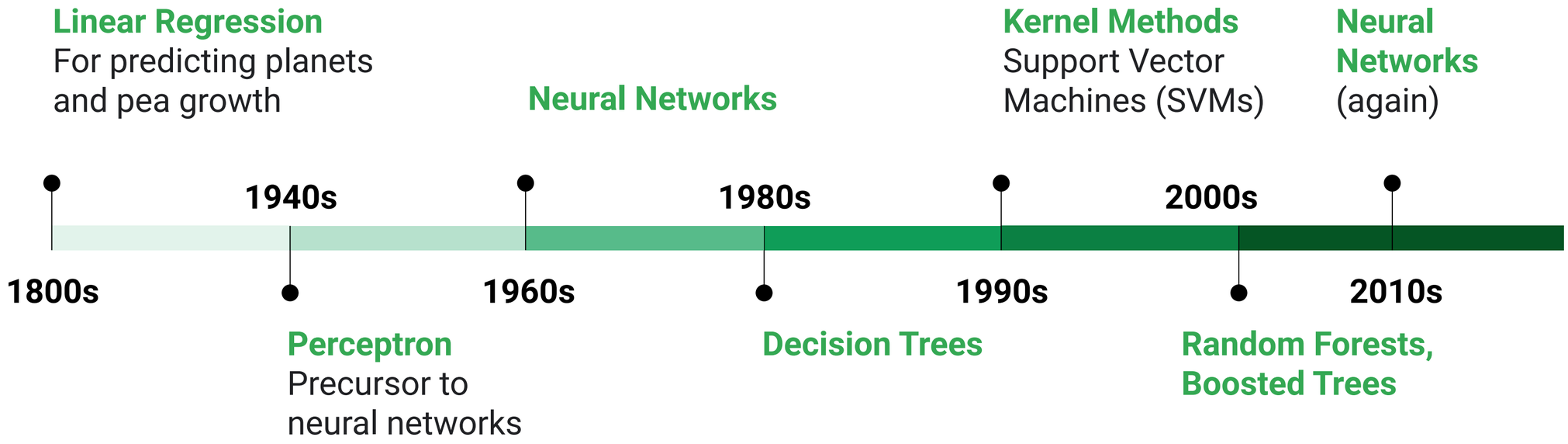
Random forests, bagging, and boosting are very effective predictors built by combining lots of very simple predictors



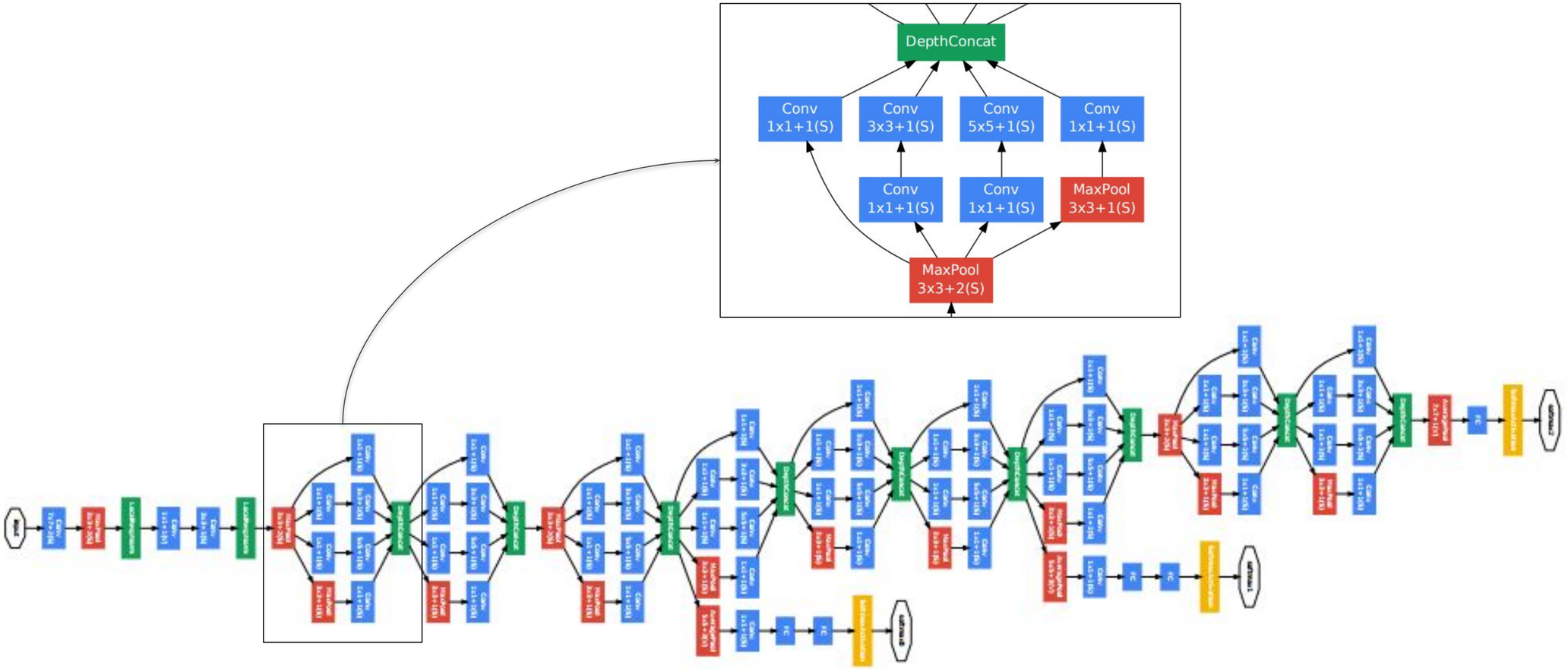
Random forest: Strong learner from many weak learners



With the advantage of technical improvements, more data, and computational power, neural networks made a comeback

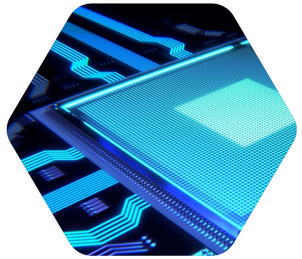


Inception/GoogLeNet Deep Neural Network



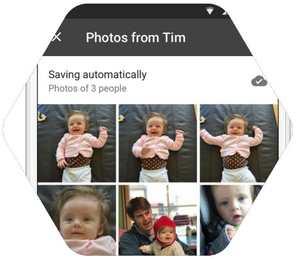
Neural networks are outperforming most other approaches in many domains

Large amounts of data



Available Computational Power

Available Infrastructure



Tasks and Goals we care about

Note that there are no models that are universally better, they're just different.

Launching into ML: Optimization

Learn how to ...

Quantify model performance using loss functions.

Use loss functions as the basis for an algorithm called gradient descent.

Optimize gradient descent to be as efficient as possible.

Use performance metrics to make business decisions.

Agenda

Defining ML Models

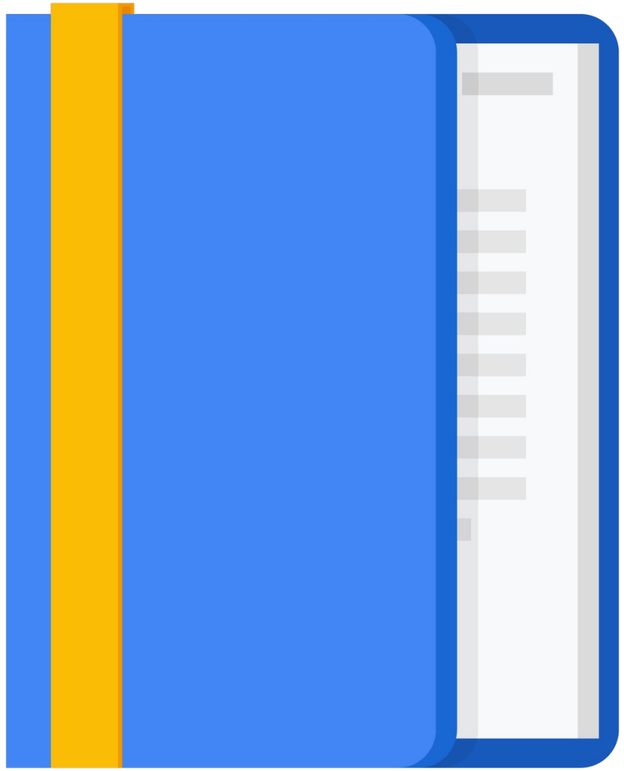
Introducing Loss Functions

Gradient Descent

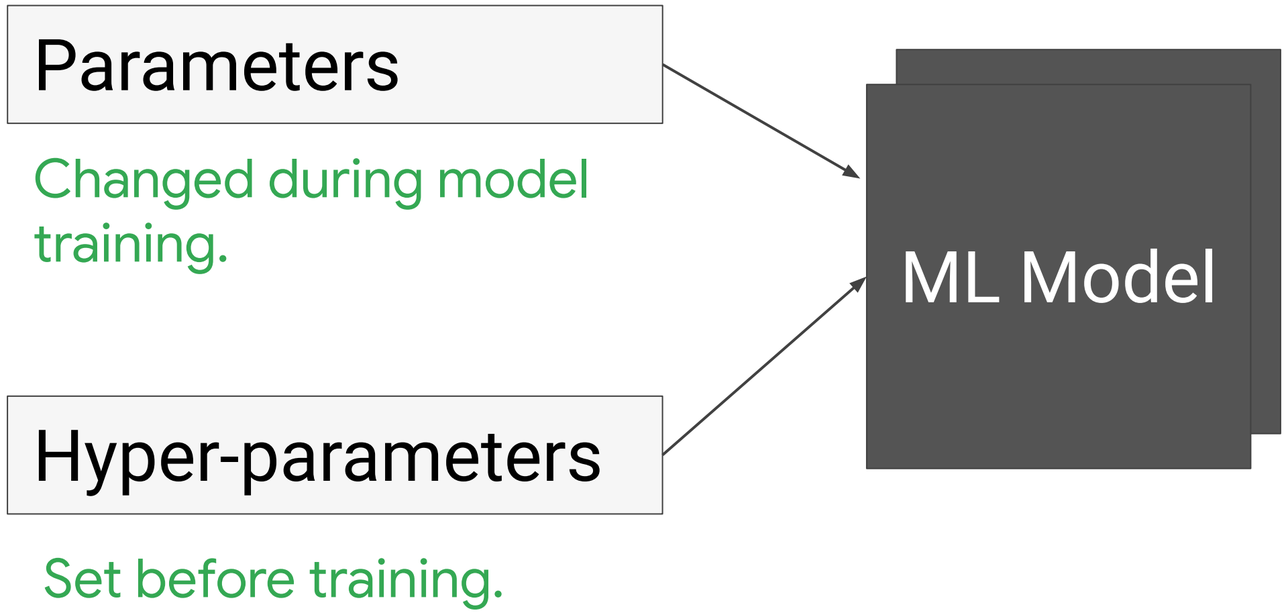
TensorFlow Playground

Lab: Develop an Intuitive Understanding of Neural Networks Using Tensorflow Playground

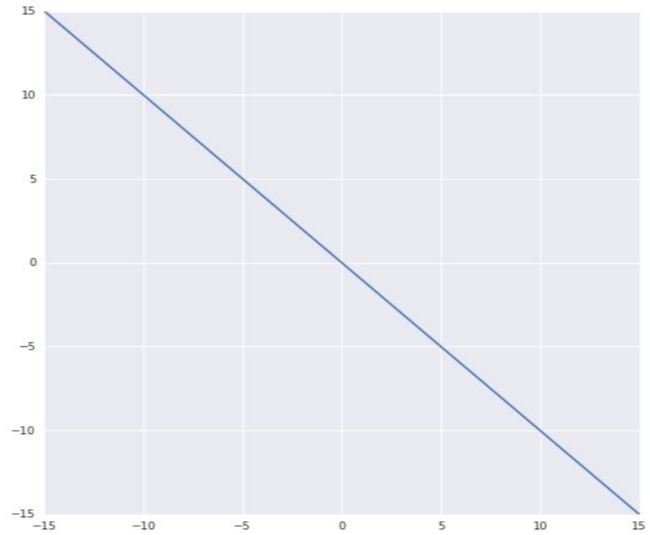
Performance Metrics



ML models are mathematical functions with parameters and hyper-parameters



Linear models have two types of parameters: Bias and weight



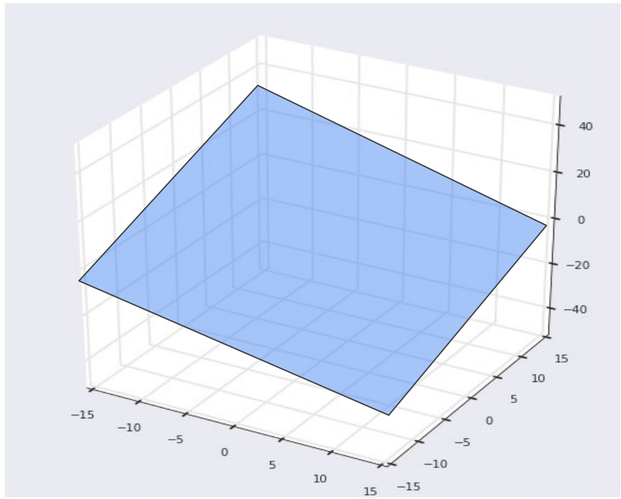
Linear

Output Bias Term Input Weight

$$y = b + x \times m$$

$$y = b + X \times w$$

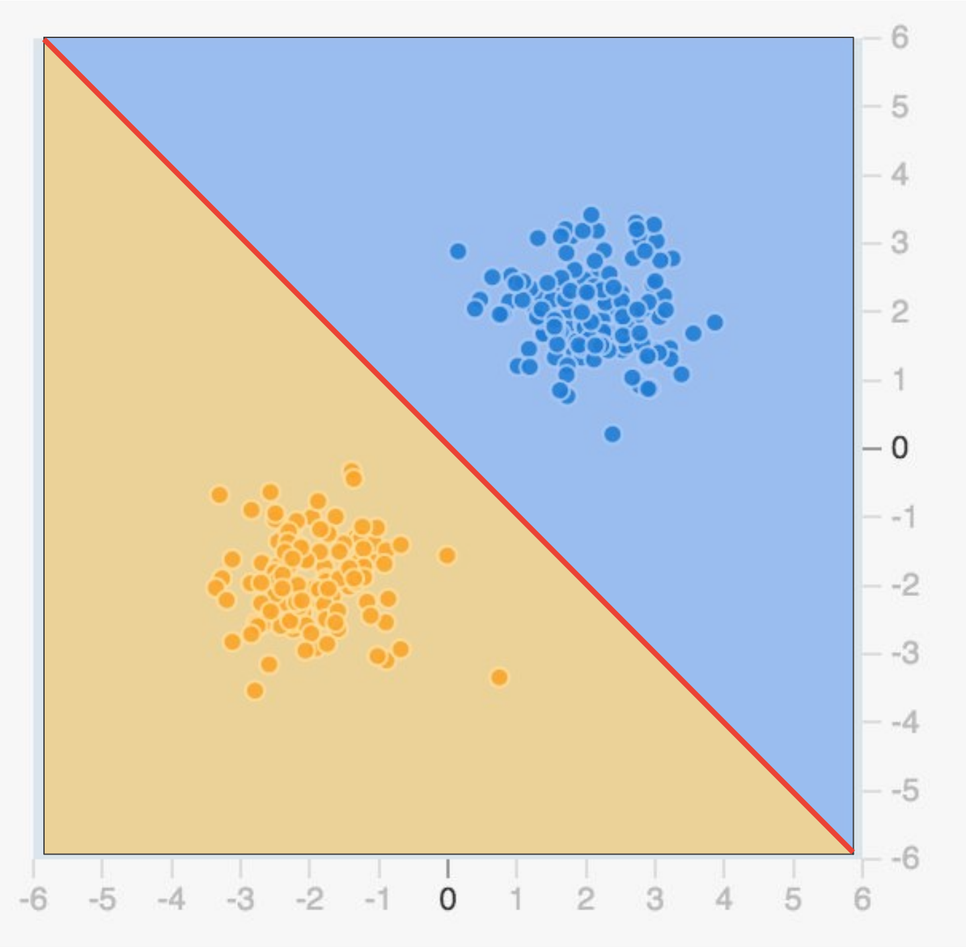
Model Parameters



Hyperplane

How can linear models classify data?

Classification explained graphically.



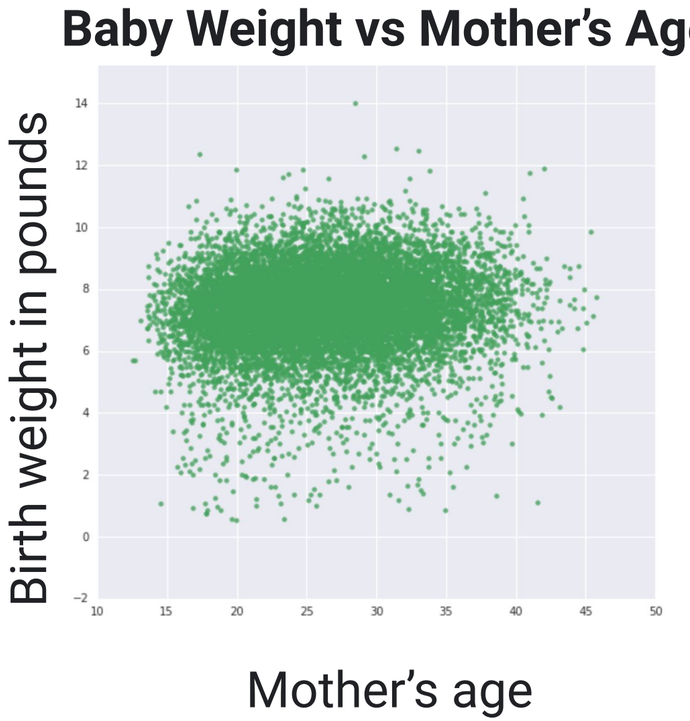
How do we predict a baby's health *before* they are born?

Which of these could be a *feature* in your model?

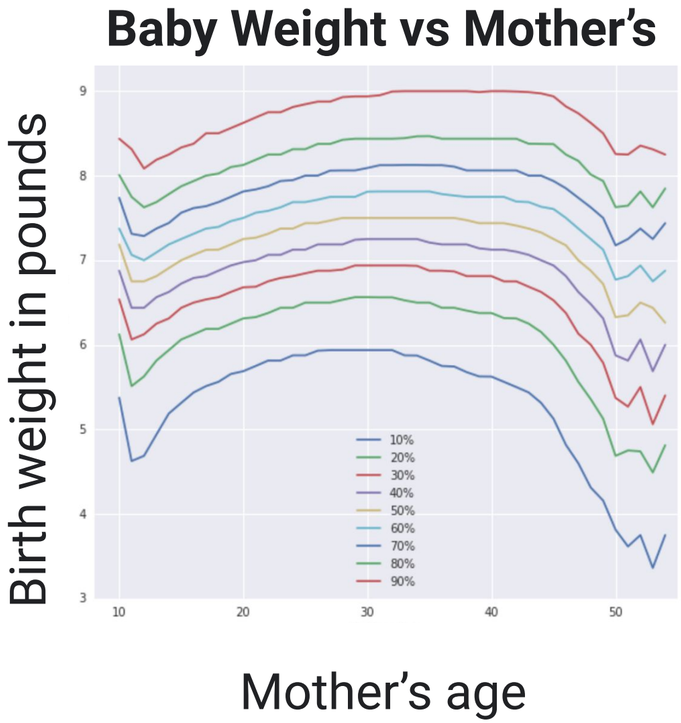
A. Mother's Age
B. Birth Time
C. Baby Weight

Which could be a *label?*

Exploring the data visually



Scatterplots are made from samples of large datasets rather than from the whole dataset.



Graph representing groups of data, specifically, quantiles.

Equation for a linear model tying mother's age and baby weight

The slope of the line is given by w1.

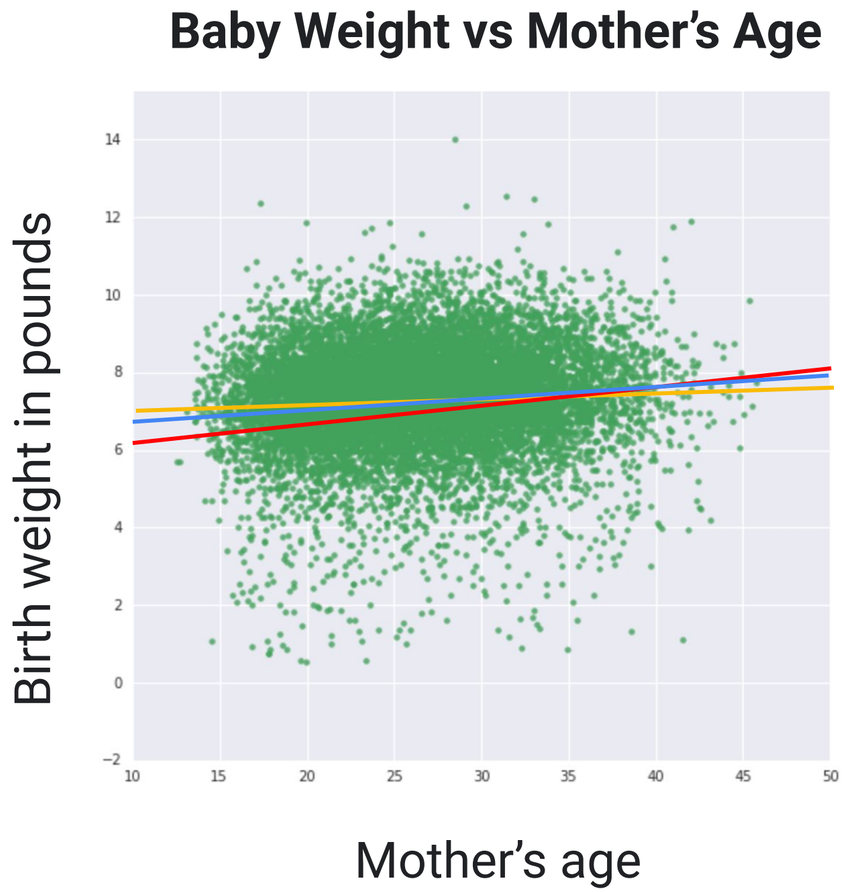
$$y = w_1 x_1 + b$$

- x_1 is the **feature** (e.g. mother's age)
- w_1 is the **weight** for x_1

Line: y= .02x + 6.83

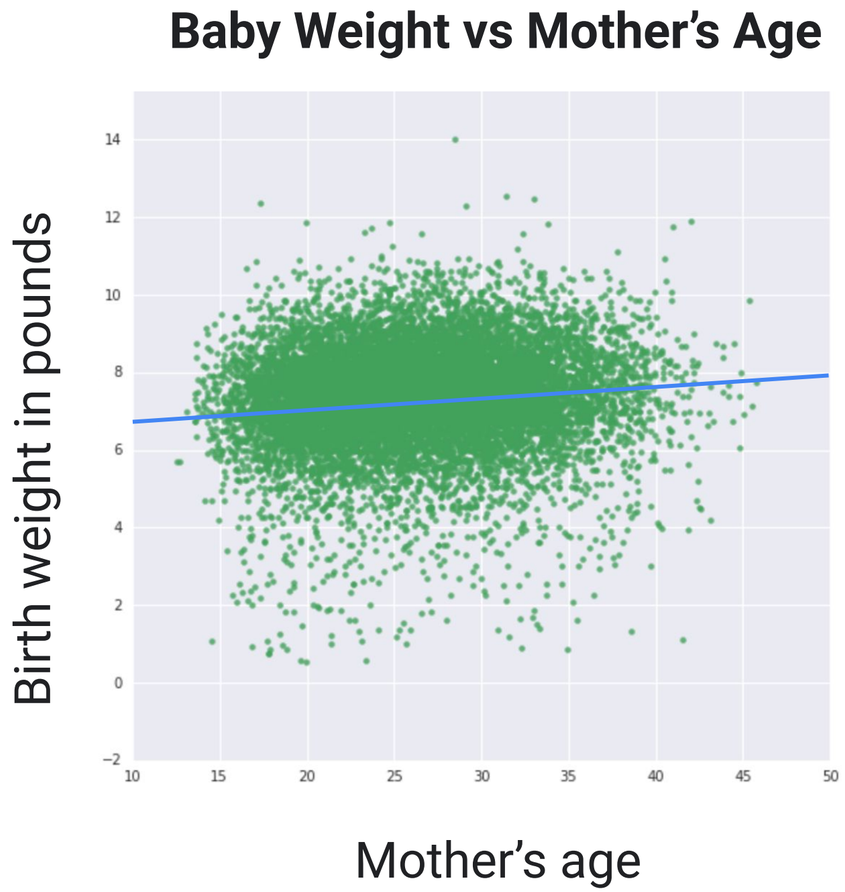
Line: y= .03x + 6.49

Line: y= .01x + 7.14

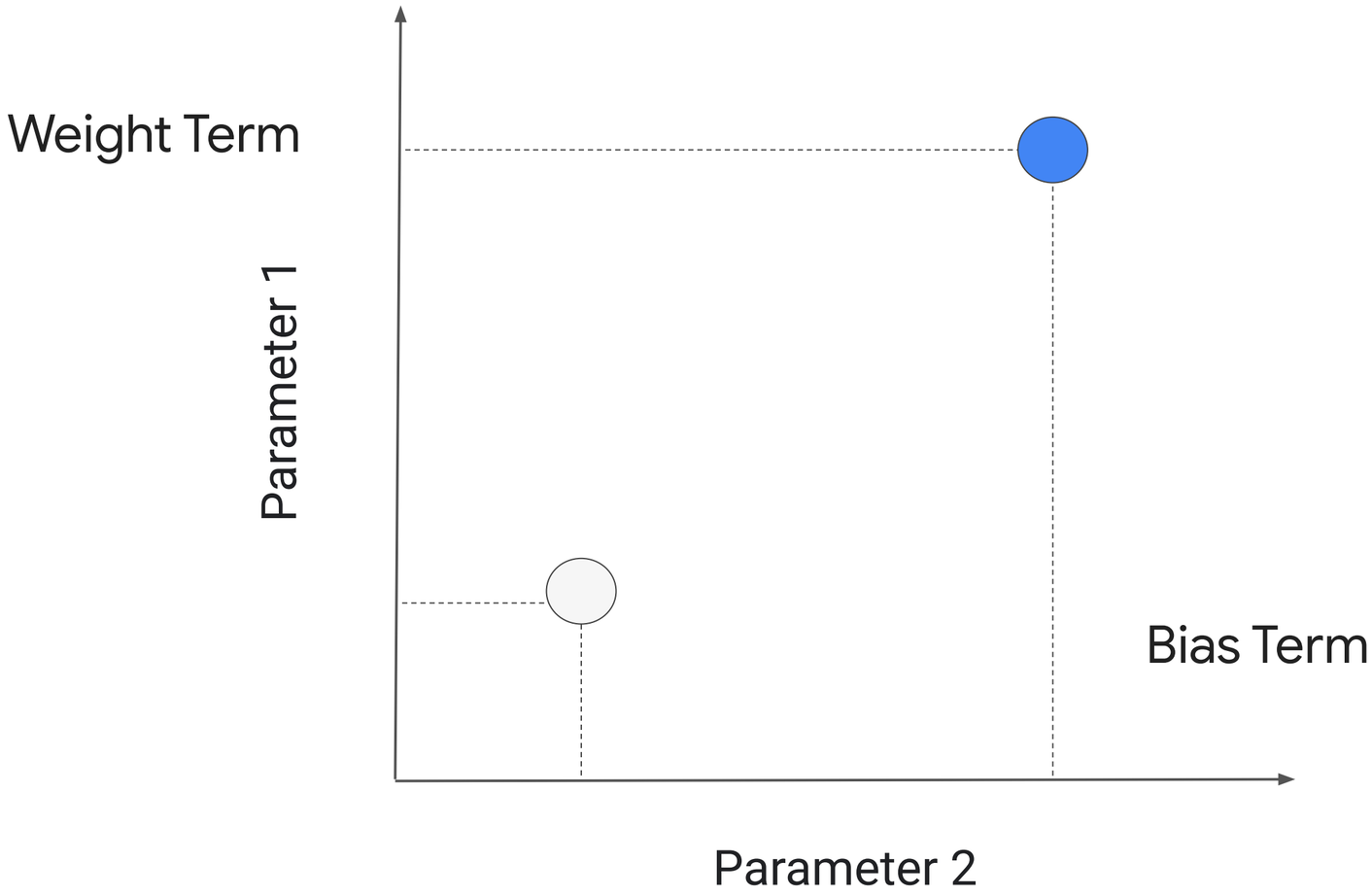


Can’t we just solve the equation using all the data?

When an analytical solution is no longer an option, you use gradient descent.



Searching in parameter-space



Agenda

Defining ML Models

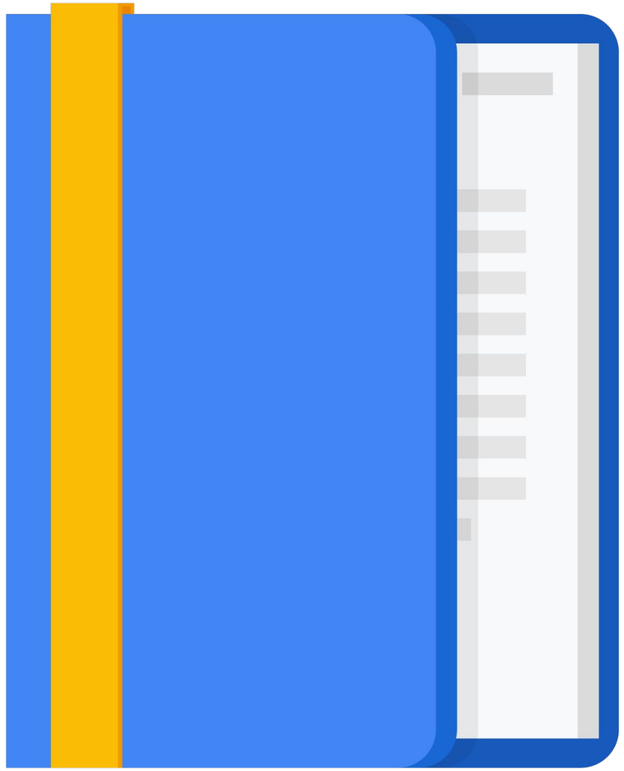
Introducing Loss Functions

Gradient Descent

TensorFlow Playground

Lab: Develop an Intuitive Understanding of Neural Networks Using Tensorflow Playground

Performance Metrics

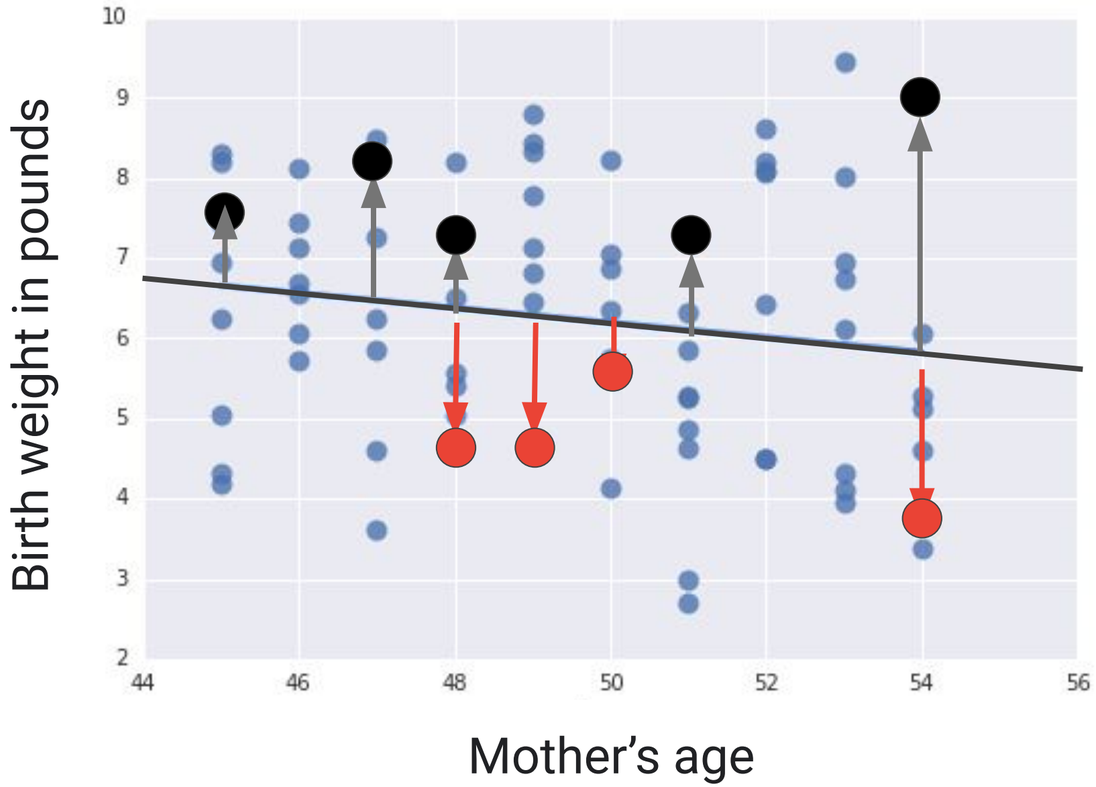


Compose a loss function by calculating errors

Error = actual (true) - predicted value
Compute the errors:

```
+0.70
+1.10
+0.65
-1.20
-1.15
+1.10
+3.09
-2.10
```

Each error makes sense. How about all the errors added together?



One loss function metric is Root Mean Squared Error (RMSE)

1 Get the errors for the training examples.

```
+0.70
+1.10
+0.65
-1.20
-1.15
+1.10
+3.09
-2.10
```

2 Compute the squares of the error values.

```
0.49
1.21
0.42
1.44
1.32
1.21
9.55
4.41
```

3 Compute the mean of the squared error values.

```
2.51
```

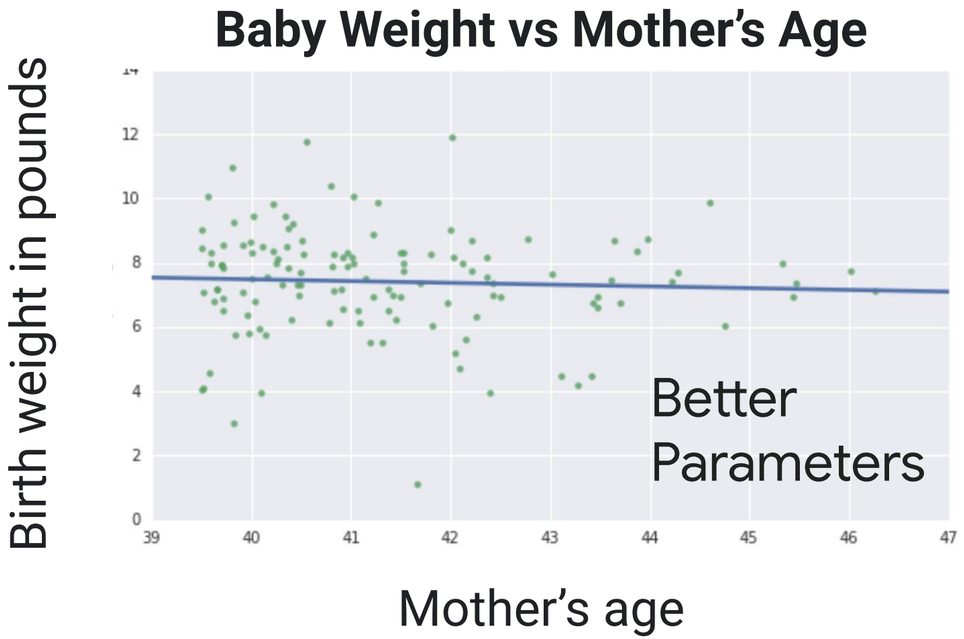
$$\sqrt{\frac{1}{n} \times \sum_{i=1}^{n} (\hat{Y}_i - Y_i)^2}$$

$\hat{Y}_i$ predicted value

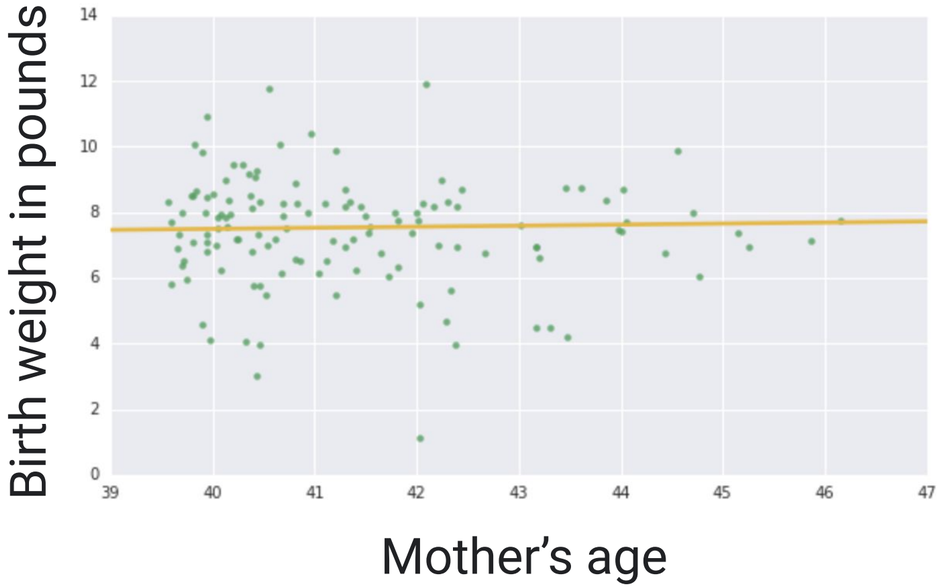
Y_i labeled value

4 Take a square root of the mean. **1.58**

Lower RMSE indicates a better performing model



RMSE=.145

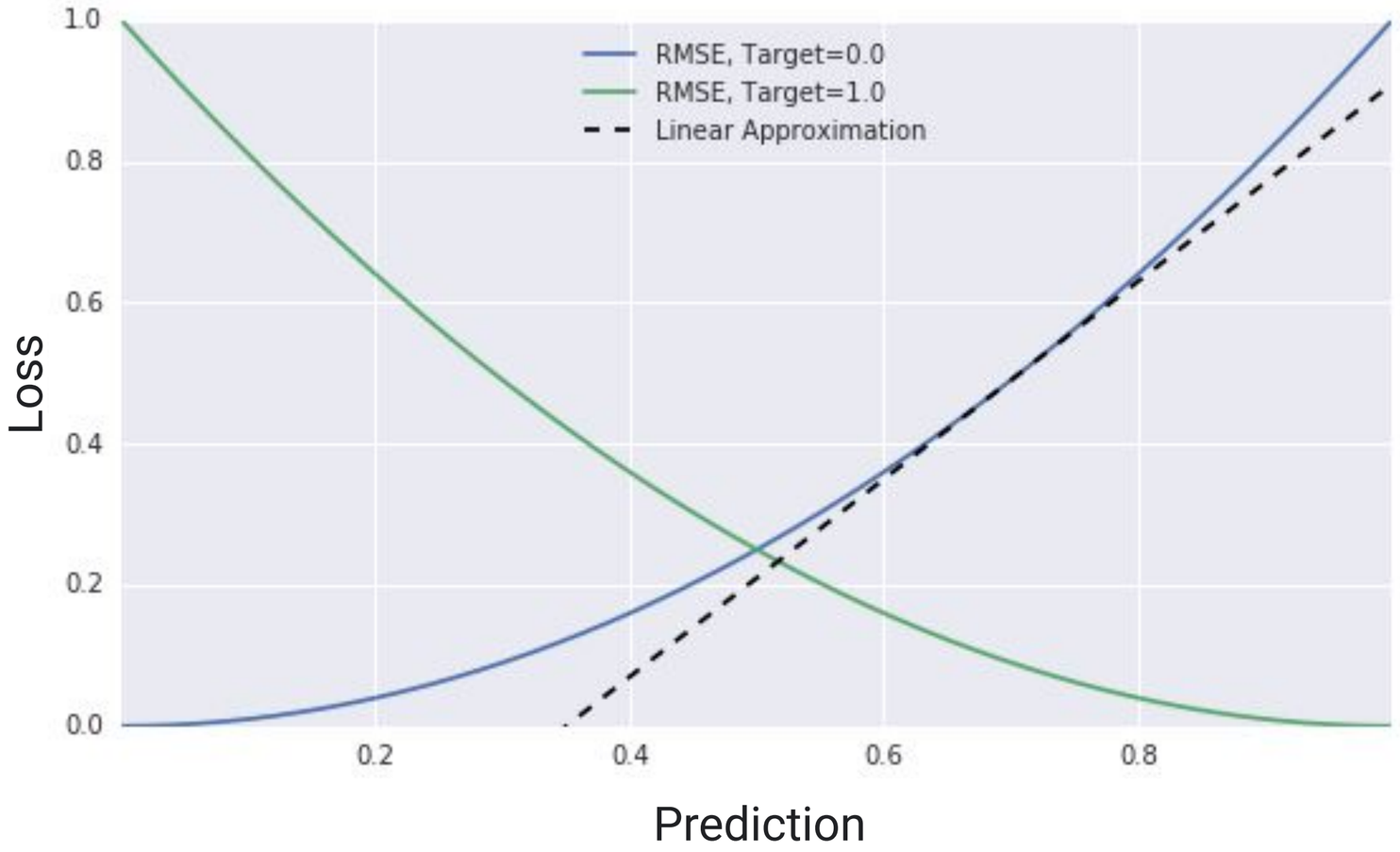


RMSE=.149

Need a way to find the best values for weight and bias.

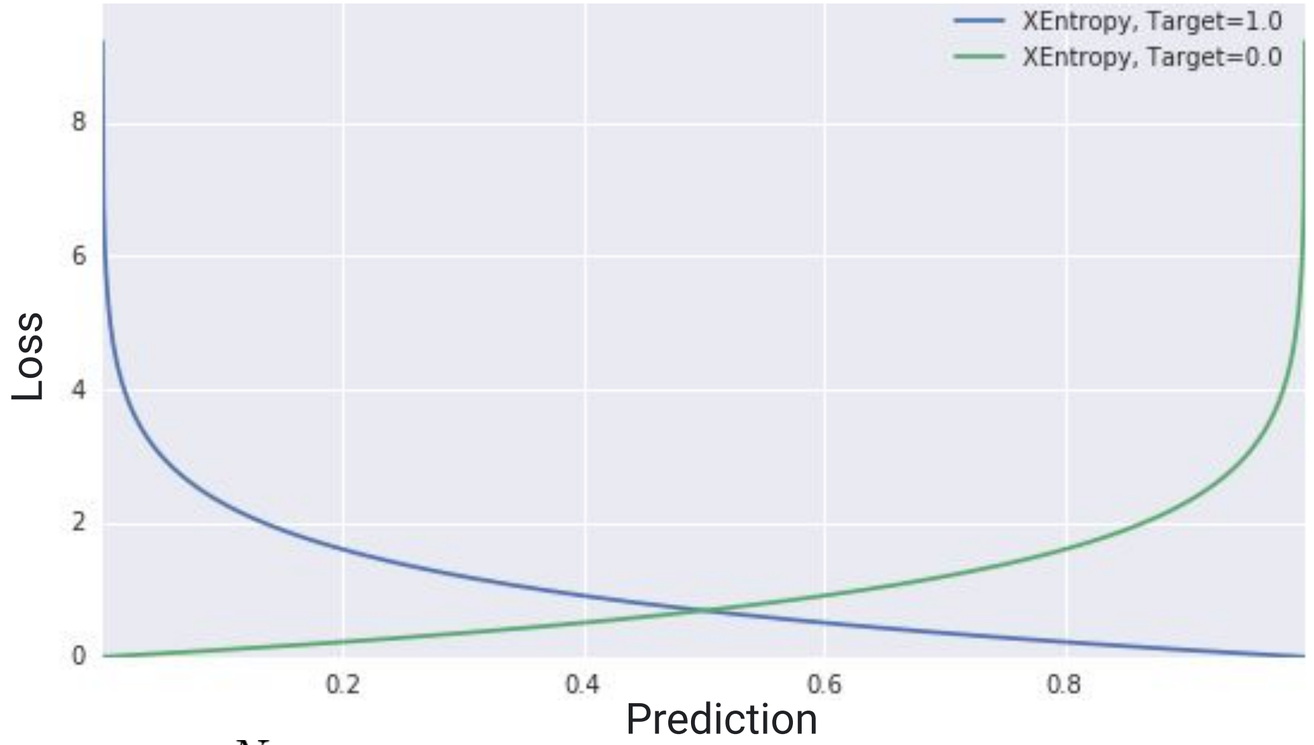
Problem: RMSE doesn't work as well for classification

RMSE doesn't penalize bad classifications appropriately.



Problem: RMSE doesn't work as well for classification

Bad classifications are penalized appropriately.



$$\frac{-1}{N} \times \sum_{1}^{N} y_i \times log(\hat{y}_i) + (1 - y_i) \times log(1 - \hat{y}_i)$$

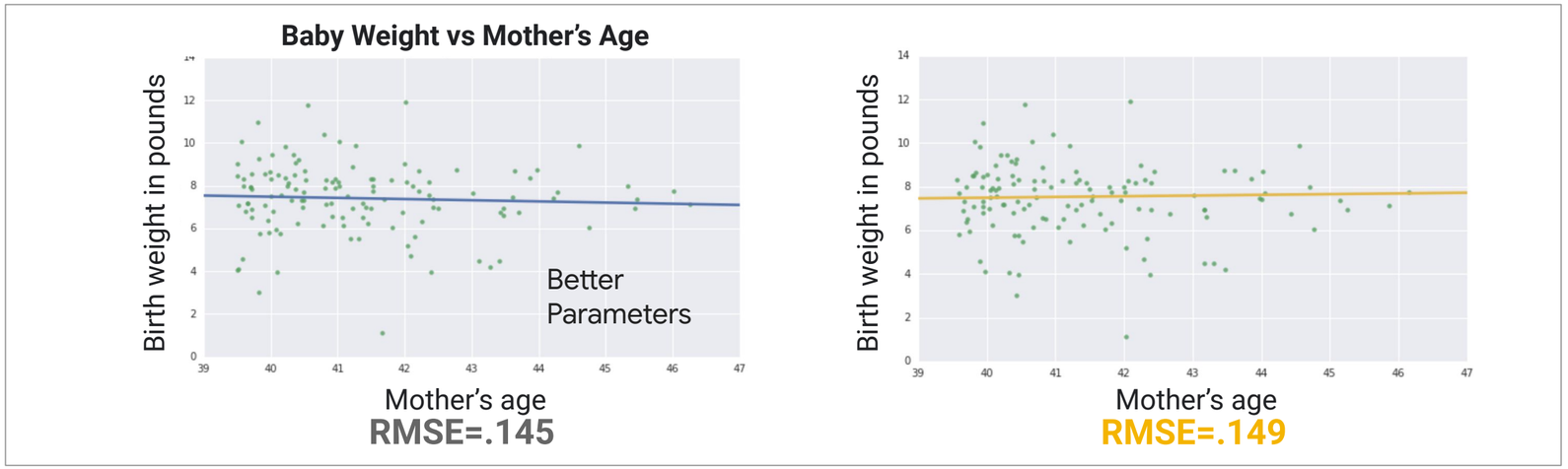
Computing cross-entropy loss

$$\frac{-1}{N} \times \sum_{1}^{N} \overbrace{y_i \times log(\hat{y}_i)}^{\text{Positive term}} + \overbrace{(1 - y_i) \times log(1 - \hat{y}_i)}^{\text{Negative term}}$$

X	Y_i	$\hat{Y}_i$
	1	.7
	0	.2

$$\left\langle \begin{array}{c} (1.0*\log(.7) + \cancel{(1-1.0)*\log(1-.7)} \\ + \\ \cancel{(0.0*\log(.2)} + (1-0.0)*\log(1-.2)) \end{array} \right\rangle *(-½)=.13$$

From loss functions to gradient descent



X	Y_i	$\hat{Y}_i$
	1	.7
	0	.2

Positive term Negative term

$$\frac{-1}{N} \times \sum_{1}^{N} y_i \times log(\hat{y}_i) + (1 - y_i) \times log(1 - \hat{y}_i)$$

(1.0*log(.7) + ~~(1-1.0)*log(1-.7)~~
+
~~(0.0*log(.2)~~ + (1-0.0)*log(1-.2))

*(-½)=.13

Agenda

Defining ML Models

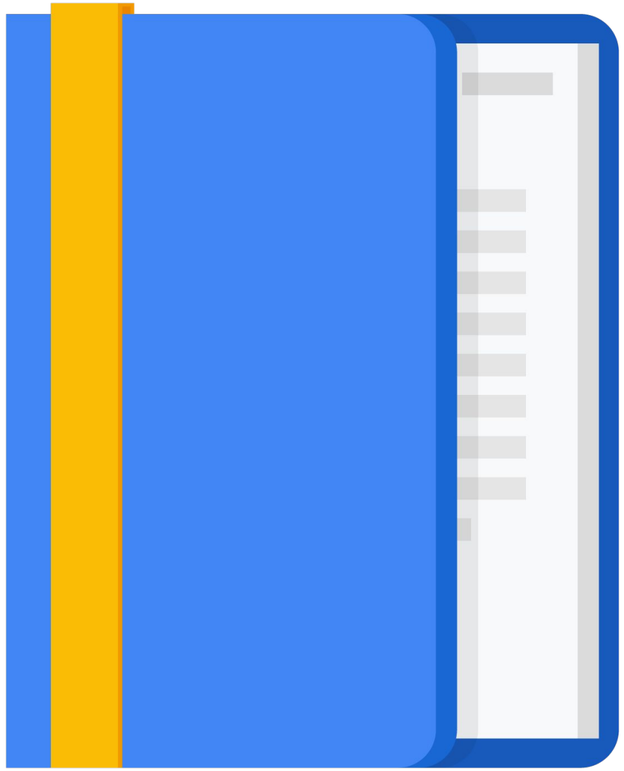
Introducing Loss Functions

Gradient Descent

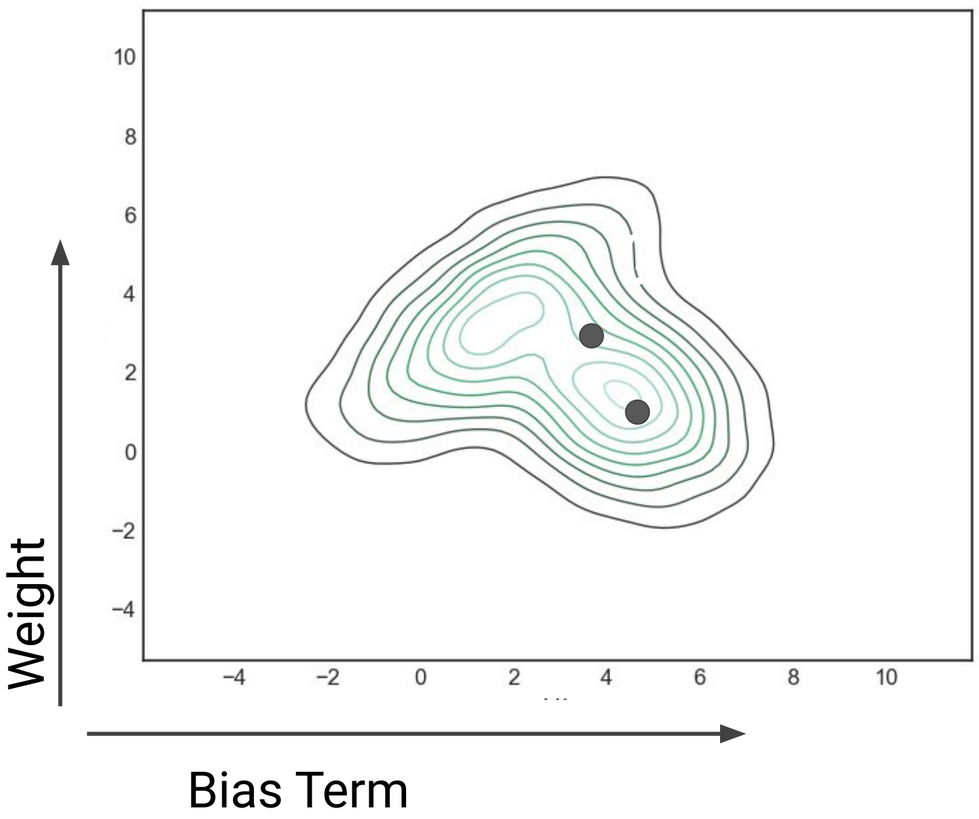
TensorFlow Playground

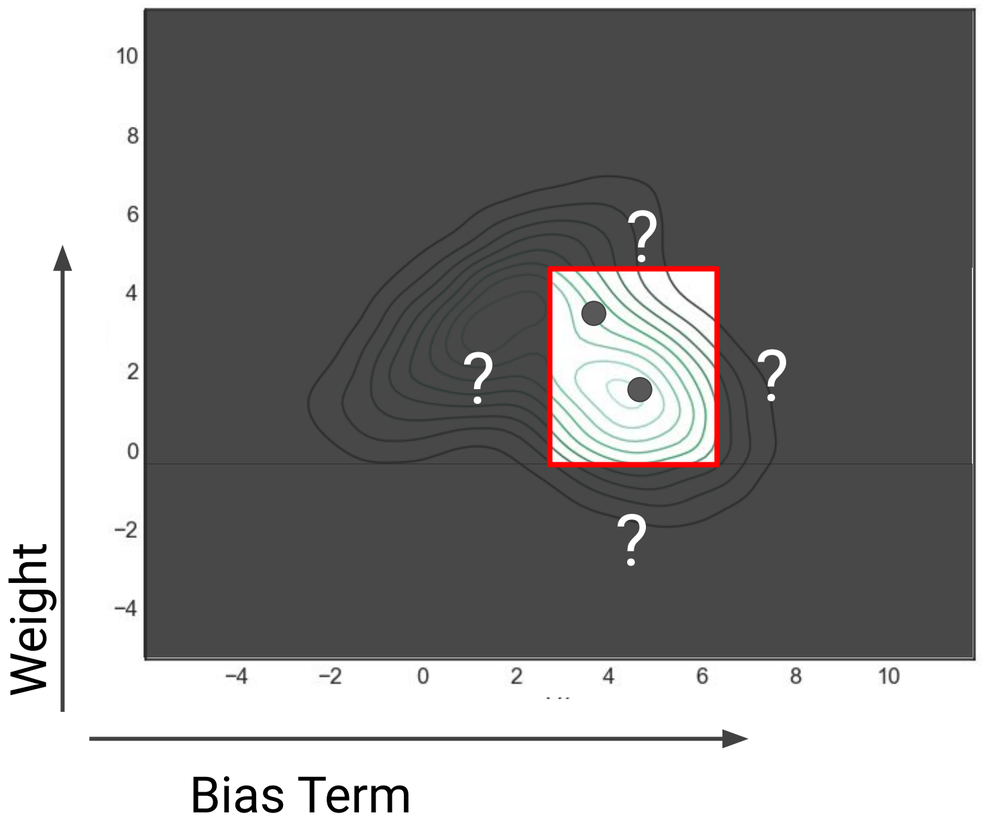
Lab: Develop an Intuitive Understanding of Neural Networks Using Tensorflow Playground

Performance Metrics



Loss functions lead to loss surfaces





Finding the bottom

Which direction should I head?

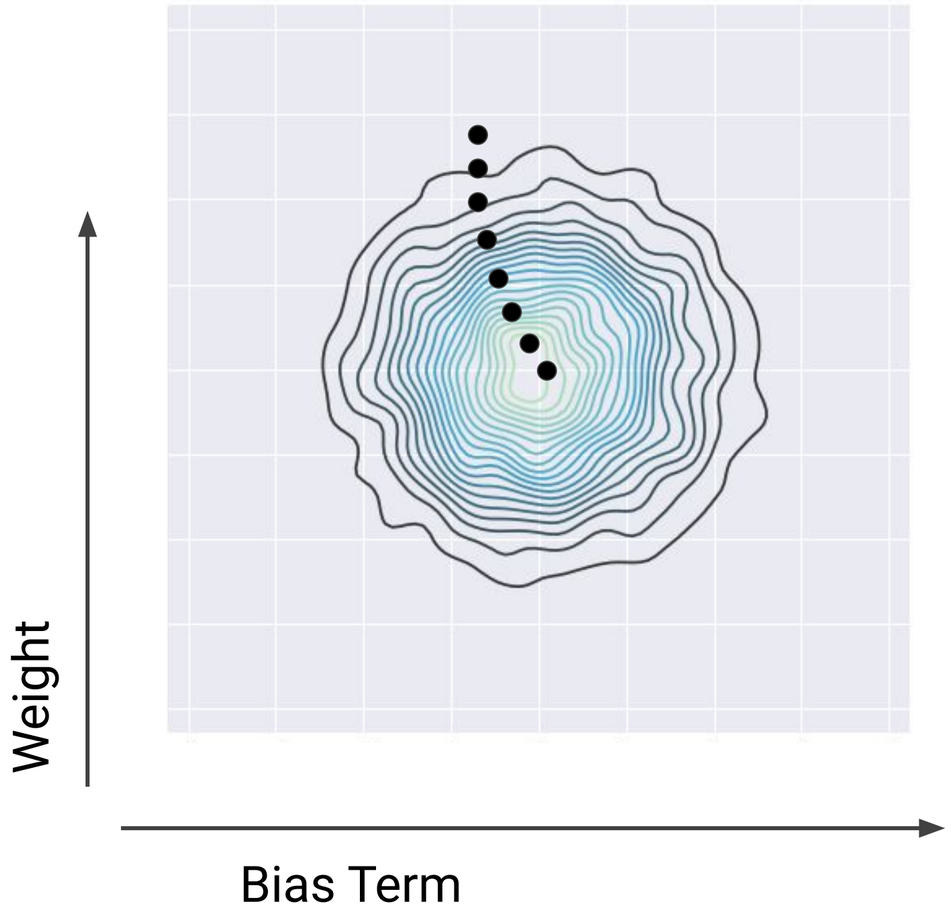
How large or small a step?

A simple algorithm to find the minimum

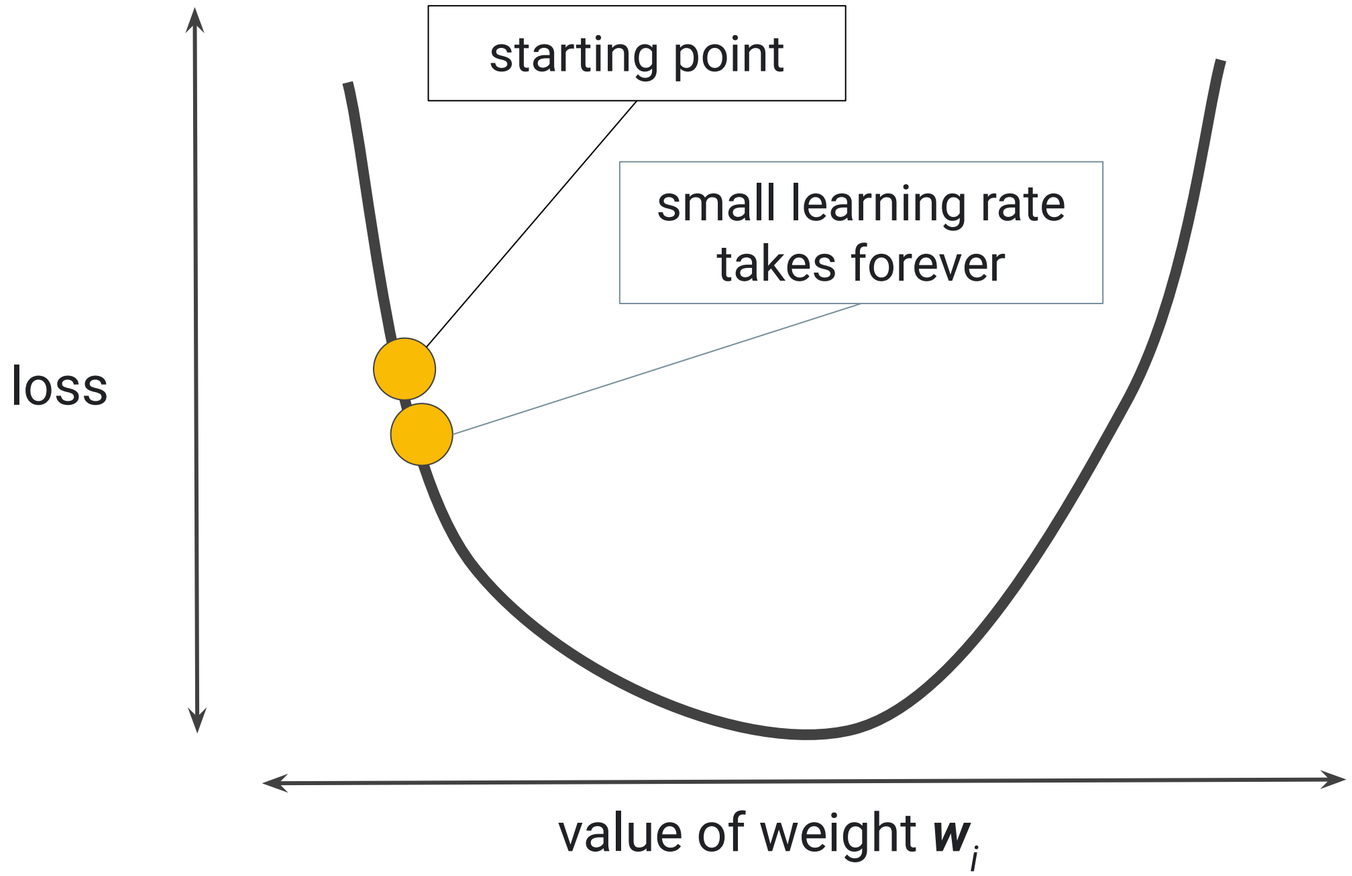
```
while loss is > Epsilon:
  direction = computeDirection()
  for i in range(self.params):
    self.params[i] = //
              self.params[i] //
                  + stepSize * direction[i]
  loss = computeLoss()
```

Epsilon = A tiny Constant

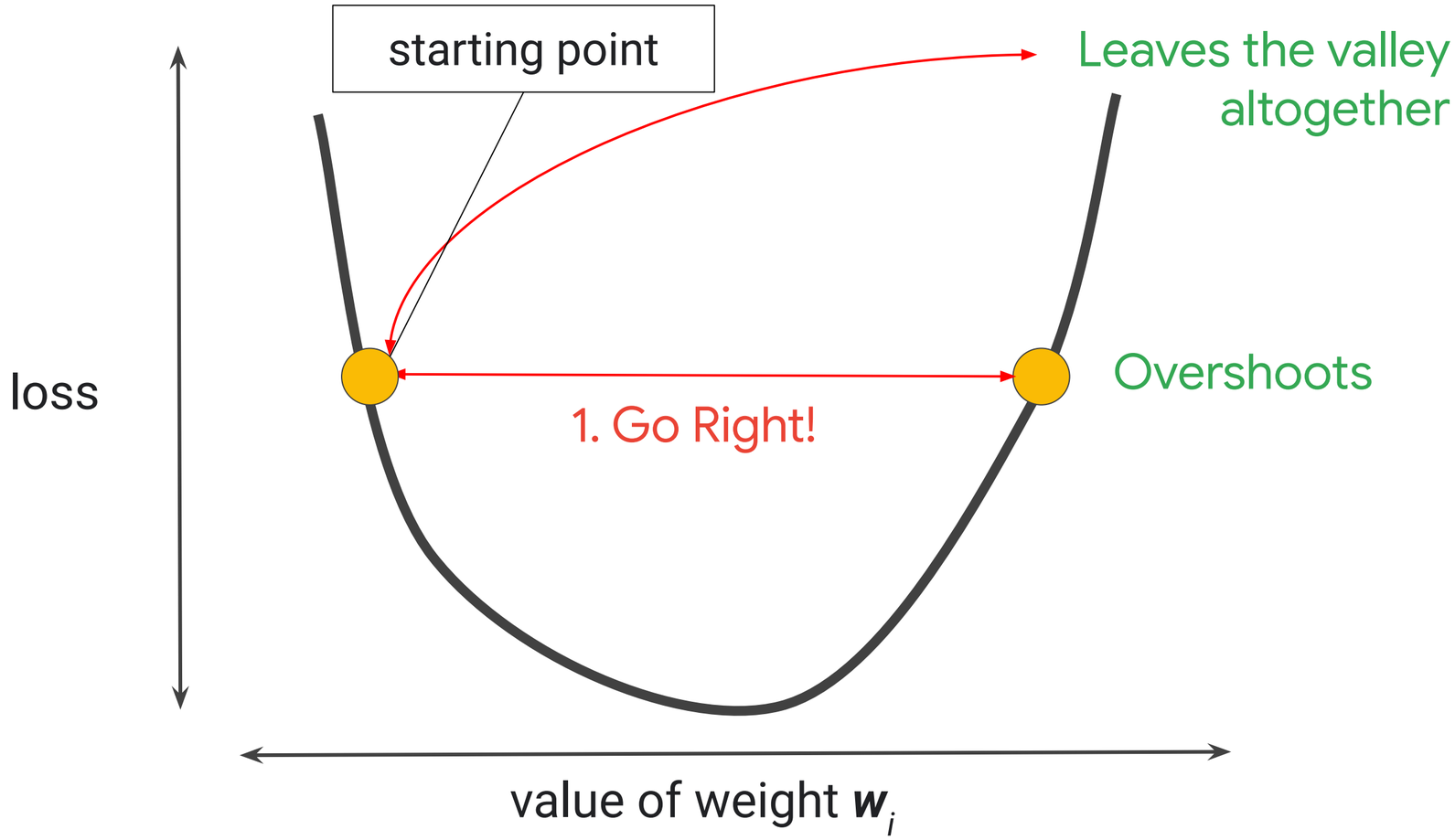
Search for a minima by descending the gradient



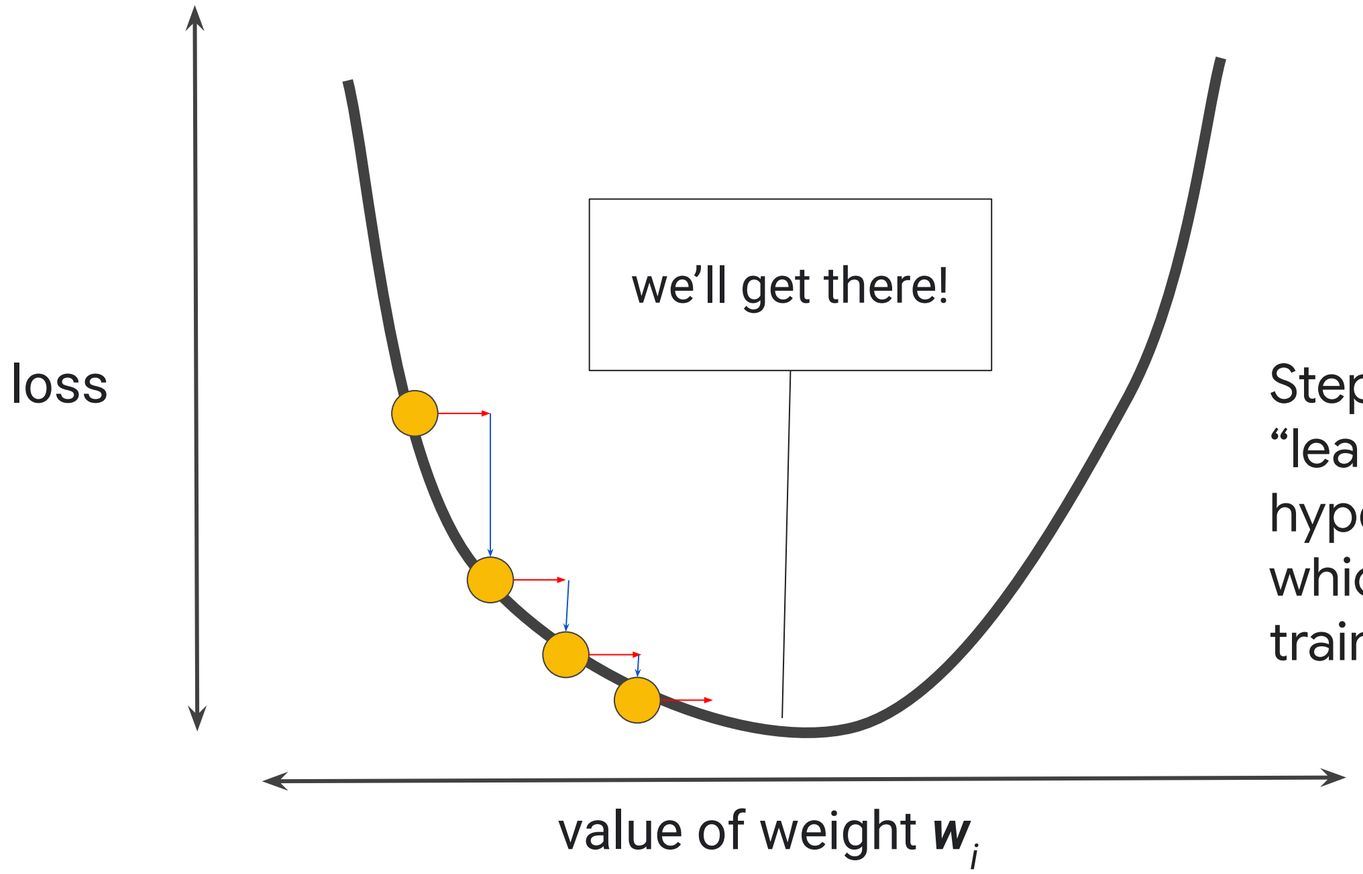
Small step sizes can take a very long time to converge



Large step sizes may never converge to the true minimum

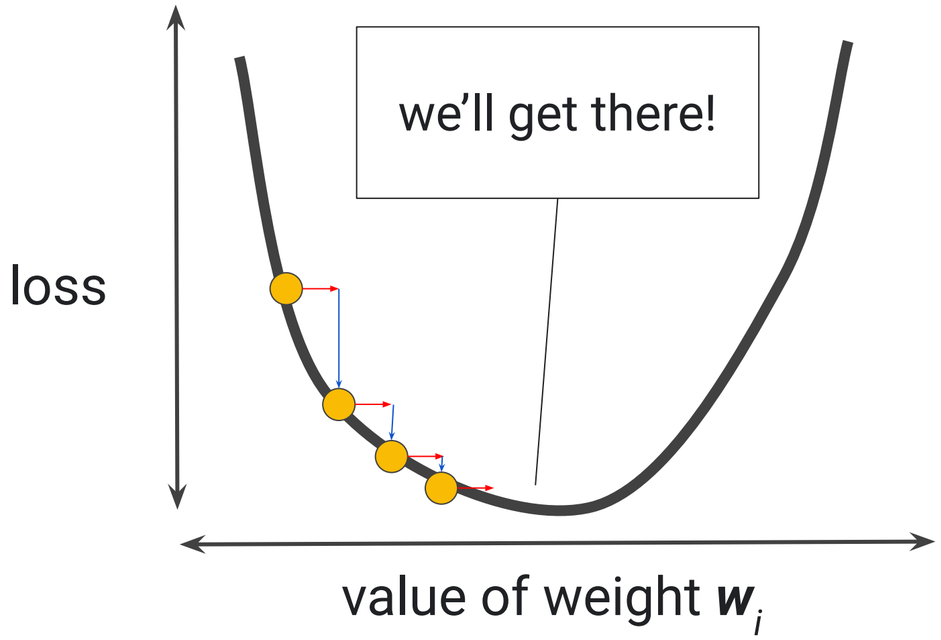


A correct and constant step size can be difficult to find

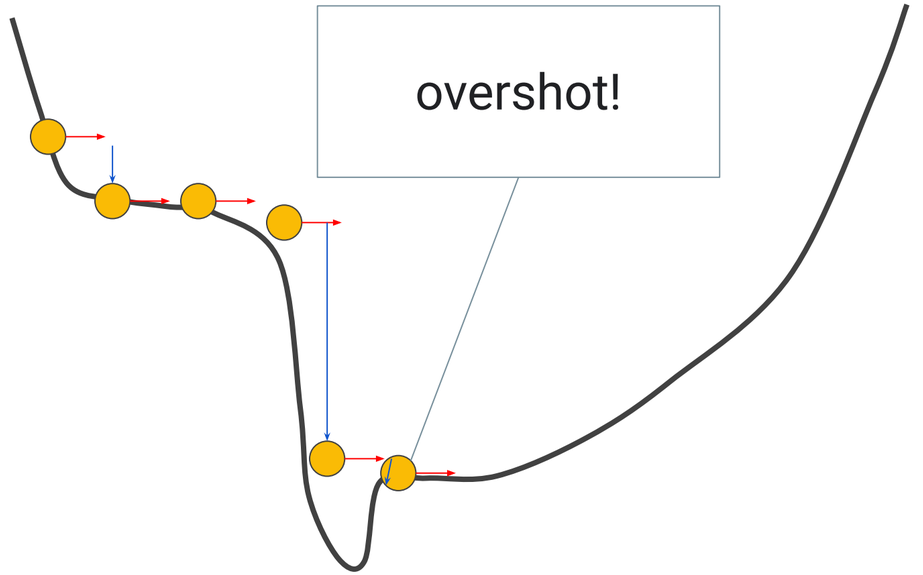


Step size or "learning rate" is a hyper-parameter which is set before training.

A correct and constant step size can be difficult to find

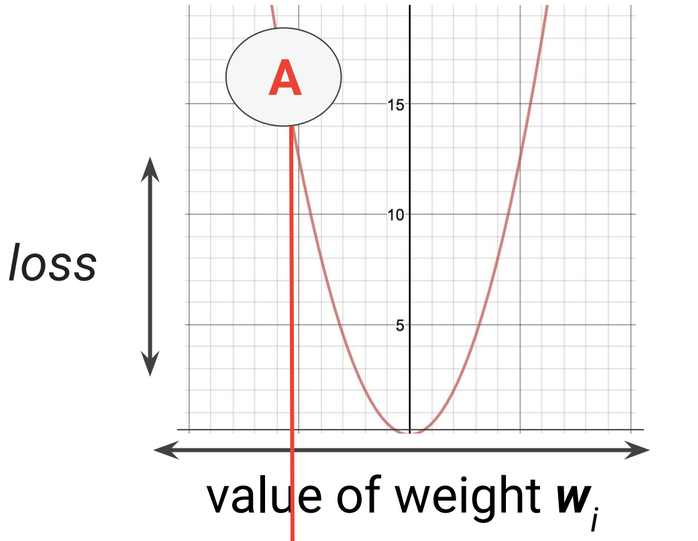


Step size or "learning rate" is a hyper-parameter which is set before training.



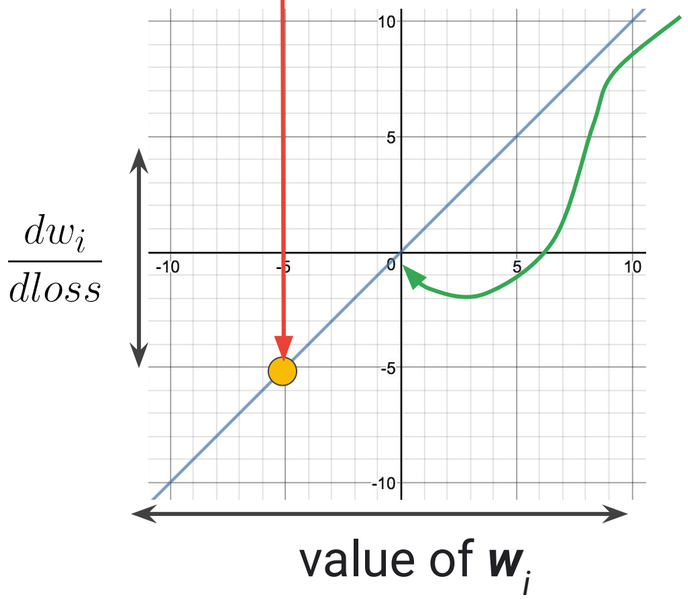
One size does not fit all models.

The Loss Function slope provides direction and step size in your search



Loss Function
(e.g. RMSE)

Remember, your goal is to find the minimum loss which is where the Loss Function slope is 0.

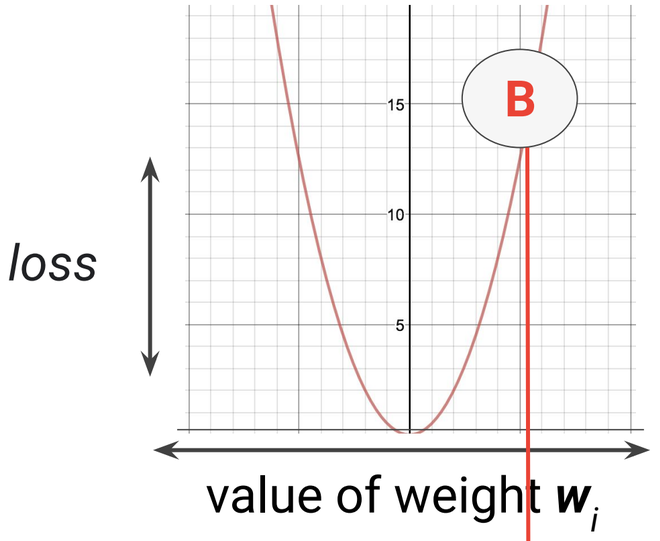


Loss Function Slope (Derivative)

Slope is Negative
Direction: Go Right!

Magnitude is (-5)
Step Size: Big

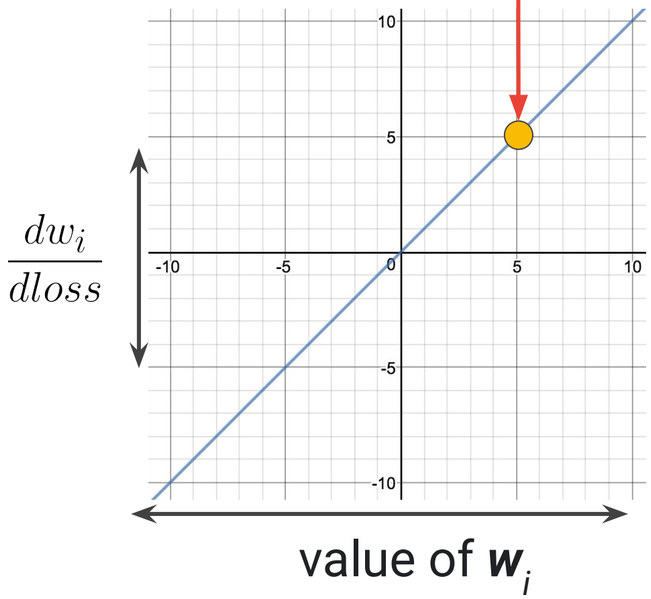
The Loss Function slope provides direction and step size in your search



Loss Function
(e.g. RMSE)

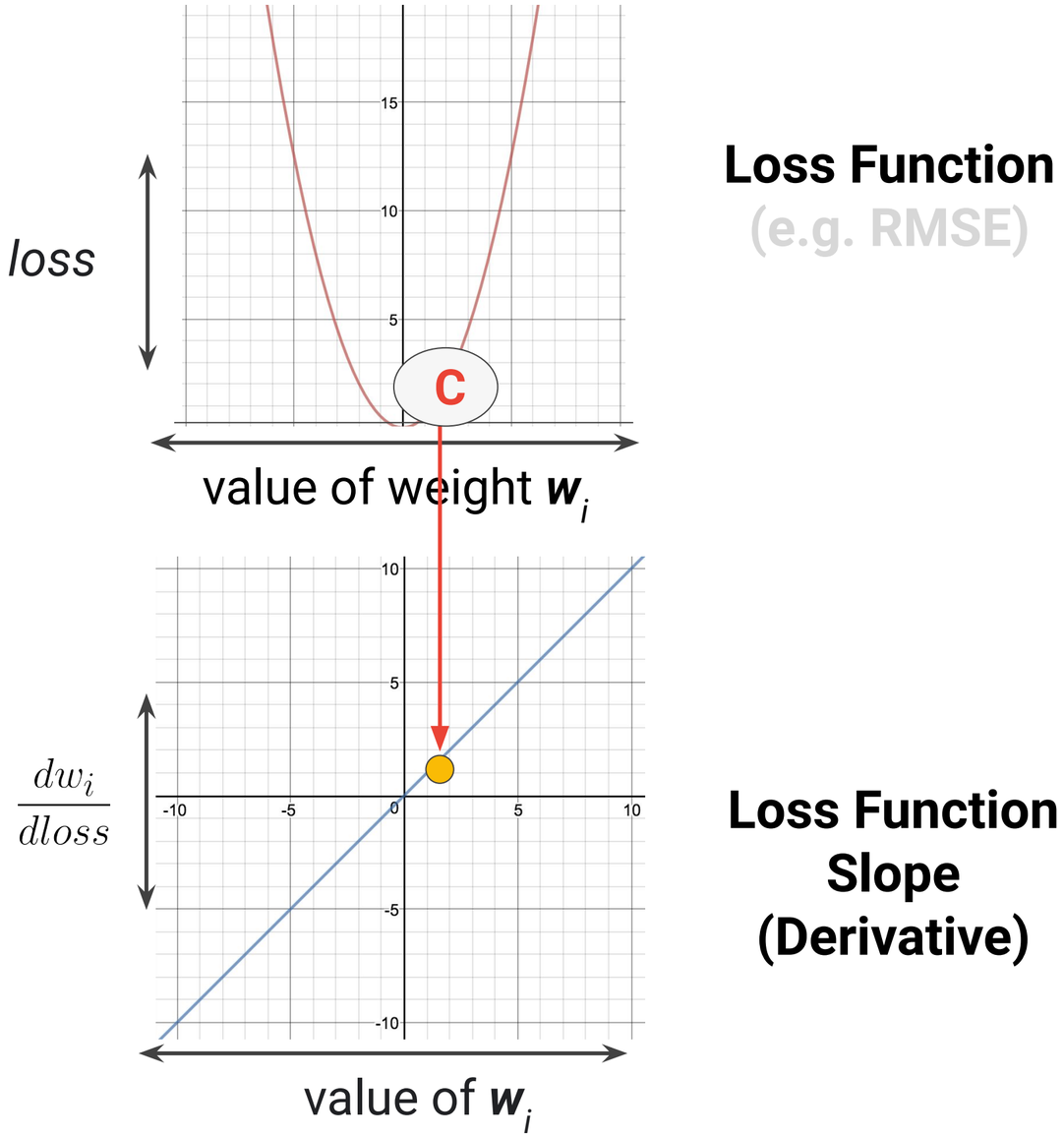
Slope is Positive
Direction: Go Left!

Magnitude is 5
Step Size: Big



Loss Function Slope (Derivative)

The Loss Function slope provides direction and step size in your search



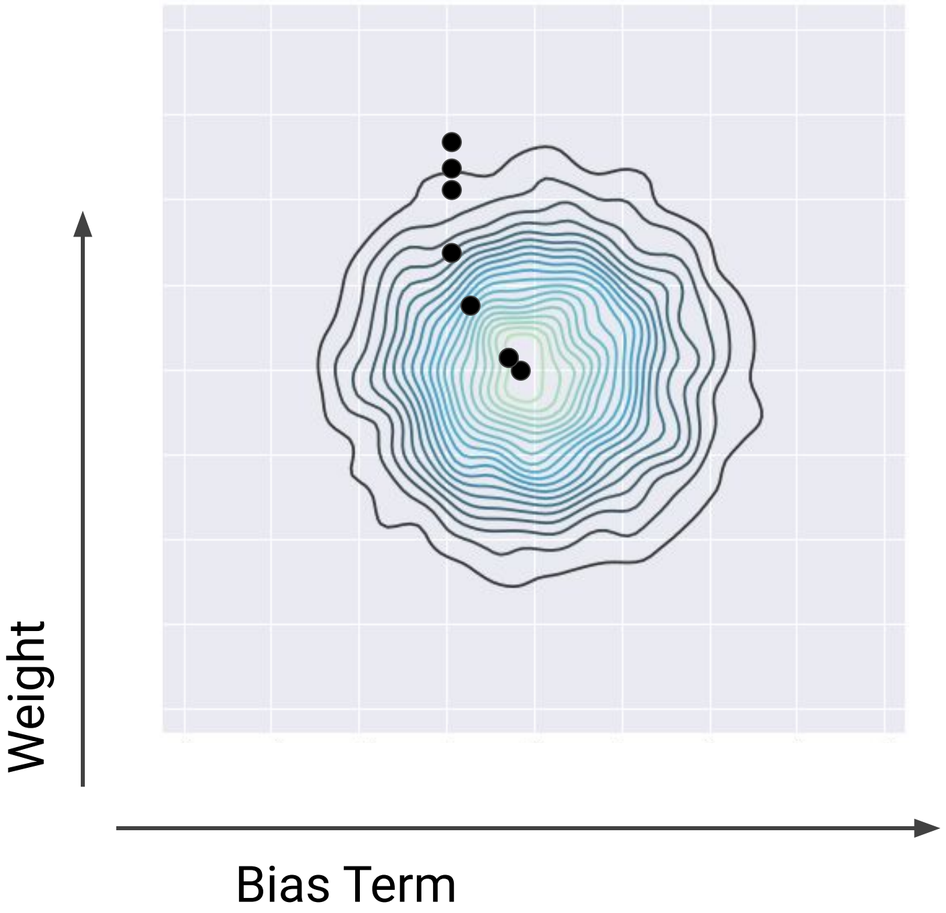
Loss Function
(e.g. RMSE)

Loss Function Slope (Derivative)

Slope is Positive
Direction: Go Left!

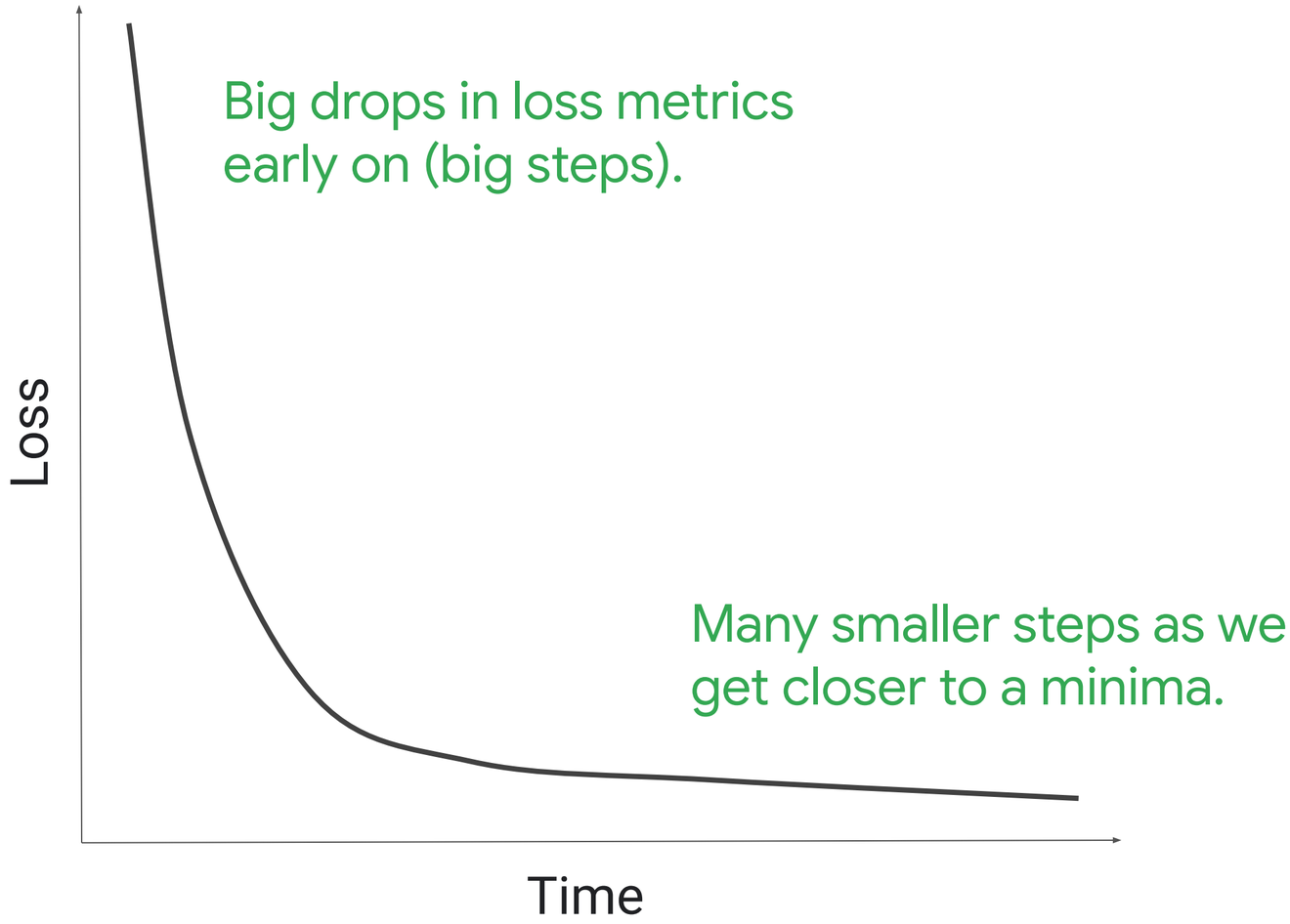
Magnitude is 2
Step Size

Are you done yet?

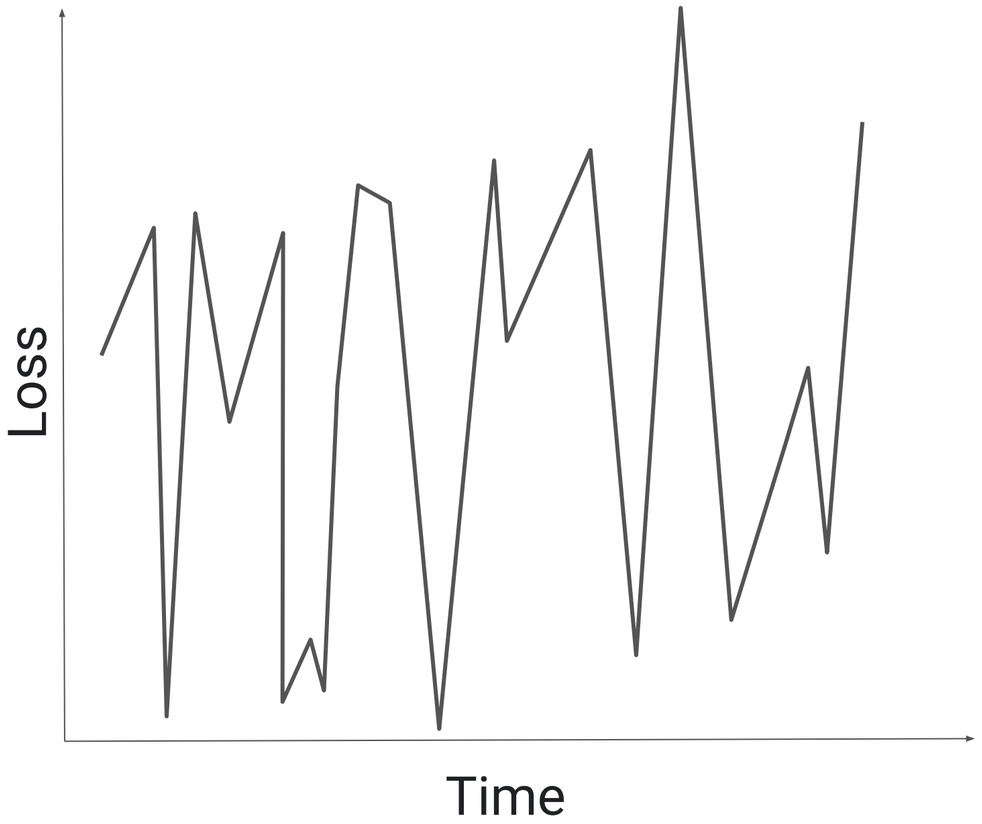


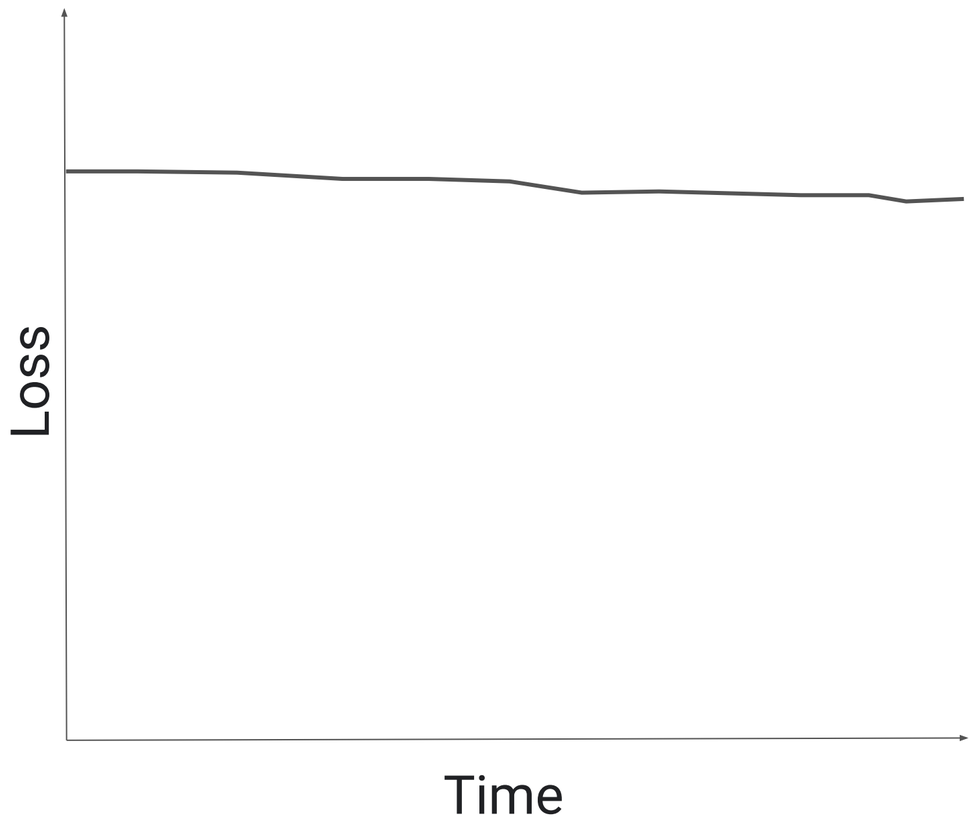

```
while loss is > Epsilon:
   derivative = computeDerivative()
   for i in range(self.params):
      self.params[i] = //
                  self.params[i] //
                        - derivative[i]
   loss = computeLoss()
```

A typical loss curve



Troubleshooting a Loss Curve



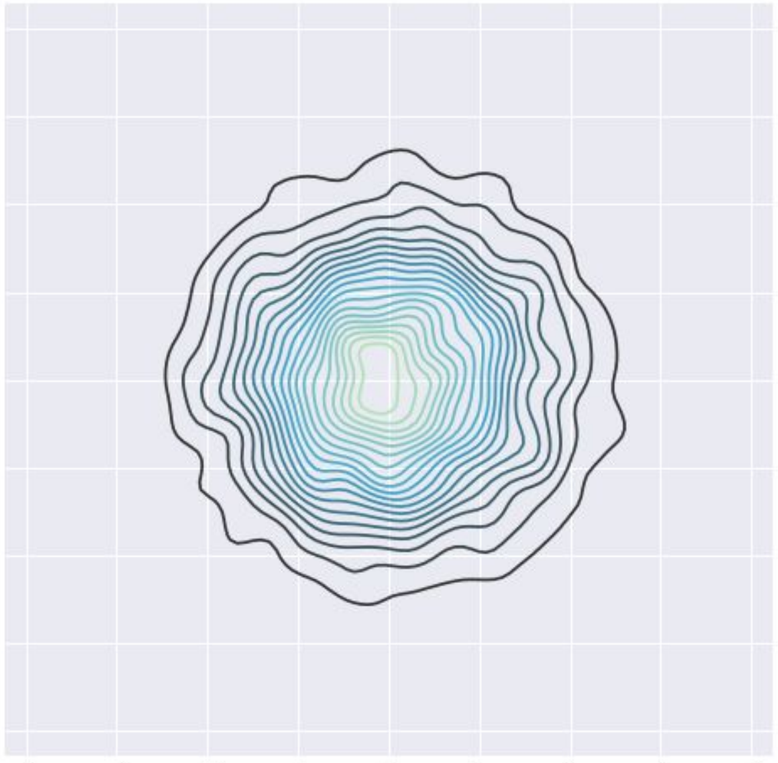


Adding a scaling hyperparameter

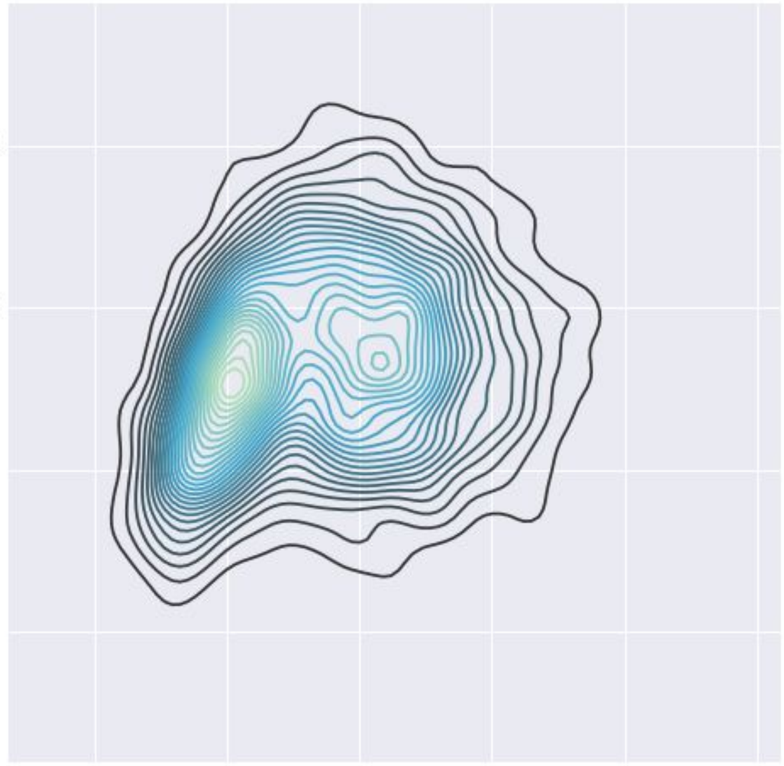
```
while loss is > Epsilon:
    derivative = computeDerivative()
  for i in range(self.params):
    self.params[i] = //
              self.params[i] //
                  - learning_rate //
                  * derivative[i]
  loss = computeLoss()
```

Problem: My model changes every time I retrain it

Loss Surface with a global minimum

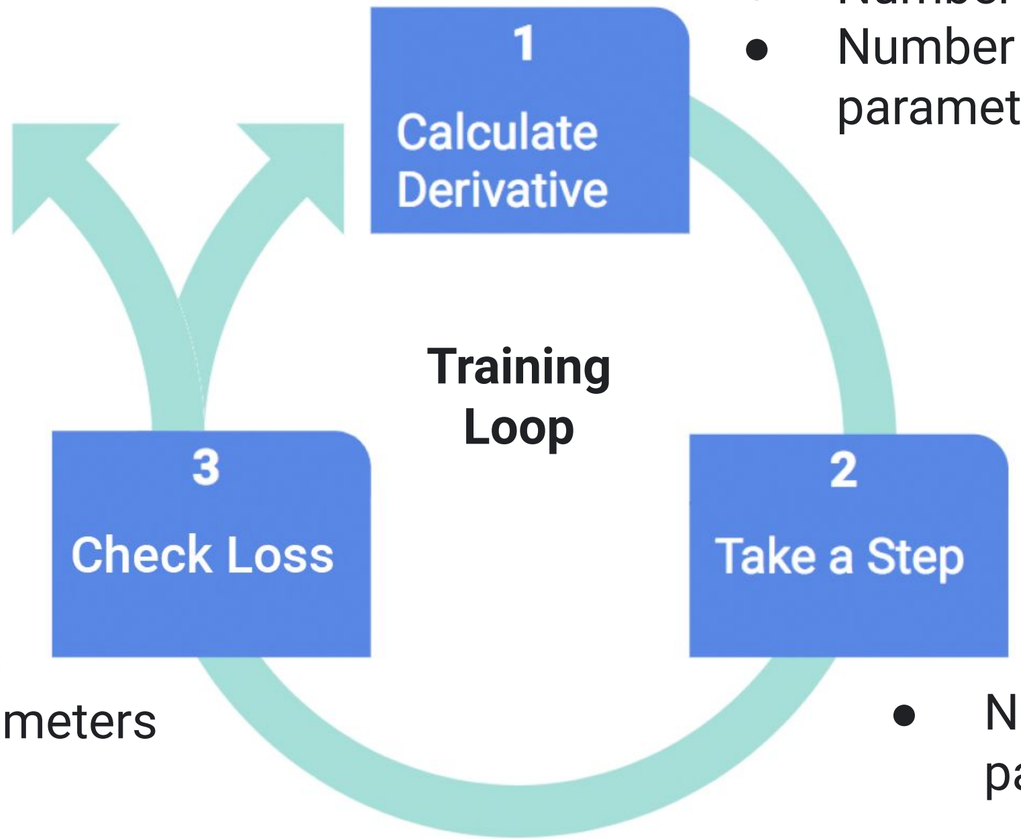


Loss Surface with more than one minima



Problem: Model training is still too slow

Done

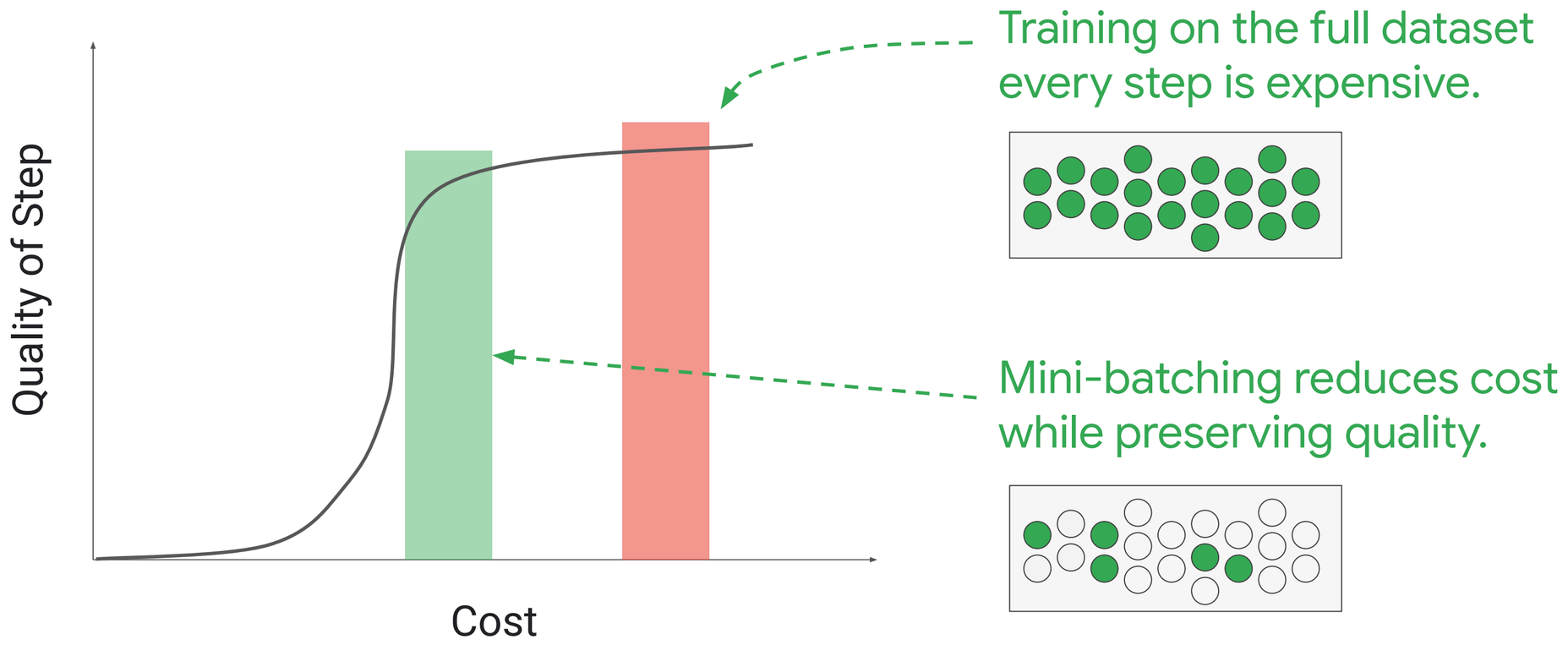


- Number of data points
- Number of model parameters

- Number of model parameters

- Number of data points
- Number of model parameters
- Frequency

Calculating the derivative on fewer data points



Typical values for batch size: 10 - 1000 examples.

Checking loss with reduced frequency

```
while loss is > Epsilon:
    derivative = computeDerivative()
  for i in range(self.params):
    self.params[i] = //
              self.params[i] //
                  - learning_rate //
                  * derivative[i]
  if readyToUpdateLoss():
        loss = updateLoss()
```

Popular implementations for readyToUpdateLoss():

- Time-based (e.g., every hour)
- Step-based (e.g., every 1000 steps)

Agenda

Defining ML Models

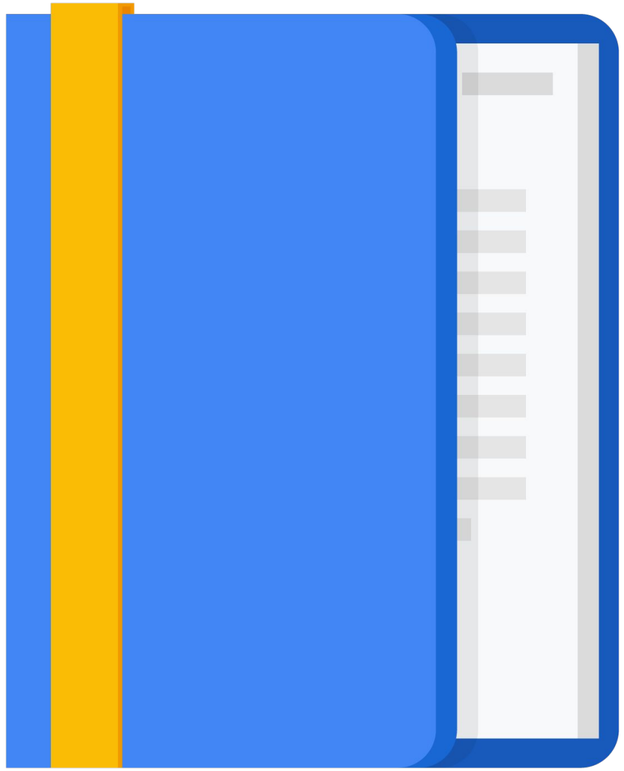
Introducing Loss Functions

Gradient Descent

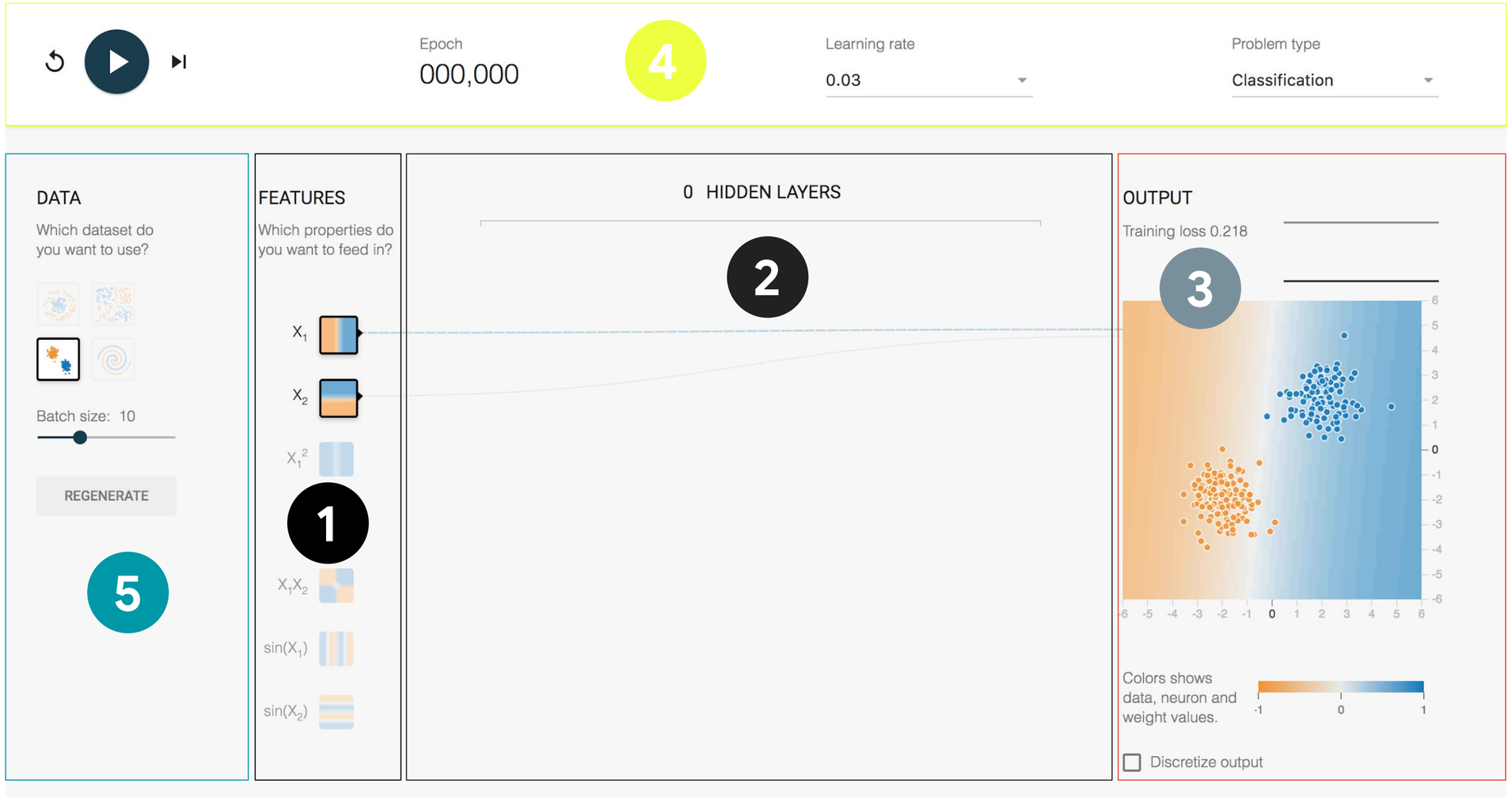
TensorFlow Playground

Lab: Develop an Intuitive Understanding of Neural Networks Using Tensorflow Playground

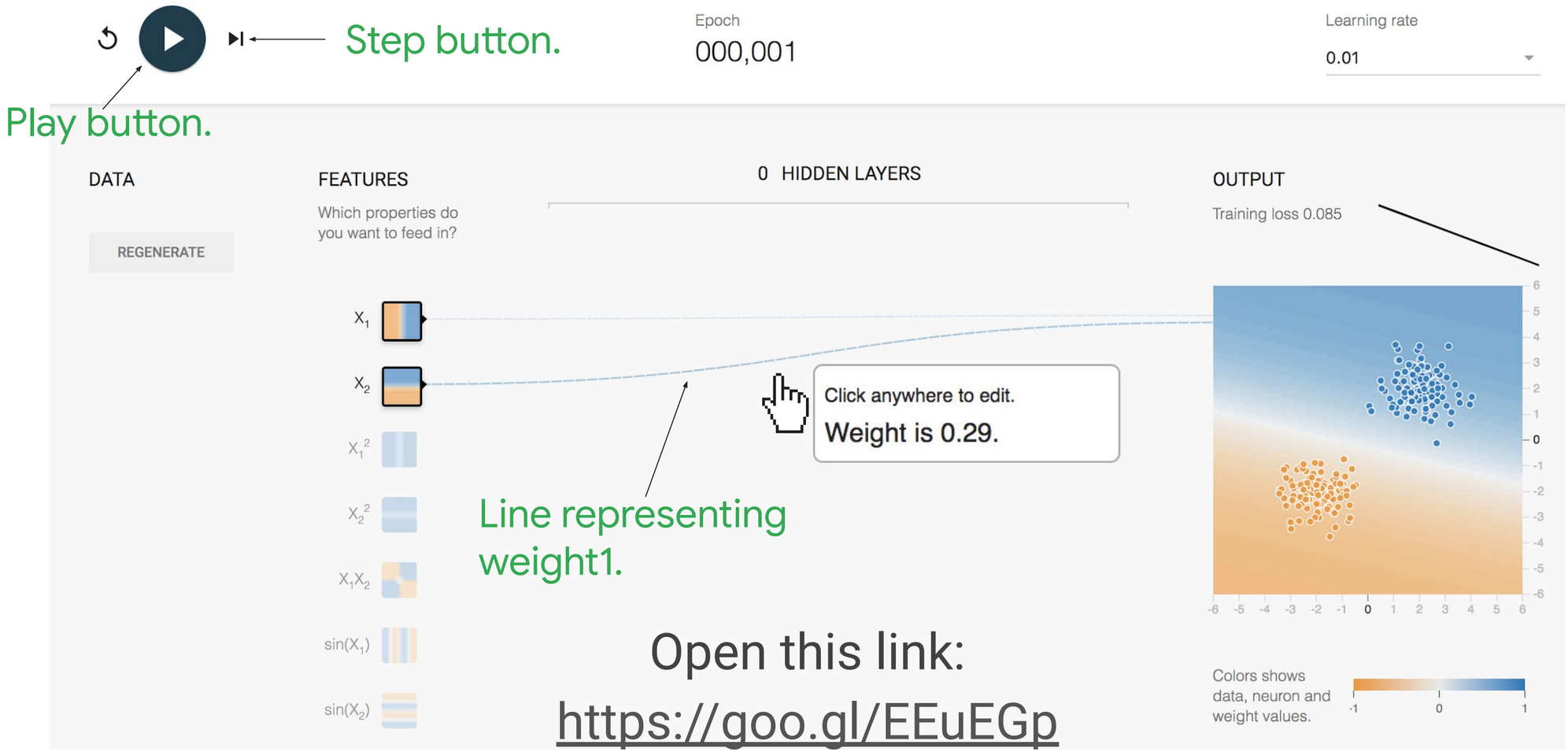
Performance Metrics



TensorFlow Playground Interface



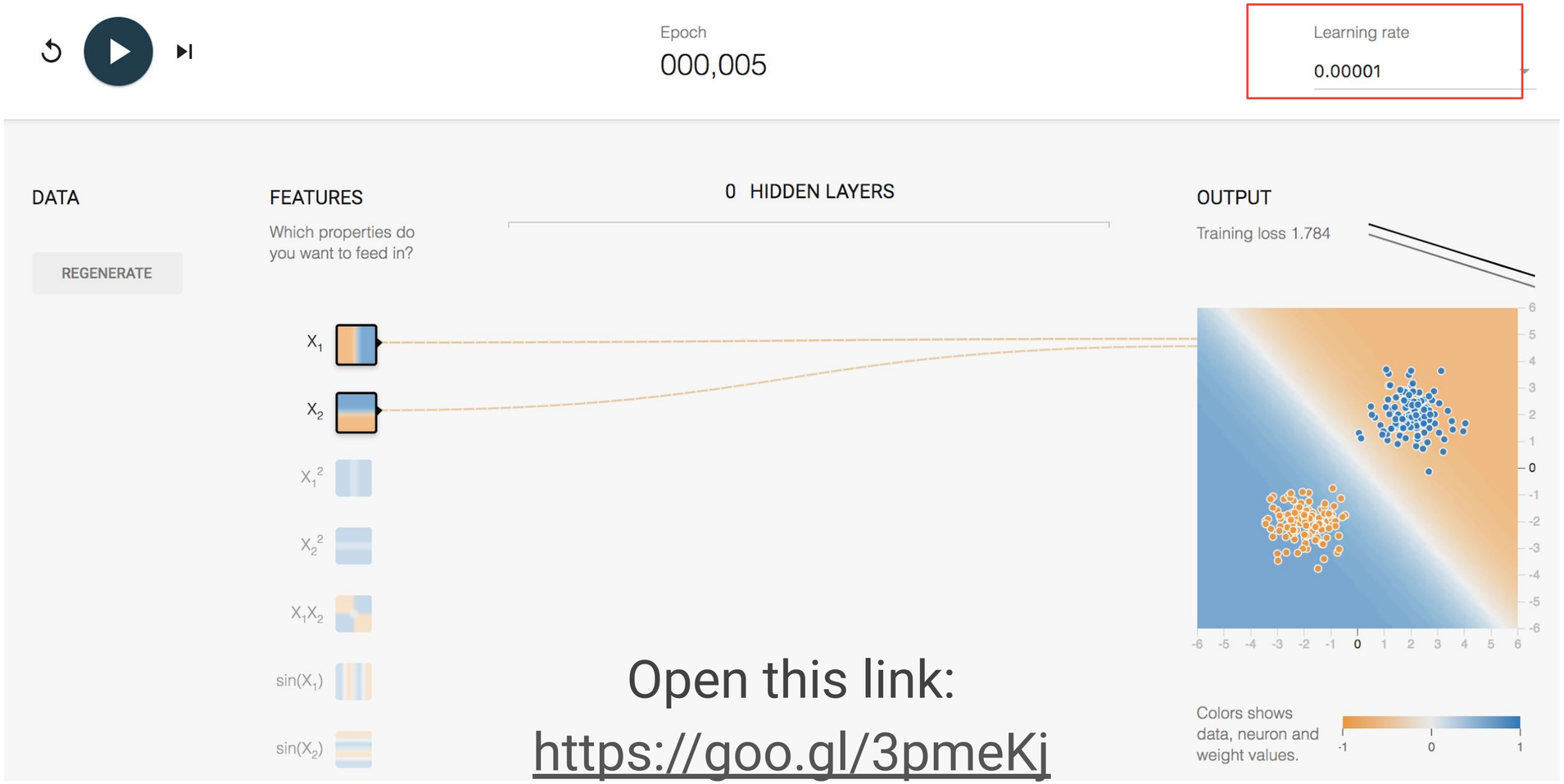
Train your first model



Open this link:

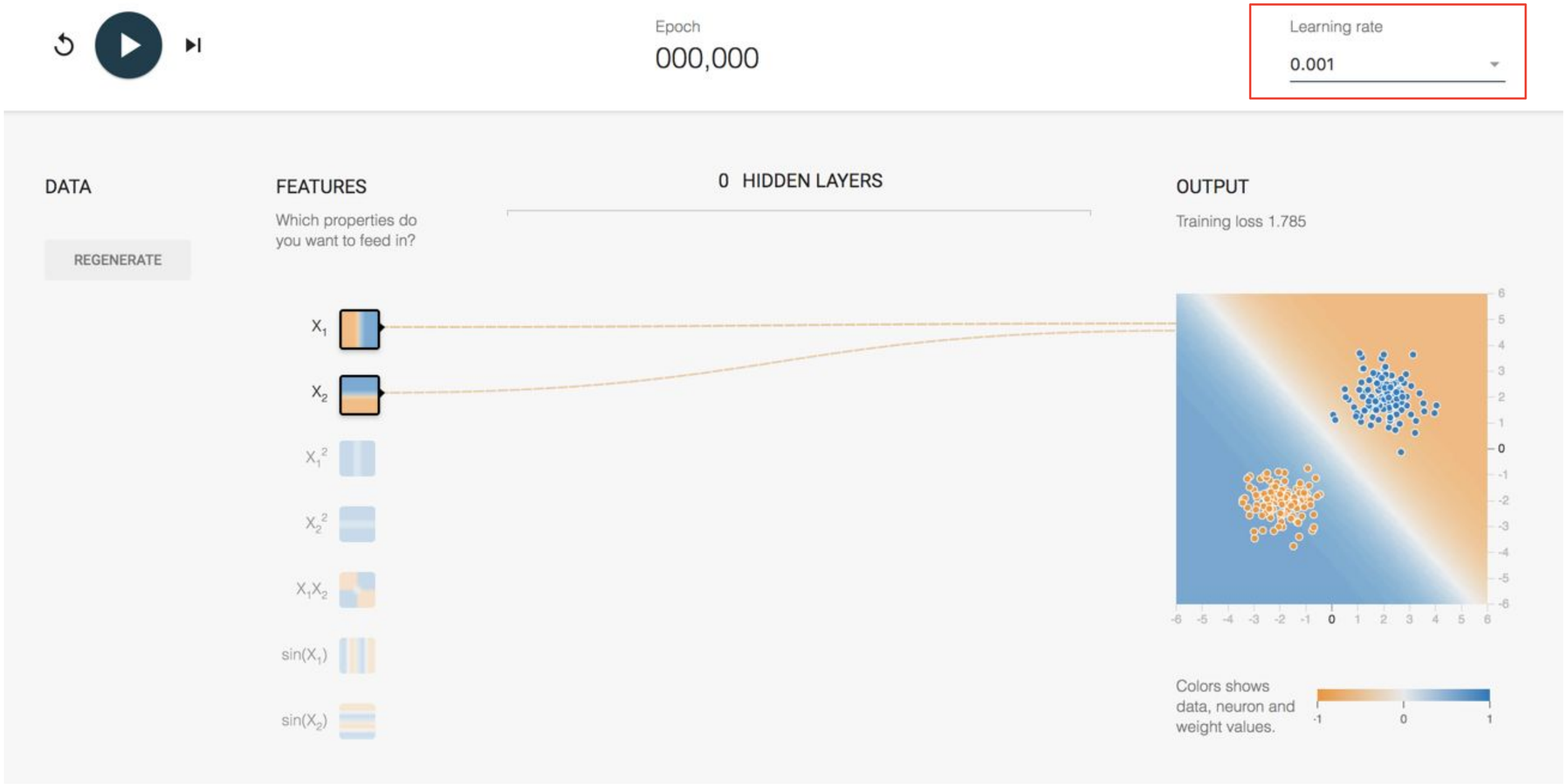
https://goo.gl/EEuEGp

Experiment with Learning Rate: 0.00001 (tiny)

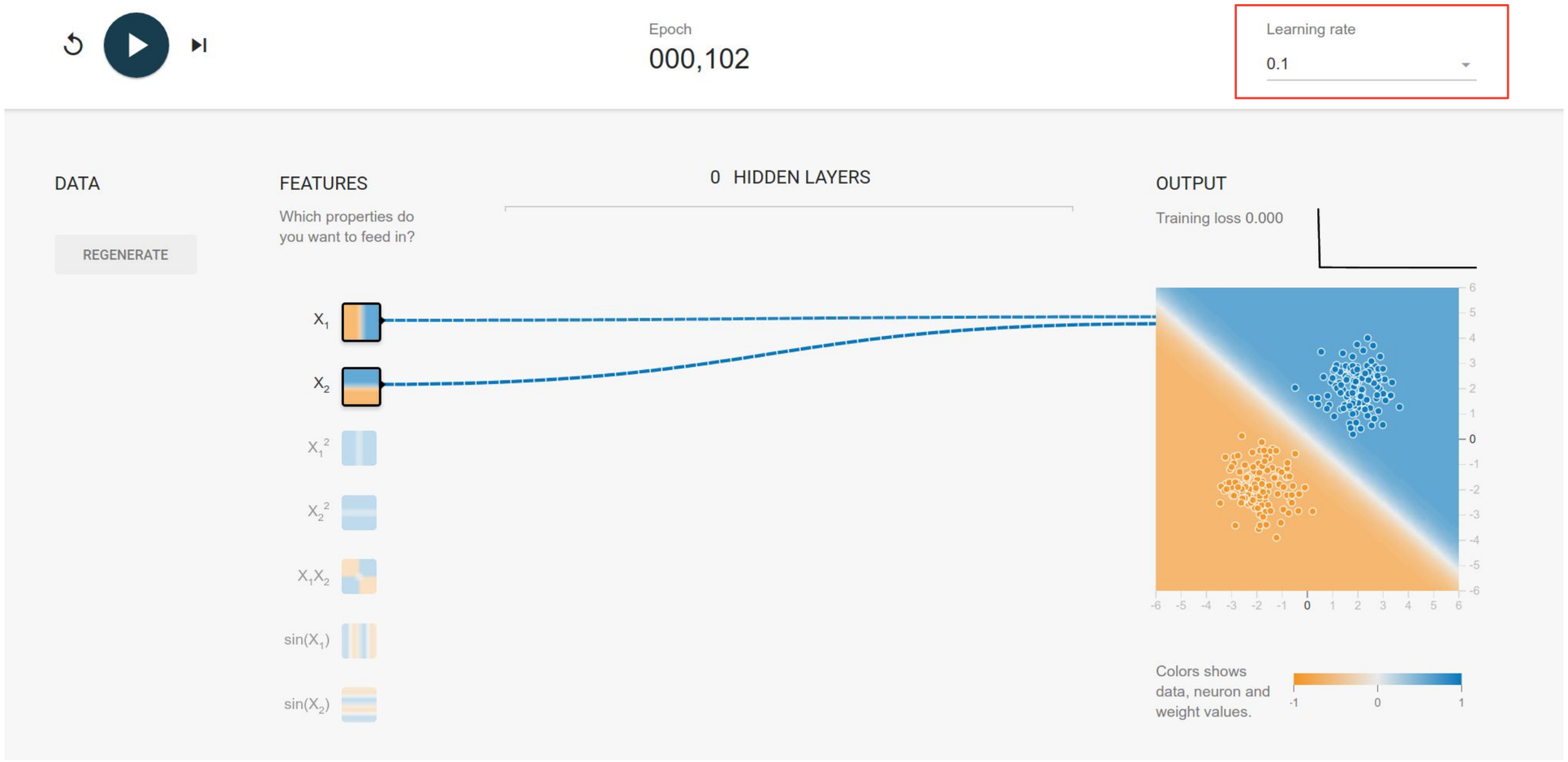


Open this link:
https://goo.gl/3pmeKj

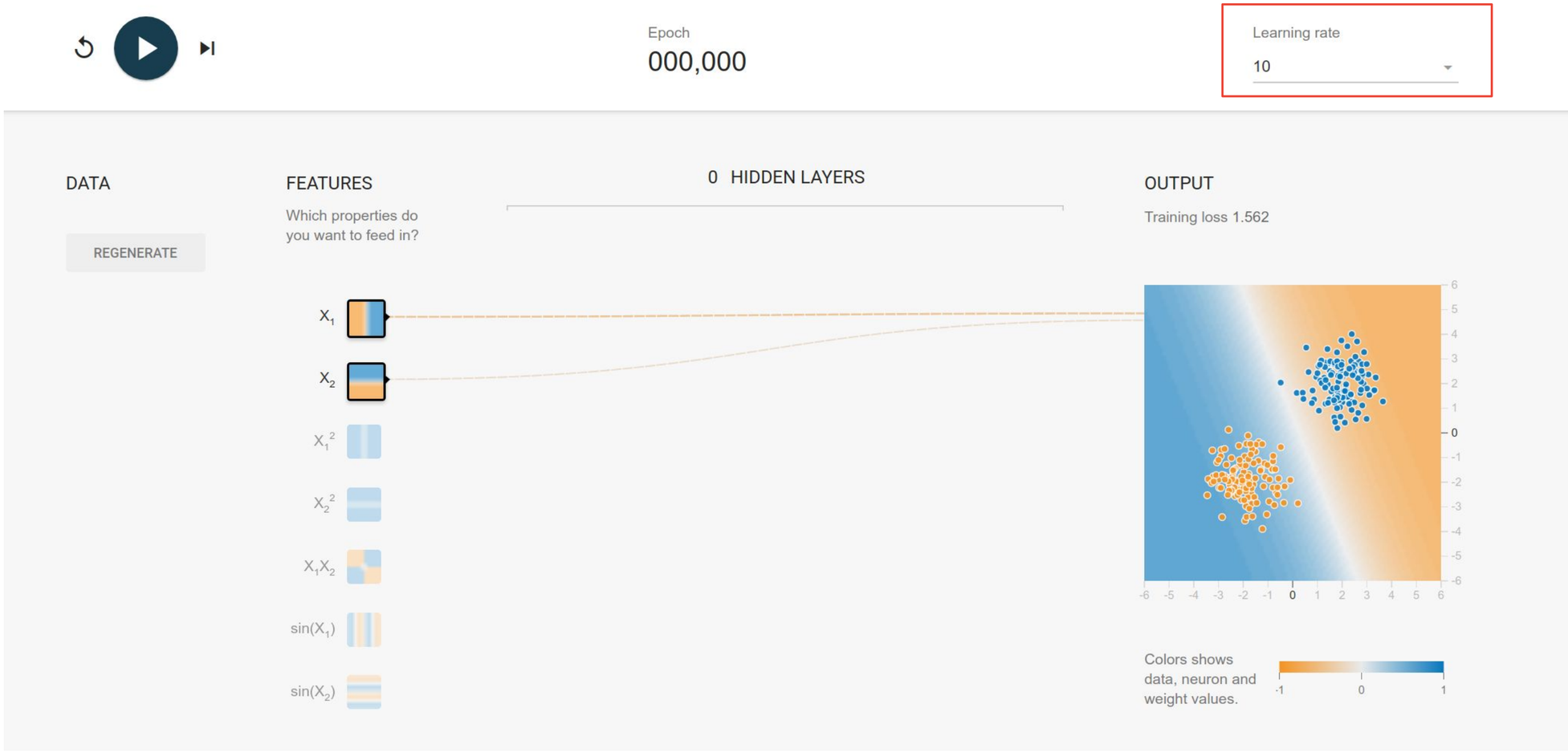
Experiment with Learning Rate: 0.001 (small)



Experiment with Learning Rate: 0.1 (moderate)



Experiment with Learning Rate: 10 (huge)

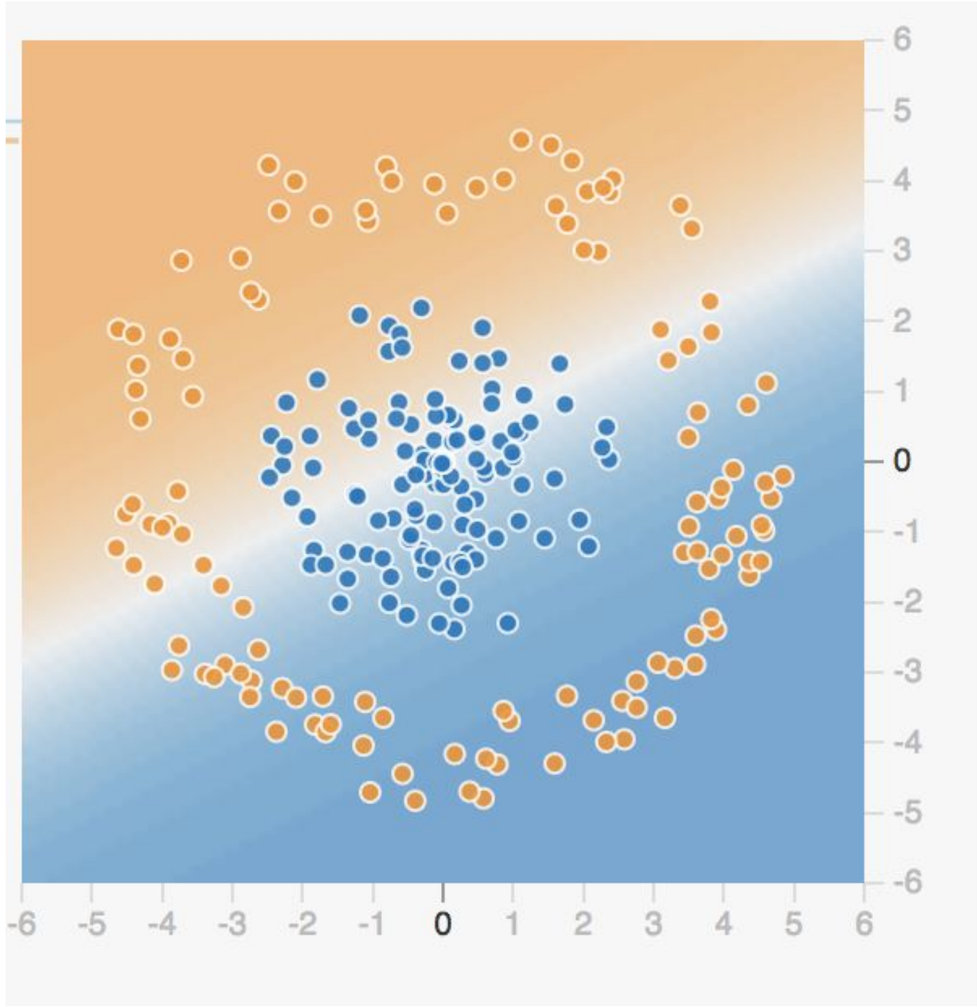


Experimenting with learning rate: Observations

Learning Rate	Weight1	Loss Over Time
.00001	.42	
.001	.65	
.1	1.0	
10	12.0	

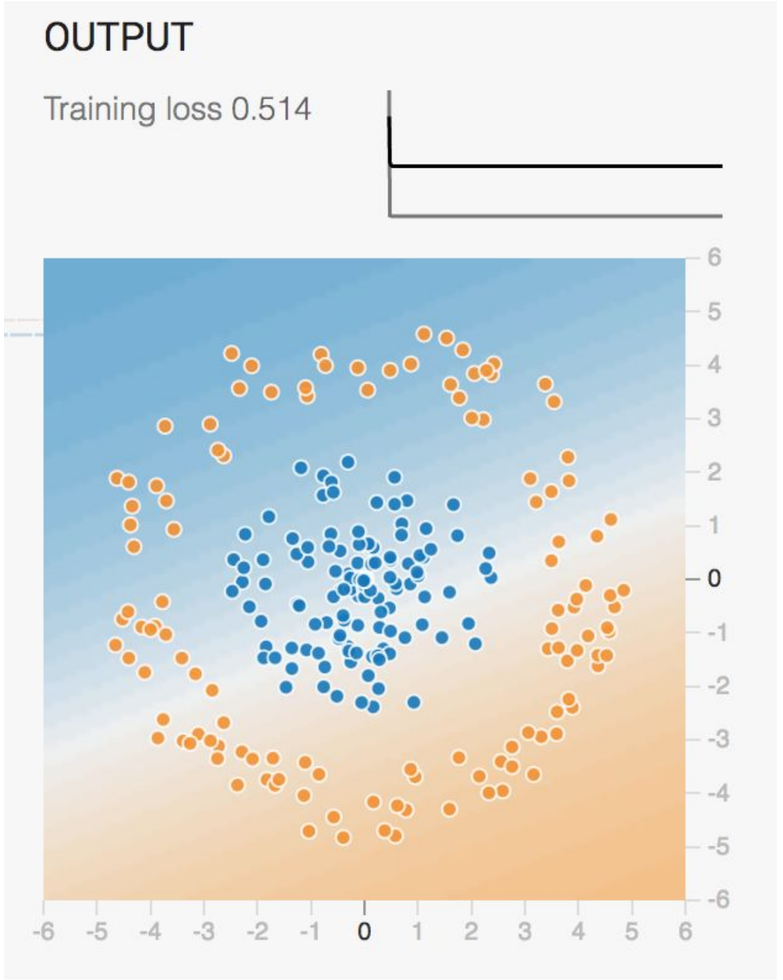
What about datasets that aren’t easy to separate?

Open this link:
https://goo.gl/ou9iMB



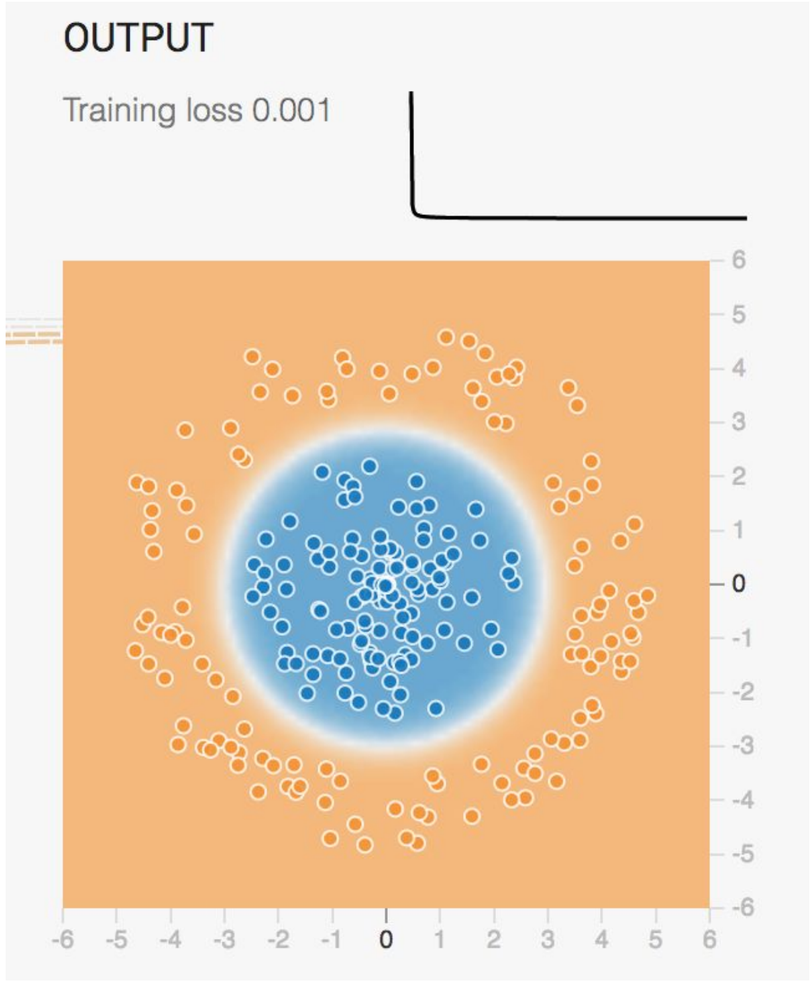
The limitations of linear models

The decision boundary does a poor job of dividing the data by class.



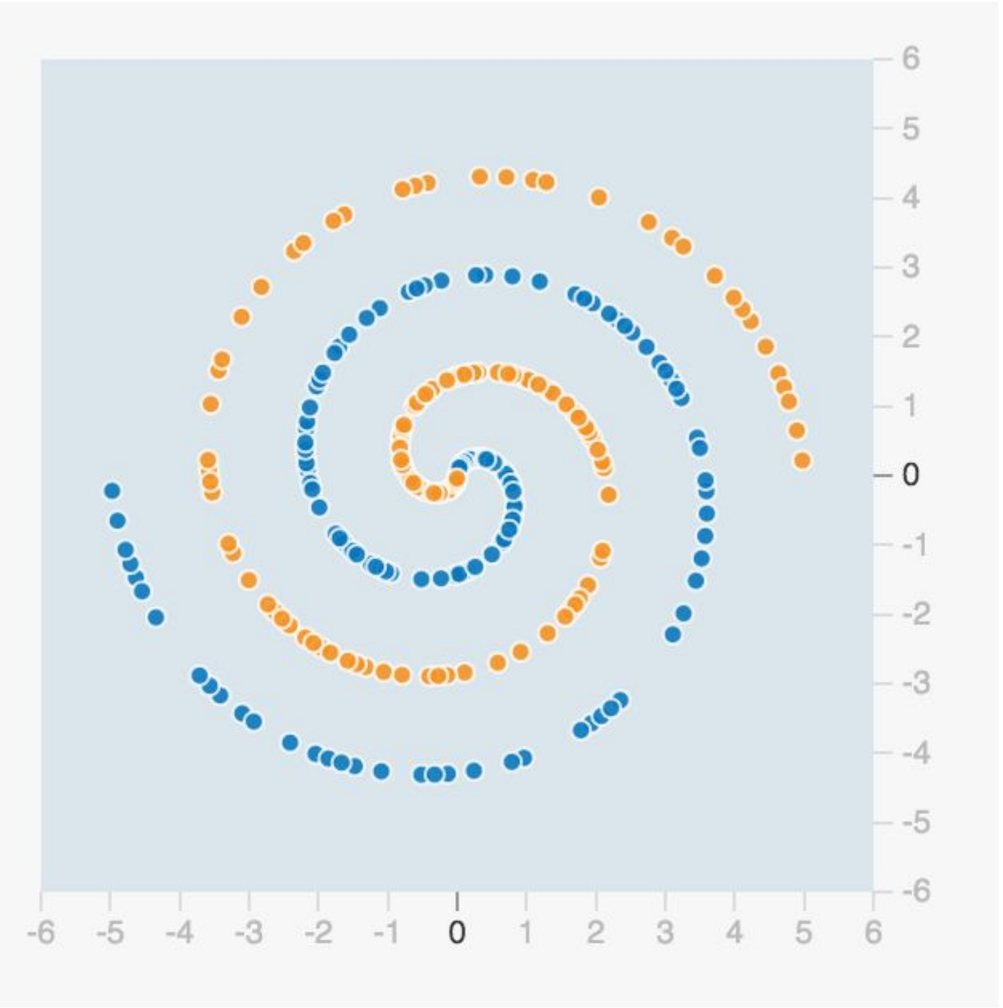
Introducing non-linear features

Output after selecting the X1 squared and X2 squared features.



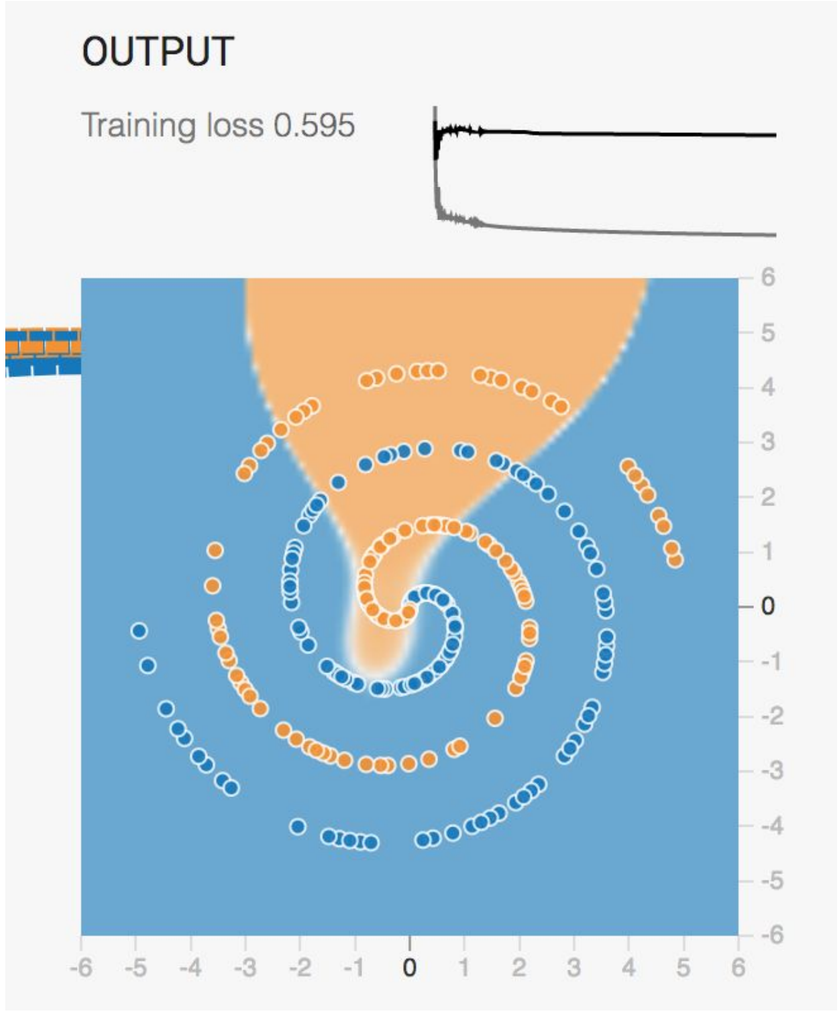
Try learning a linear model for this dataset

Open this link:
https://goo.gl/1v28Pd



Try learning a linear model for this dataset

Open this link:
https://goo.gl/1v28Pd



Agenda

Defining ML Models

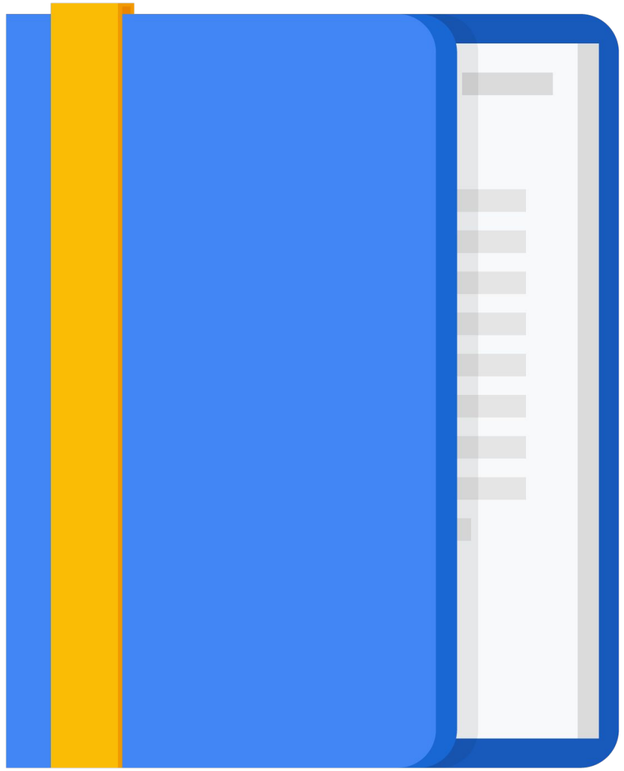
Introducing Loss Functions

Gradient Descent

TensorFlow Playground

Lab: Develop an Intuitive Understanding of Neural Networks Using Tensorflow Playground

Performance Metrics

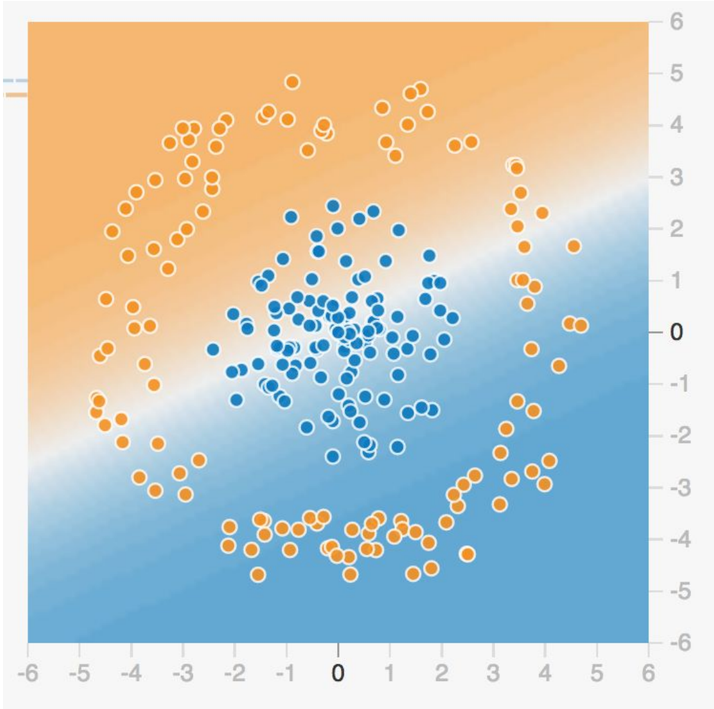


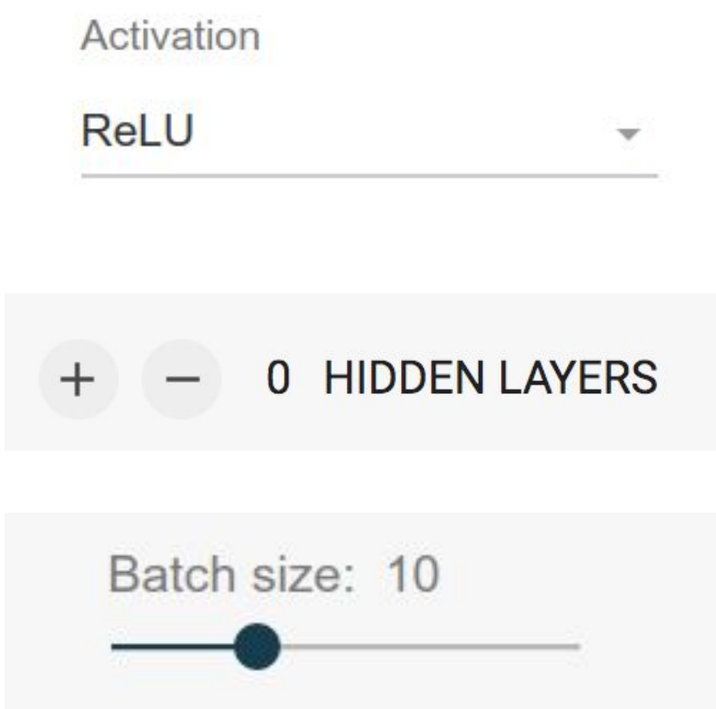
Lab Intro

Develop an Intuitive Understanding of Neural Networks Using Tensorflow Playground

Google Cloud

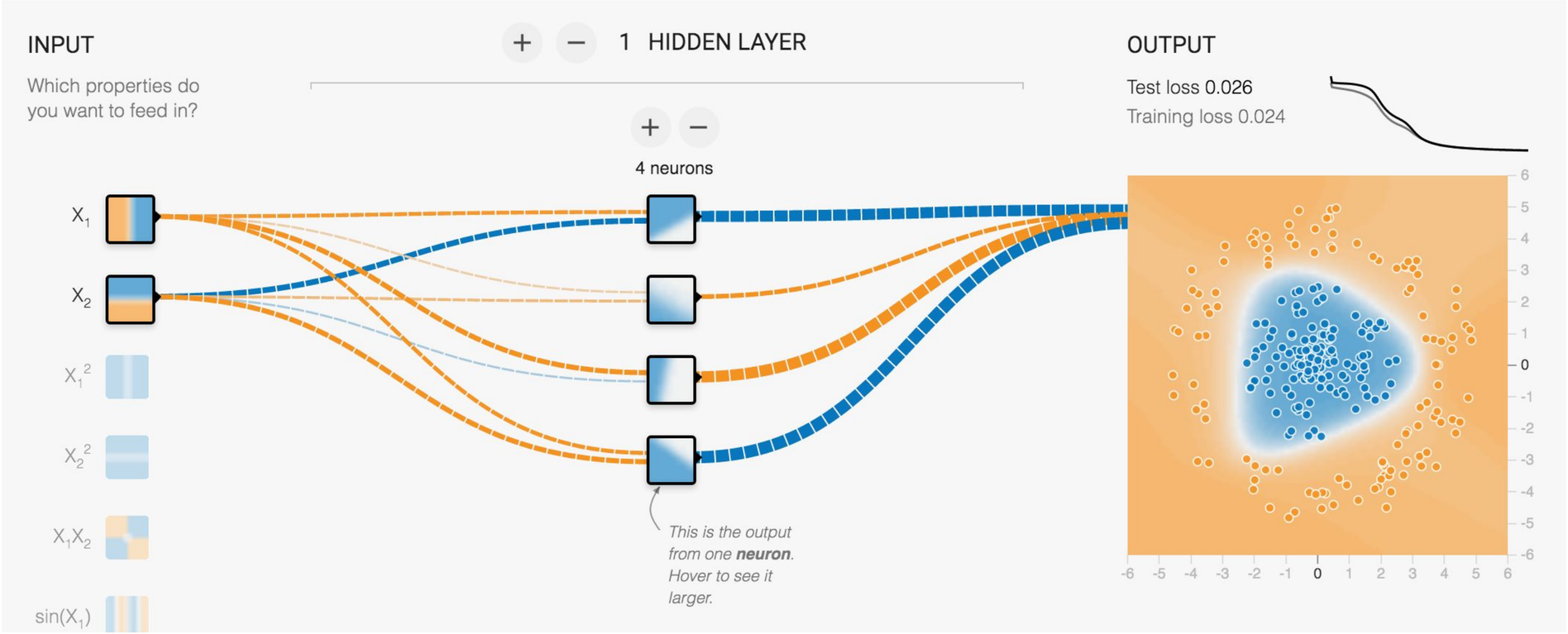
Try solving this with a neural network





https://goo.gl/VyoRWX

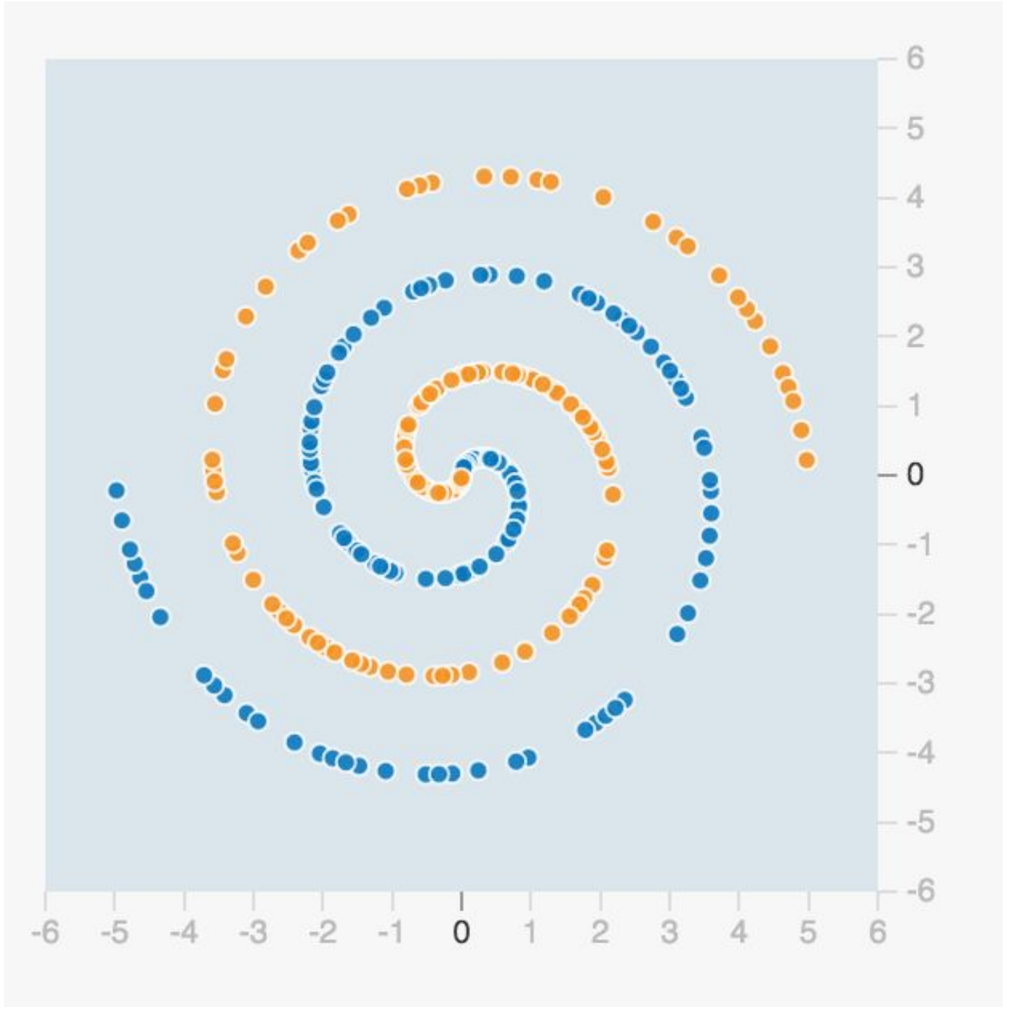
More neurons means more input combinations (features)



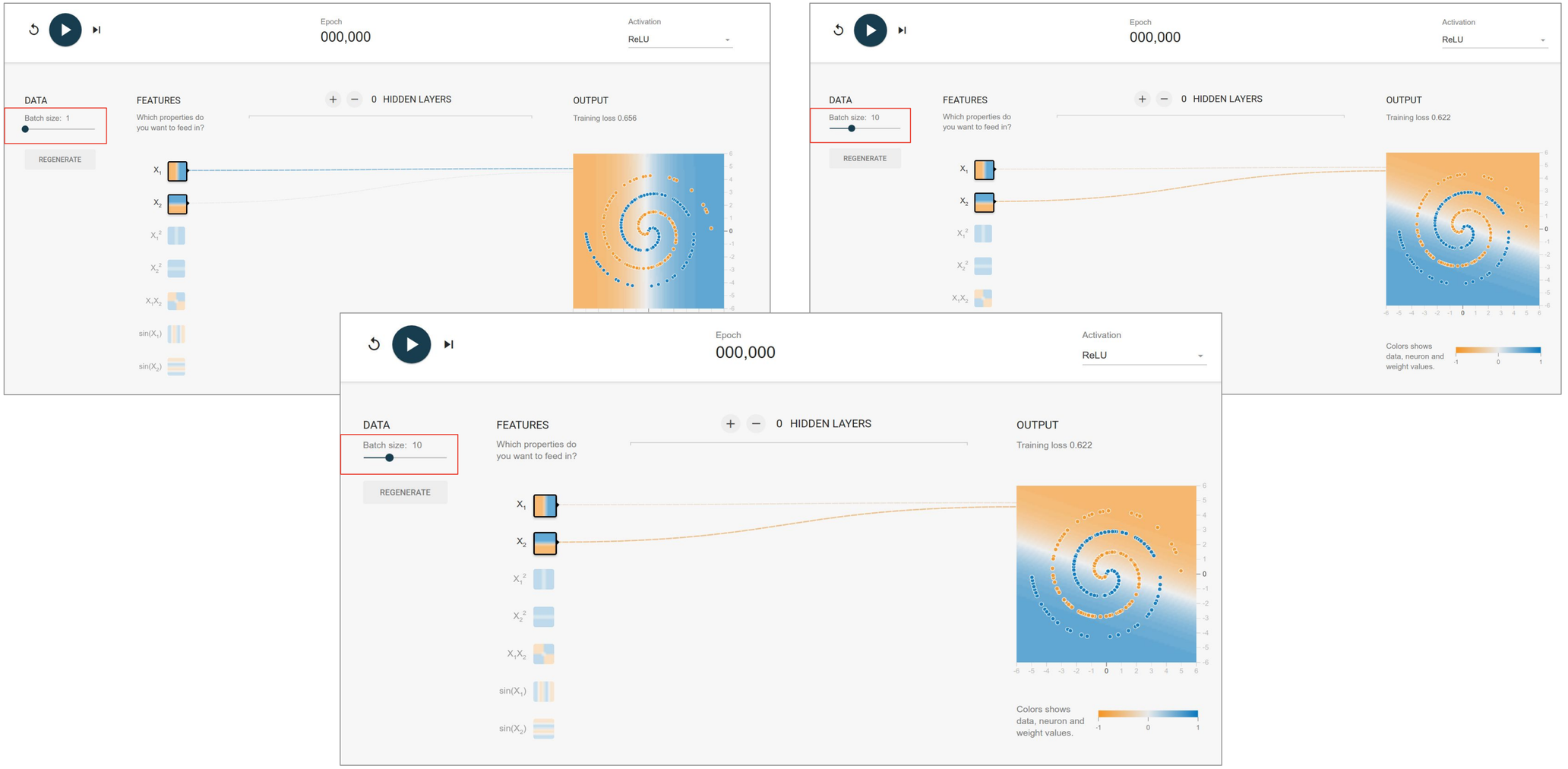
https://goo.gl/SjQf5V

Will a set of lines work?

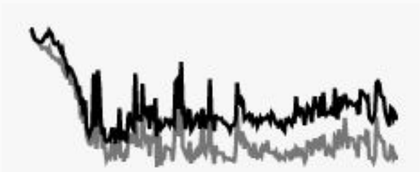
Open this link:
http://goo.gl/hrXd9T



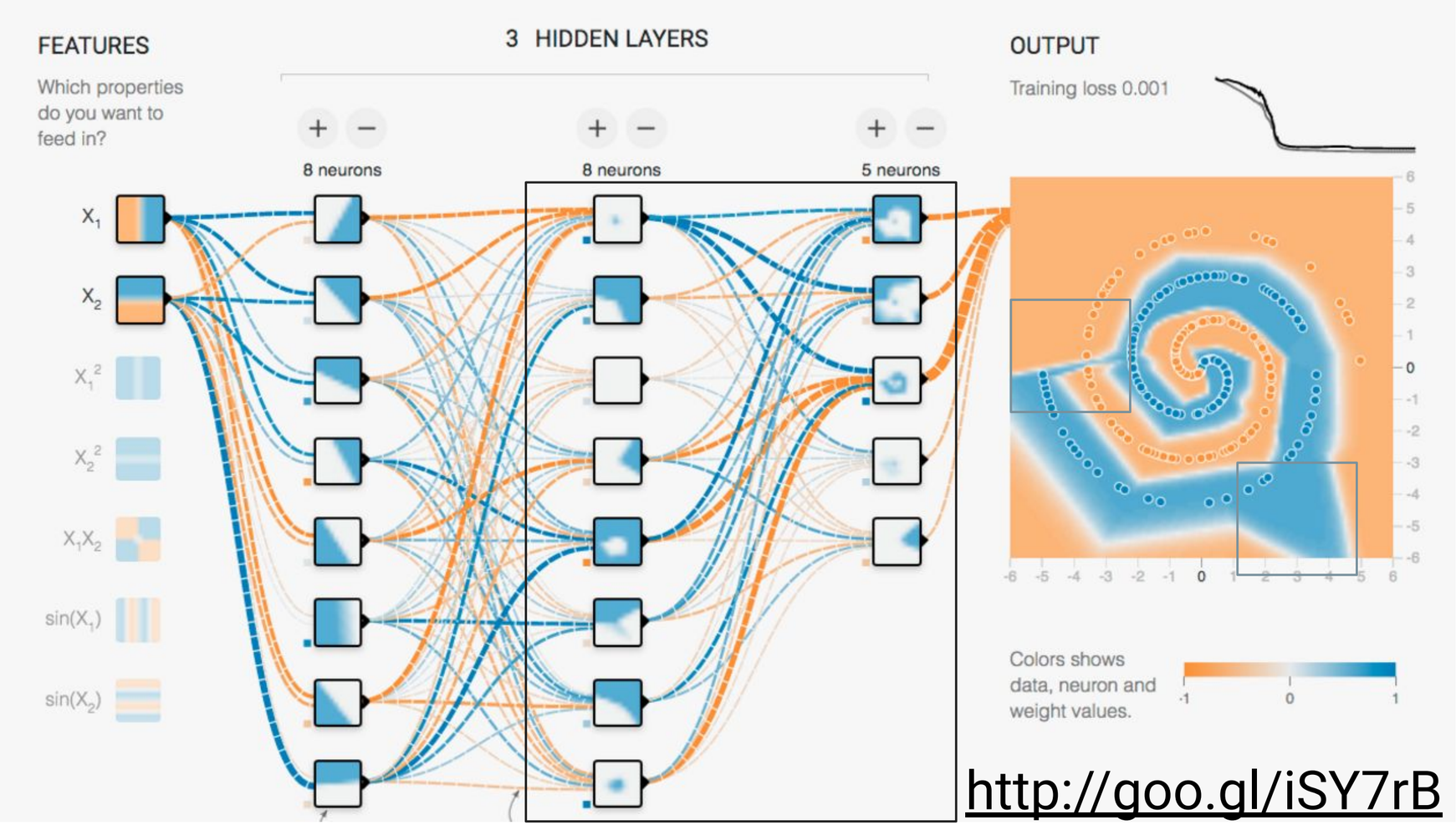
Experimenting with batch sizes 1, 10, and 30



Experimenting with batch size: Observations

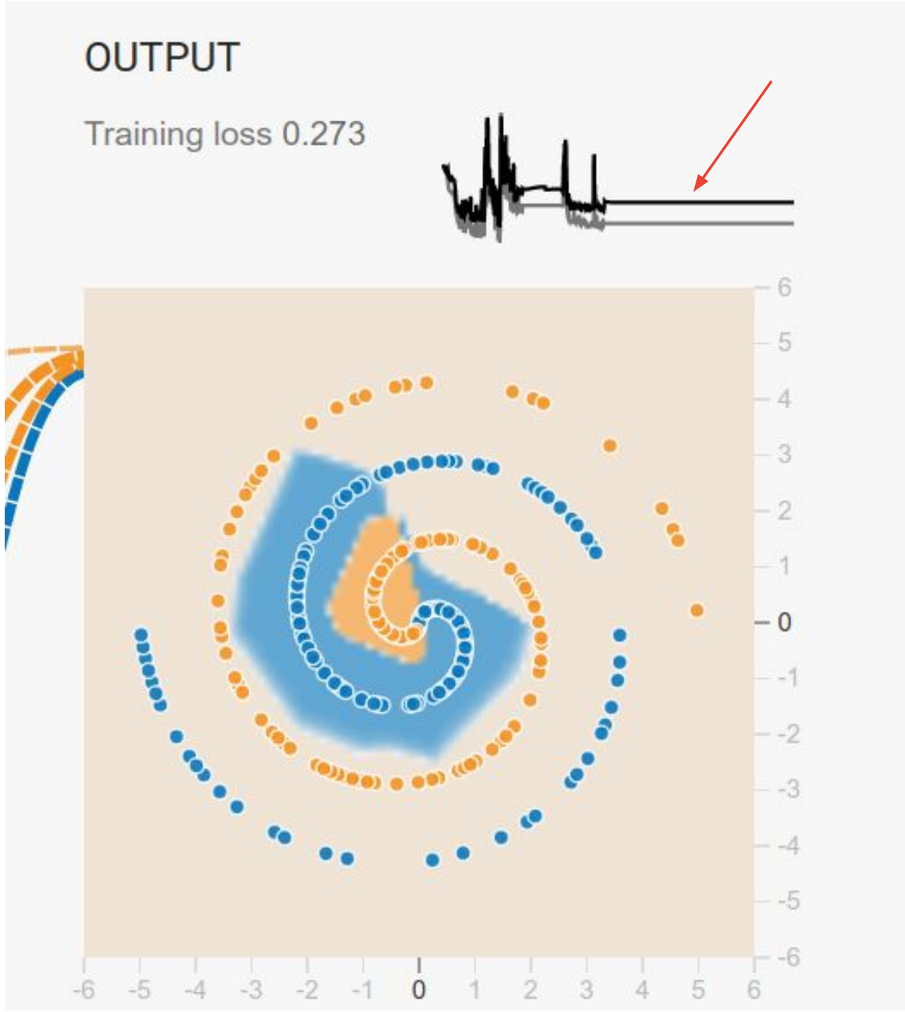
Batch Size	Loss Over Time
1	
10	
30	

More hidden layers leads to more hierarchies of features



http://goo.gl/iSY7rB

Advanced loss curve troubleshooting



Agenda

Defining ML Models

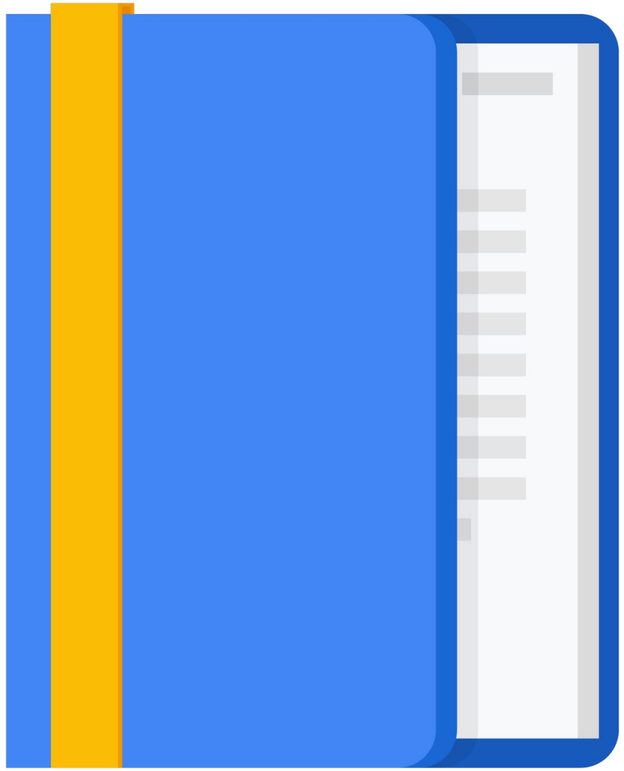
Introducing Loss Functions

Gradient Descent

TensorFlow Playground

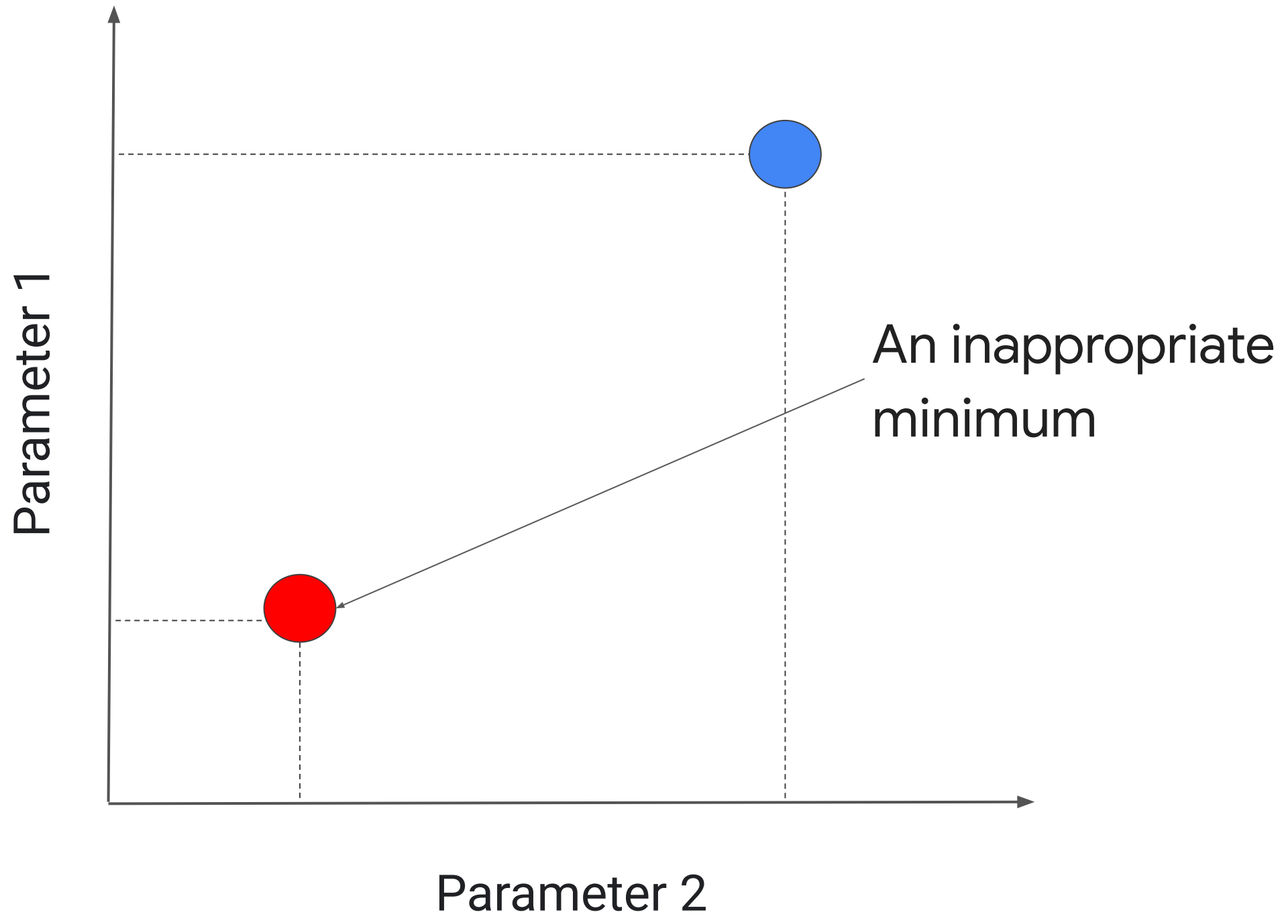
Lab: Develop an Intuitive Understanding of Neural Networks Using Tensorflow Playground

Performance Metrics

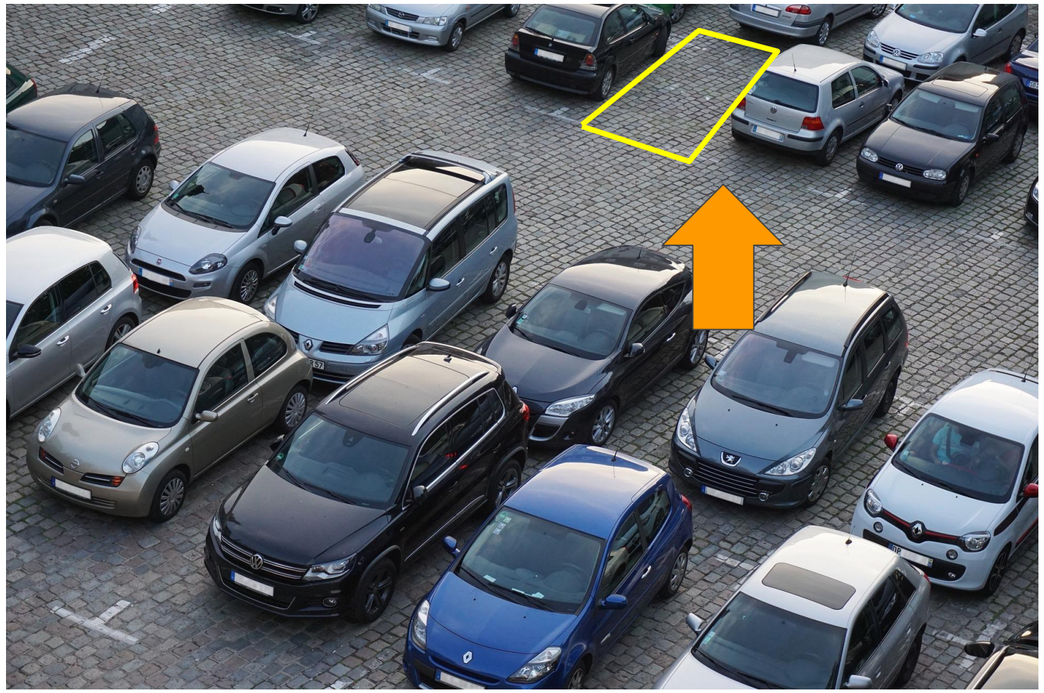


Inappropriate minima

- Doesn’t reflect the relationship between features and label.
- Won’t generalize well.



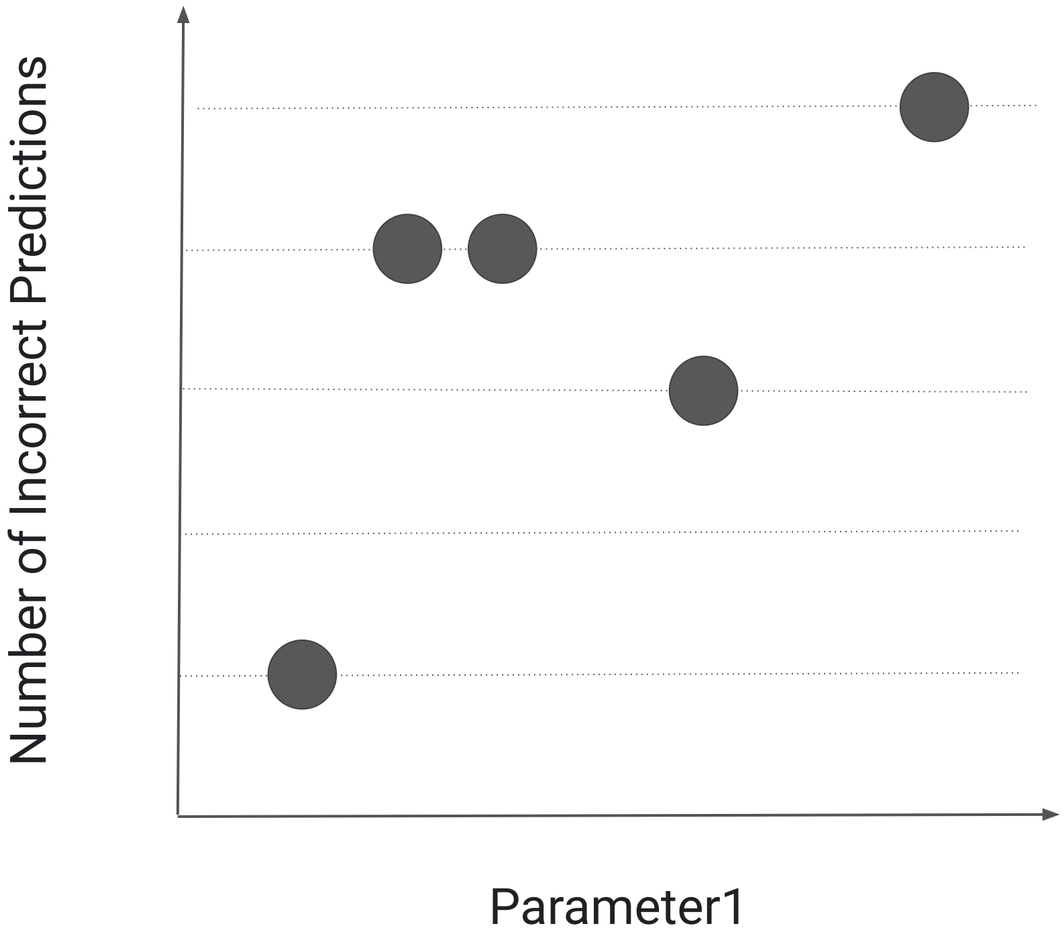
Skewed data can make inappropriate strategies seductive



1000 parking spaces.
990 of them are **taken**.
10 are **available**.

An ML model that always reported that a space was occupied would be right 99/100 times.

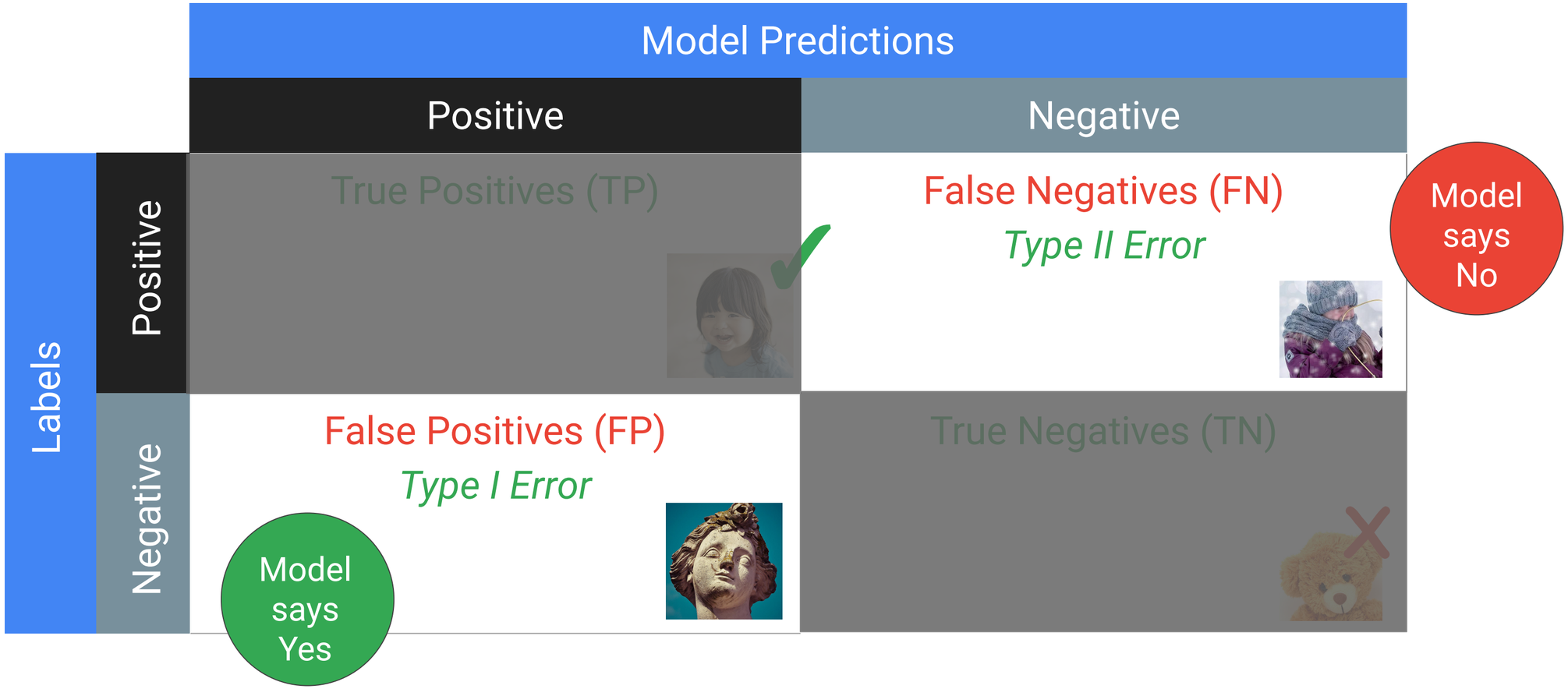
In search of a perfect loss function



Performance metrics allow you to measure what matters

Loss Functions	Performance Metrics
During training.	After training.
Harder to understand.	Easier to understand.
Indirectly connected to business goals.	Directly connected to business goals.

Use a confusion matrix to assess classification model performance



True and false positives when predicting parking spots

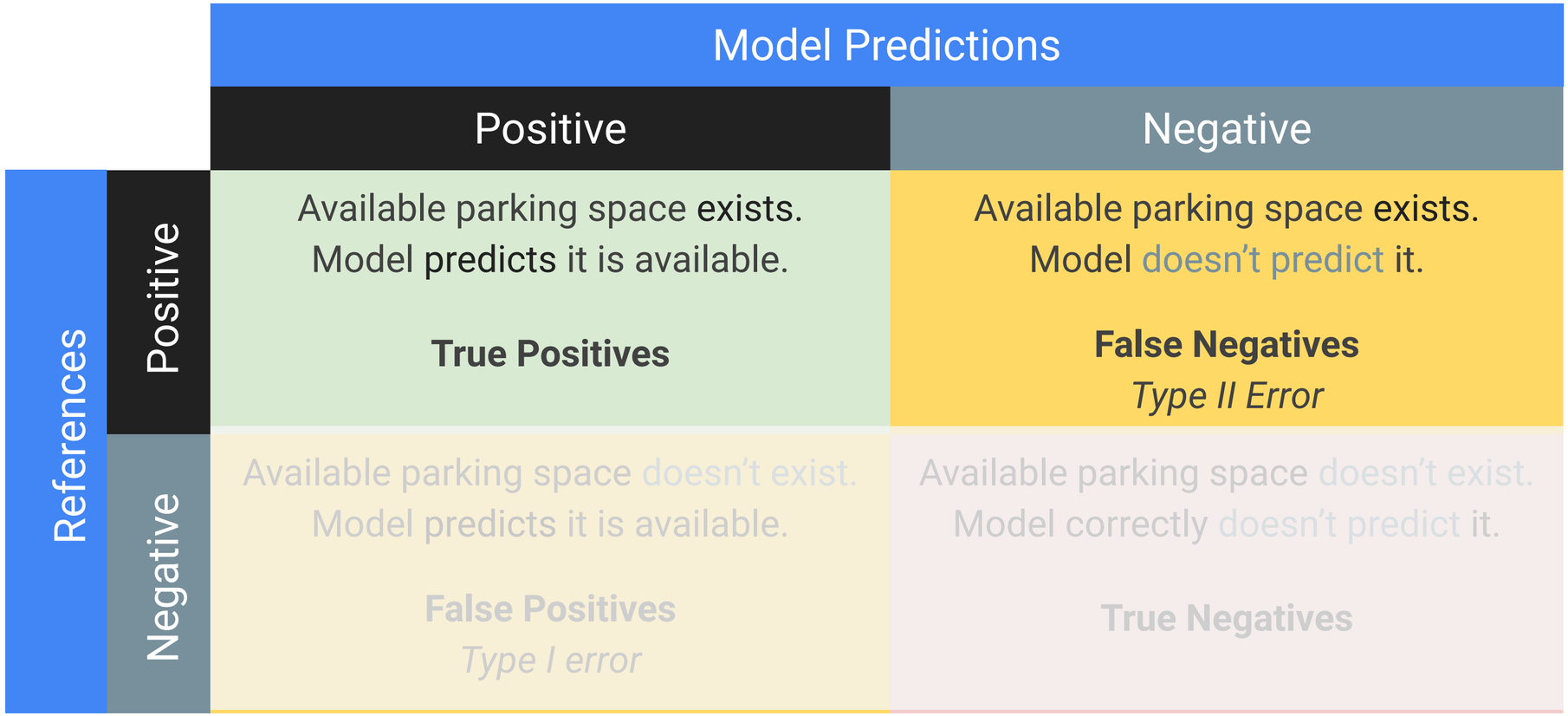
		Model Predictions: Positive	Model Predictions: Negative
References: Positive		Available parking space exists. Model predicts it is available. **True Positives**	Available parking space exists. Model doesn't predict it. **False Negatives** *Type II Error*
References: Negative		Available parking space doesn't exist. Model predicts it is available. **False Positives** *Type I error*	Available parking space doesn't exist. Model correctly doesn't predict it. **True Negatives**

Precision: True positives / total classified as positive

		Model Predictions: Positive	Model Predictions: Negative
References: Positive		Available parking space exists. Model predicts it is available. **True Positives**	Available parking space exists. Model doesn't predict it. **False Negatives** *Type II Error*
References: Negative		Available parking space doesn't exist. Model predicts it is available. **False Positives** *Type I error*	Available parking space doesn't exist. Model correctly doesn't predict it. **True Negatives**

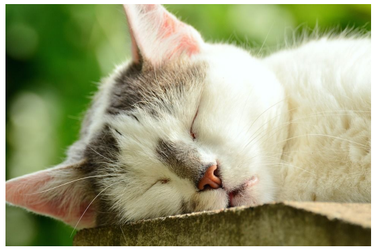
Precision

Recall: True positives / all actual positives in our reference



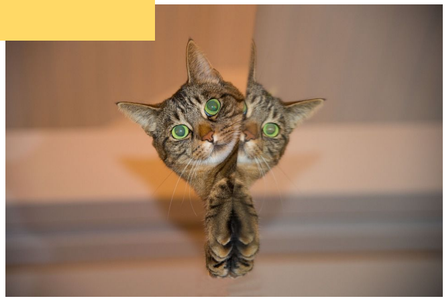
Recall

Classify each image below as either “cat” or “not cat”



Classify each image below as either “cat” or “not cat”

You:
Cat

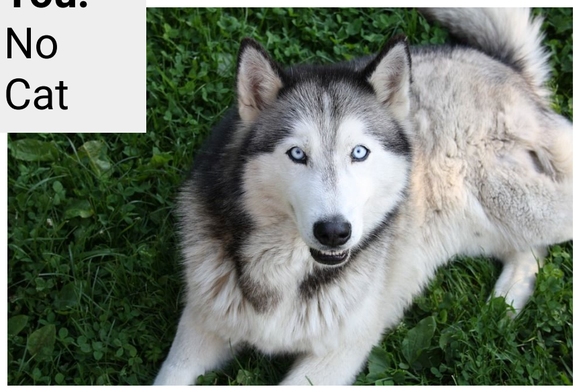


You:
Cat

You:
No
Cat

You:
Cat

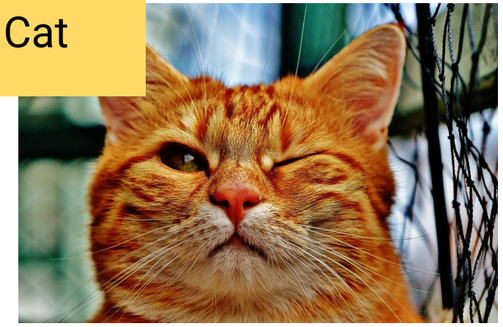
You:
No
Cat



You:
No
Cat

You:
No
Cat

You:
Cat



Hypothetical results from your classification ML model

You: Cat
ML: No Cat

You: Cat
ML: Cat

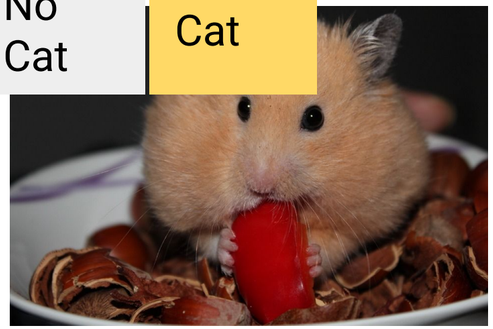
You: No Cat
ML: Cat

You: Cat
ML: No Cat

You: No Cat
ML: No Cat

You: No Cat
ML: Cat

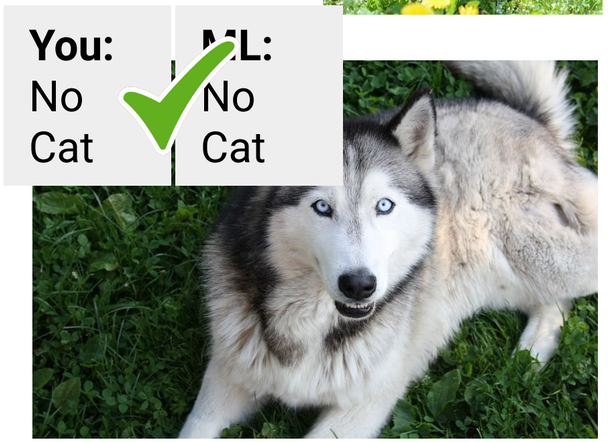
You: No Cat
ML: Cat



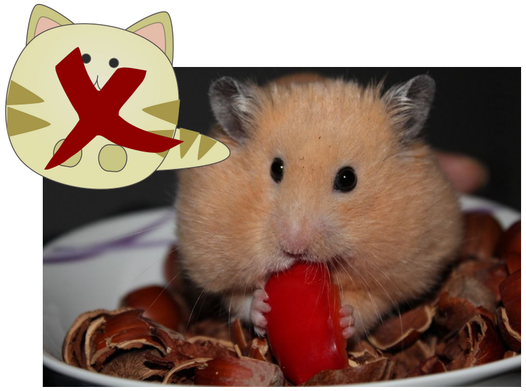
You: Cat
ML: Cat

Hypothetical results from your classification ML model

Accuracy = 3 / 8
= 0.375



How precise was your “cat” ML model?



Precision = TP / (TP + FP)

= 2 / 5 = 0.40

What recall did our model have?

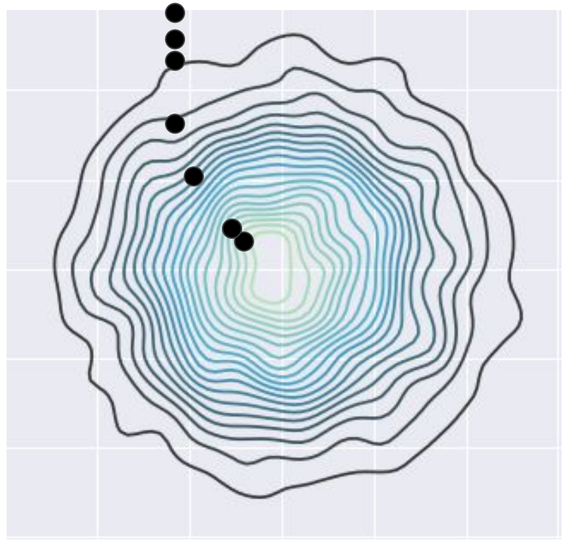
Recall = TP / (TP + FN)
= 2 / 4 = 0.50

Optimization identifies the best ML model parameters

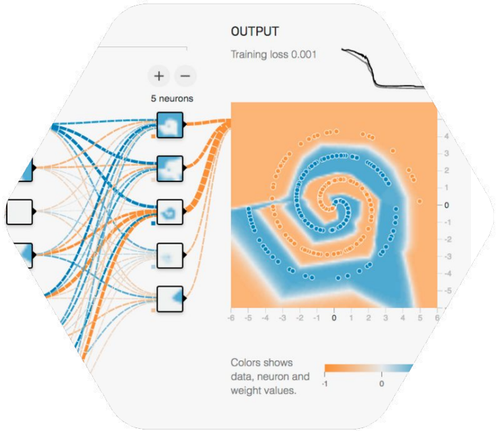
$$y = \underbrace{\beta + w^T}_{\text{Model Parameters}} \times X$$

output — Bias Term — weight — Input

Models are sets of parameters and hyper-parameters



Find the best parameters by optimizing loss functions through gradient descent



Experiment with neural networks in the TensorFlow playground

Launching into ML: Generalization and Sampling

Learn how to ...

Assess if your model is overfitting.

Gauge when to stop model training.

Create repeatable training, evaluation, and test datasets.

Establish performance benchmarks.

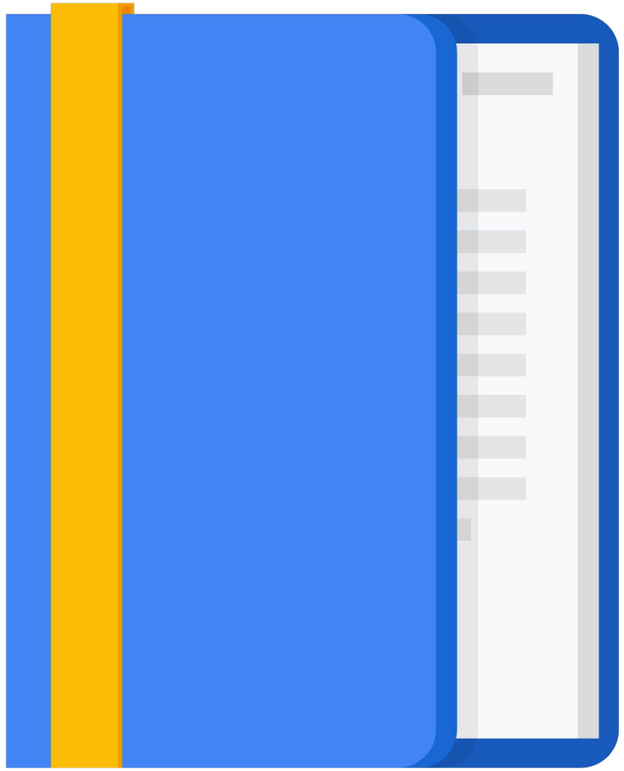
Agenda

Generalization

Sampling

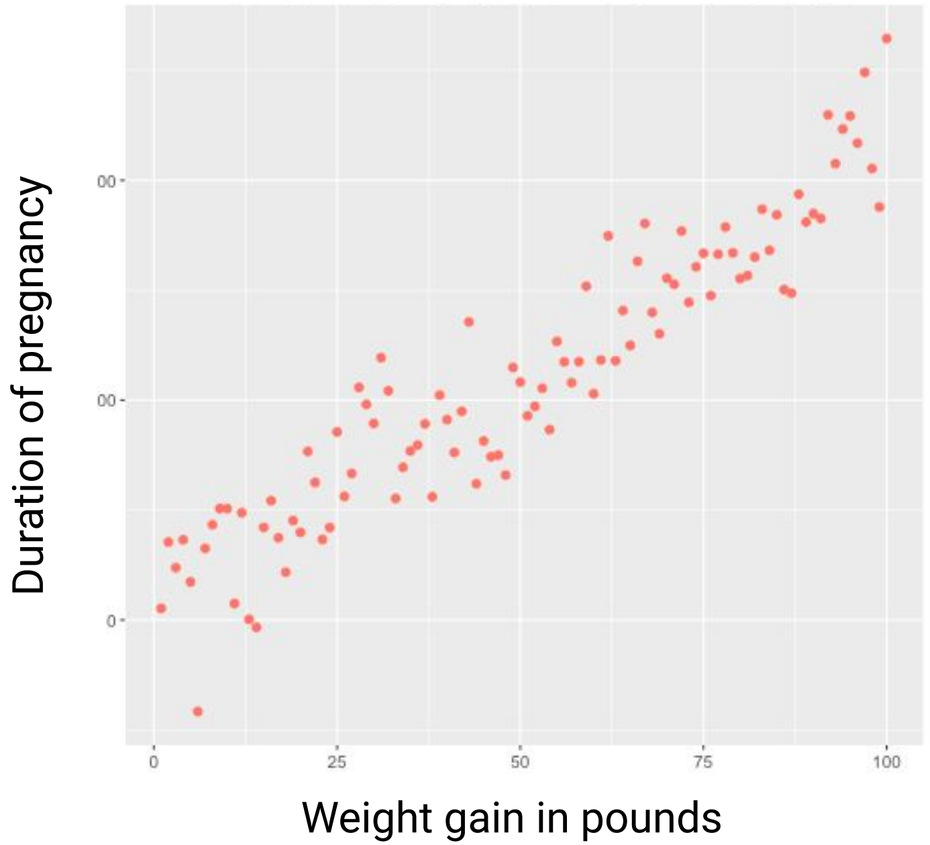
Lab: Maintaining Consistency in Training with Repeatable Datasets

Lab: Explore and Create Datasets



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

What is the error measure to optimize?

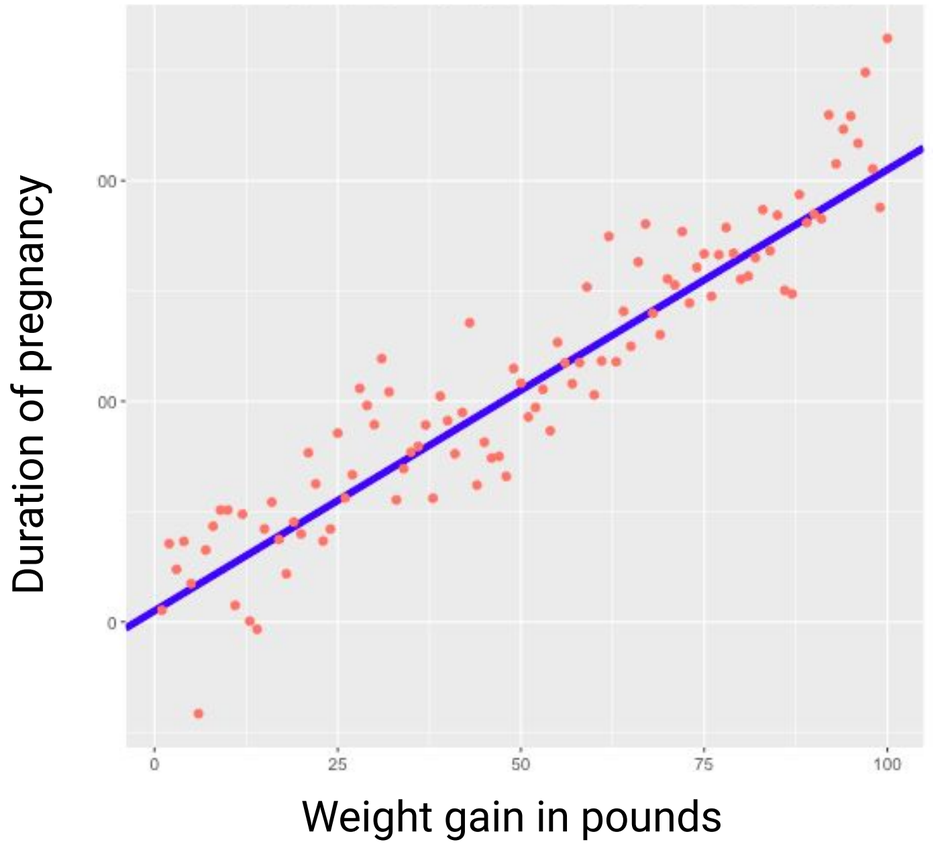


Model 1 is a linear model using linear regression

Red = training examples

Blue = model prediction for each baby

RMSE = 2.224

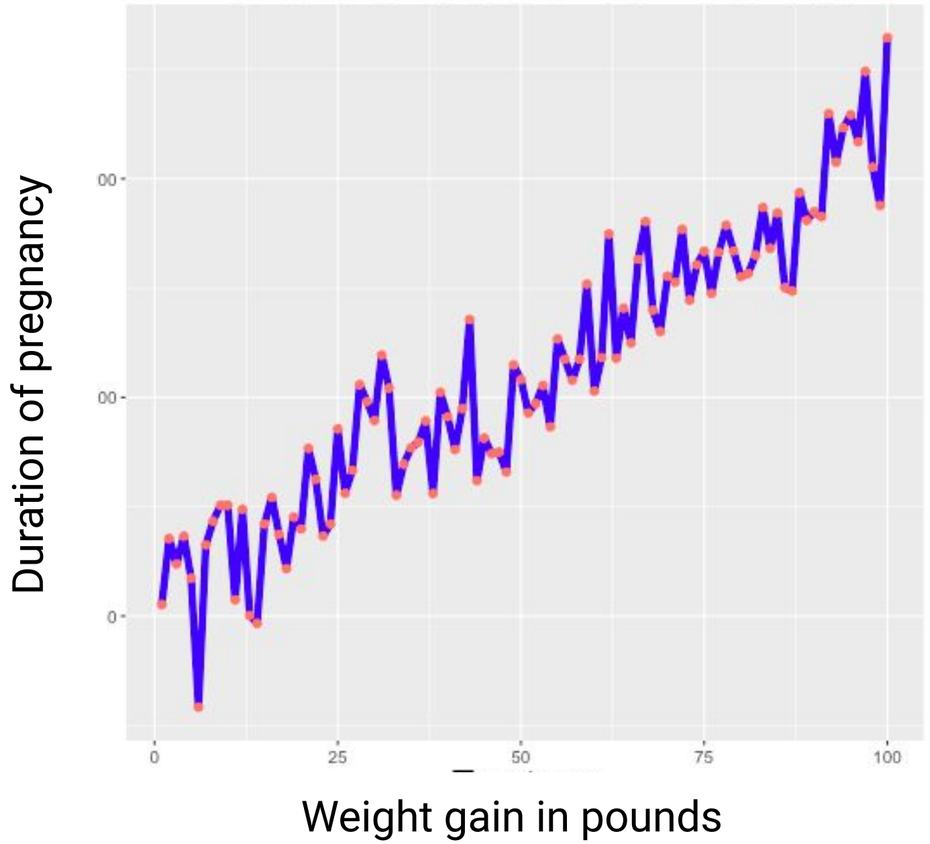


Model 2 has more free parameters

RMSE = 0

Which model is better?

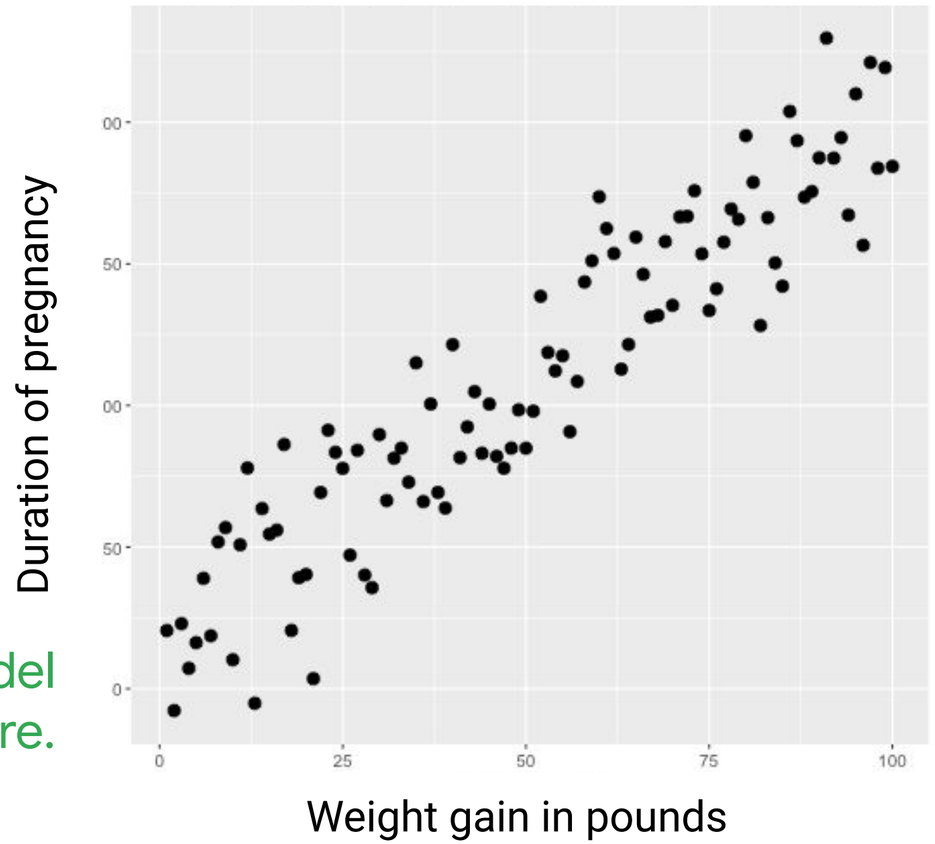
How can you tell?



Does the model generalize to new data?

Need data that were not used in training.

New data the model hasn't seen before.



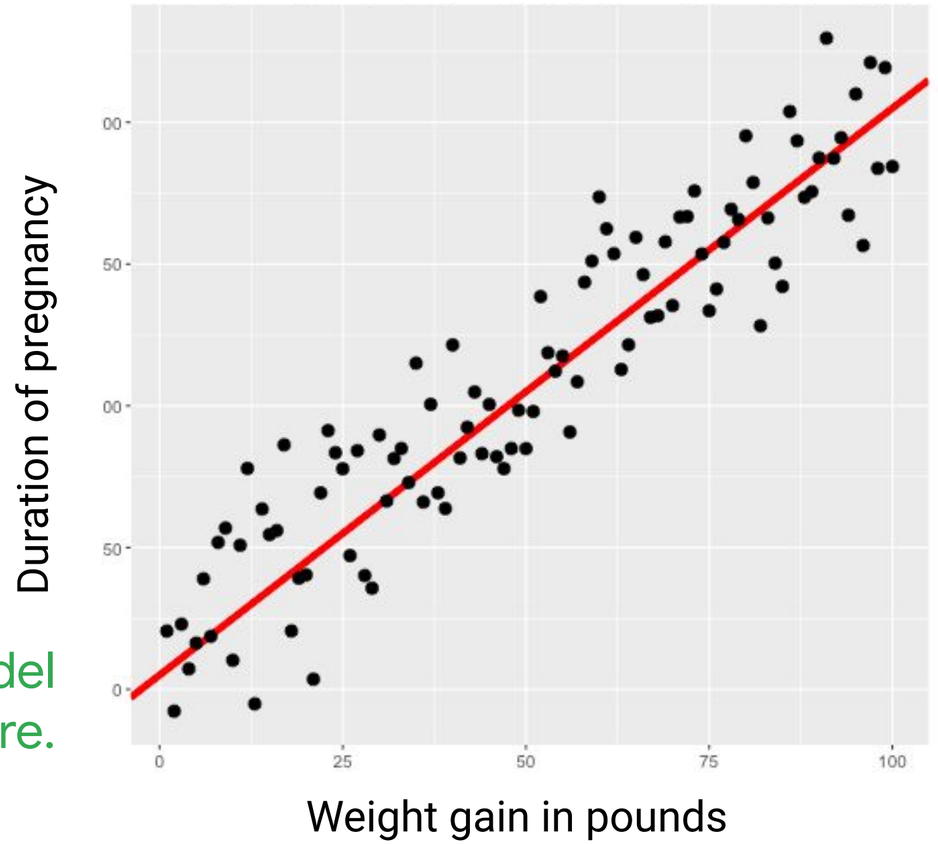
Model 1 generalizes well

Old RMSE = 2.224

New RMSE = 2.198

Pretty similar = good

New data the model hasn't seen before.



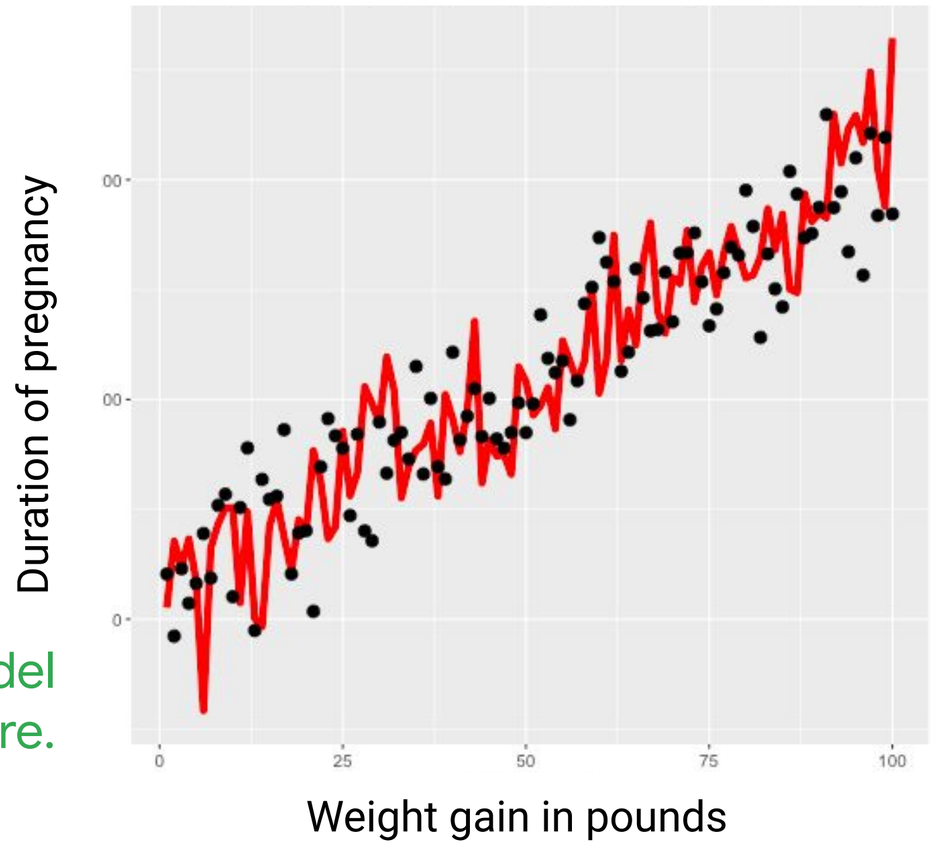
Model 2 does not generalize well

Old RMSE = 0

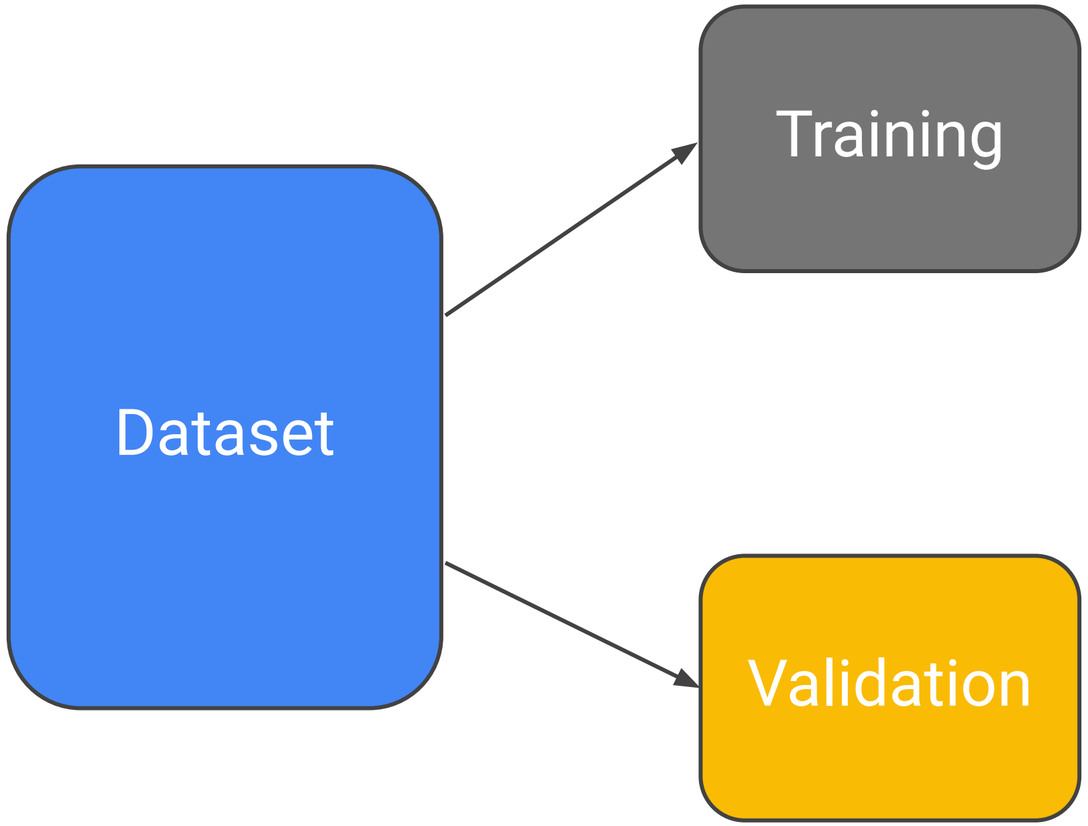
New RMSE = 3.2

This is a red flag

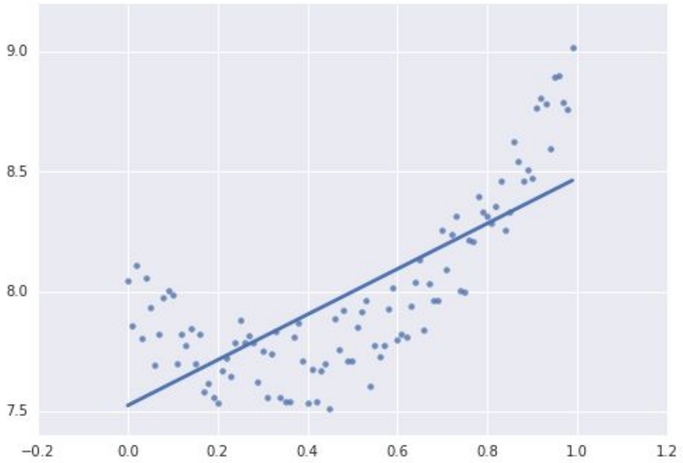
New data the model hasn't seen before.



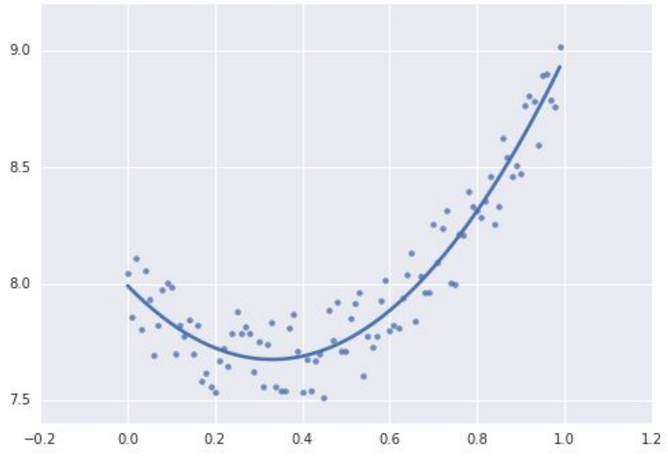
Split the dataset and experiment with models



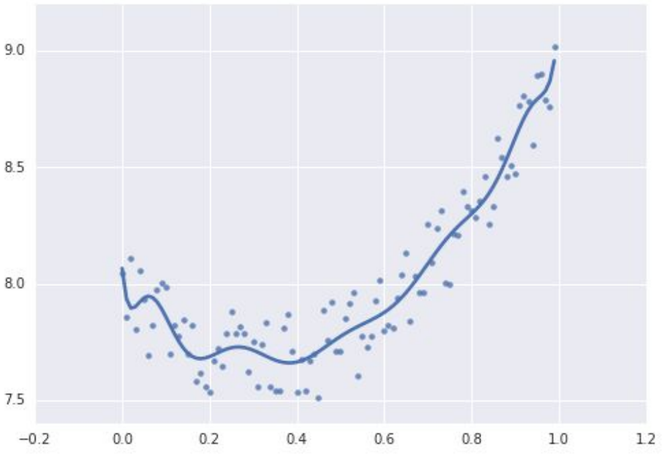
Beware of overfitting as you increase model complexity



Underfit

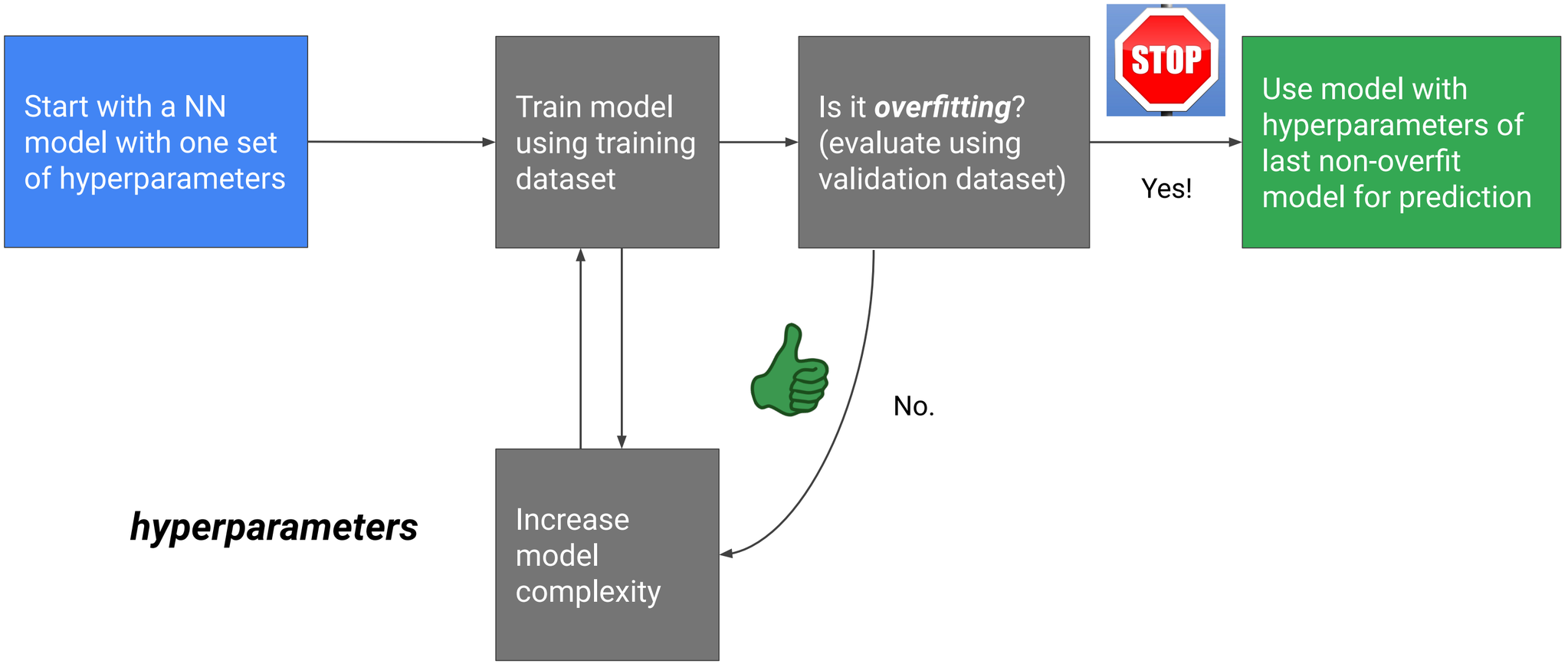


Fit

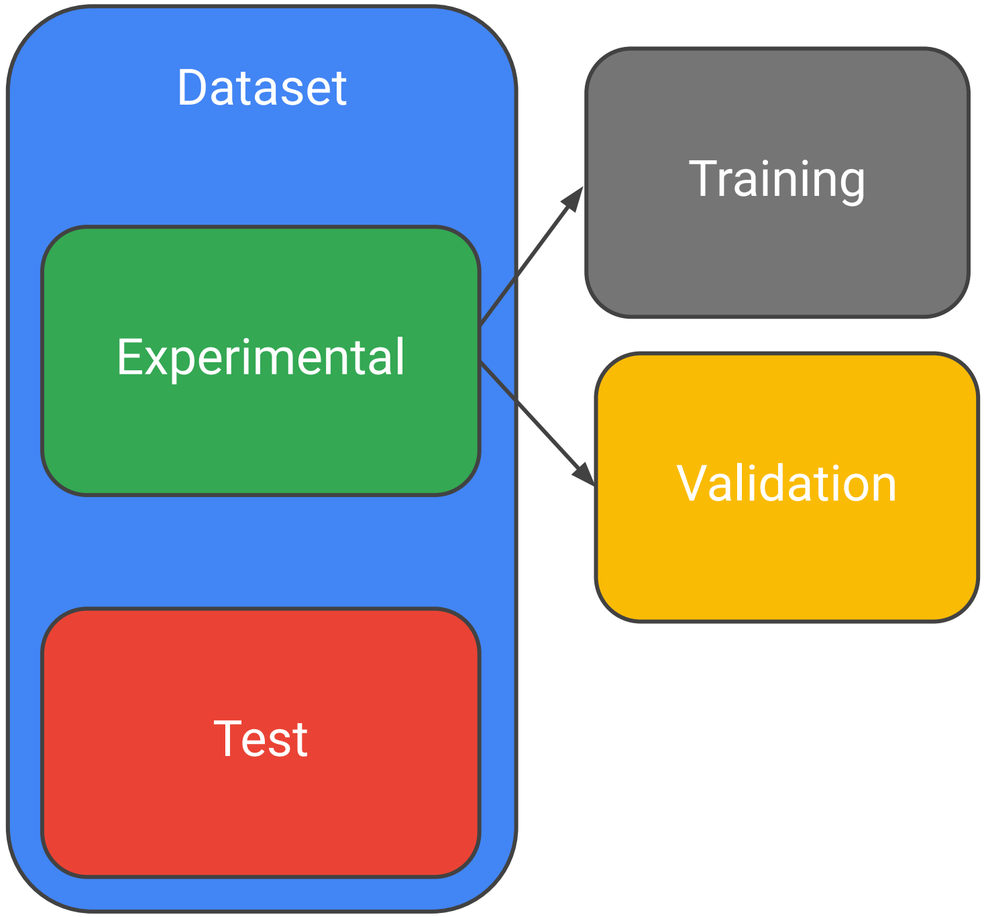


Overfit

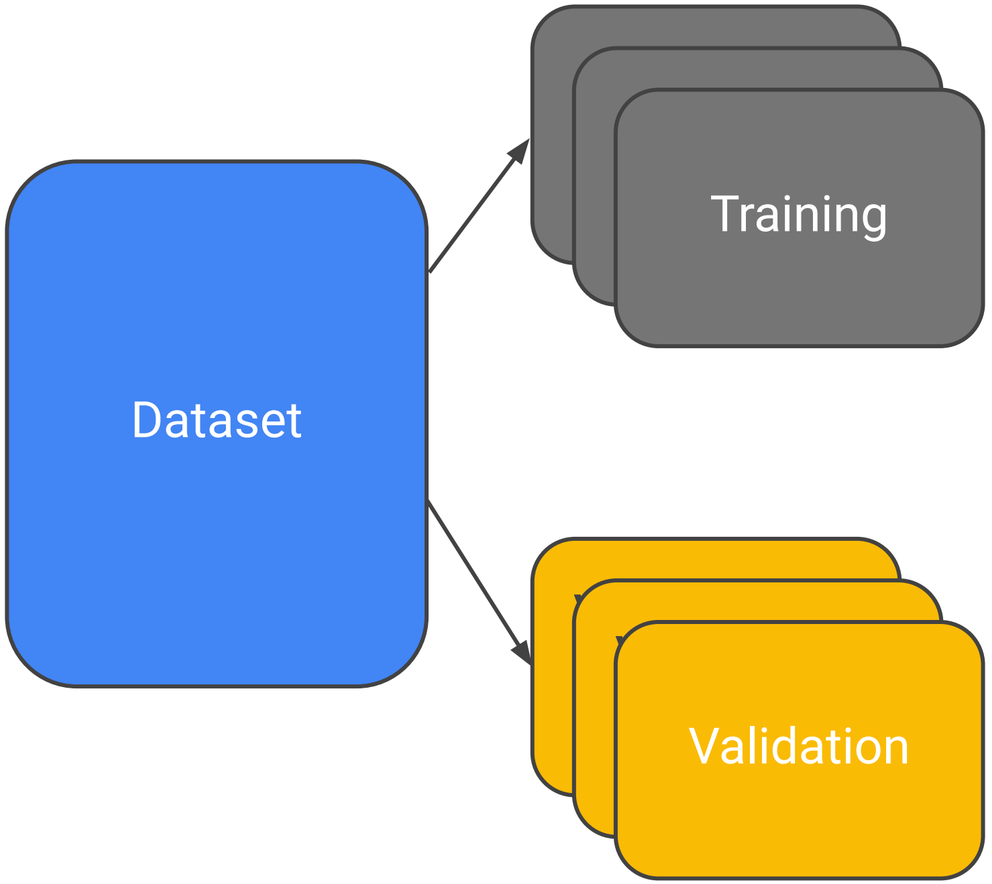
You can use the validation dataset to experiment with model complexity



Evaluate the final model with independent test data



Evaluate the final model with cross-validation



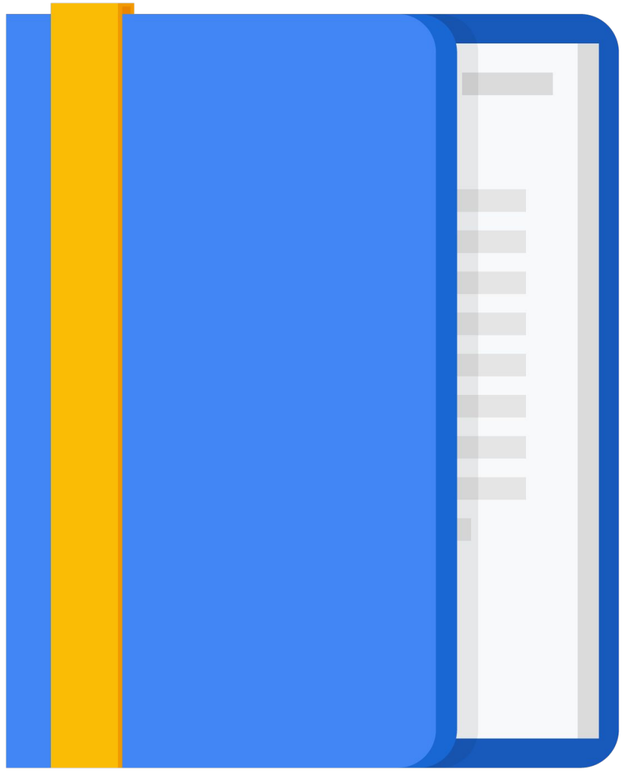
Agenda

Generalization

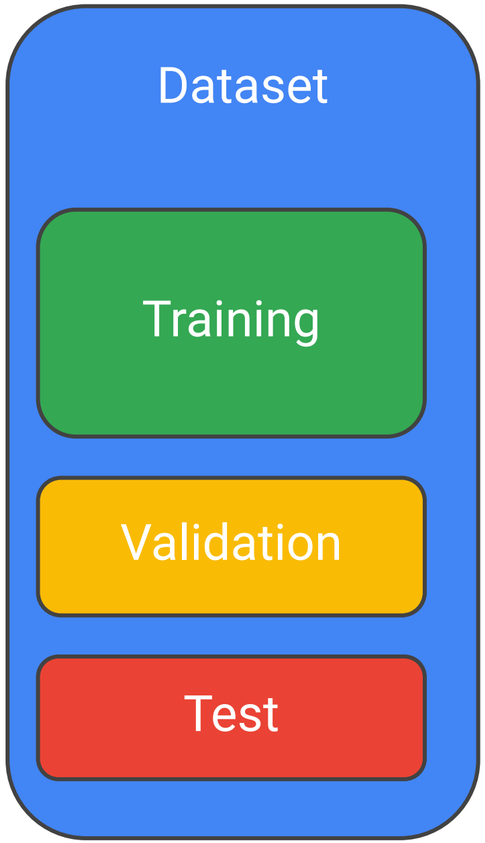
Sampling

Lab: Maintaining Consistency in Training with Repeatable Datasets

Lab: Explore and Create Datasets



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

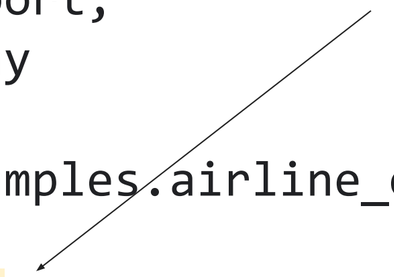


Row	date	airline	departure_airport	departure_schedule	arrival_airport	arrival_delay
1	2004-08-07	TZ	SRQ	1255	IND	-14.0
2	2004-03-05	TZ	SRQ	2117	IND	-9.0
3	2004-04-12	TZ	SRQ	2000	IND	-17.0
4	2003-04-16	TZ	SRQ	1215	IND	-5.0
5	2005-03-20	TZ	SRQ	645	IND	14.0
6	2003-04-06	TZ	SRQ	1235	IND	-8.0

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

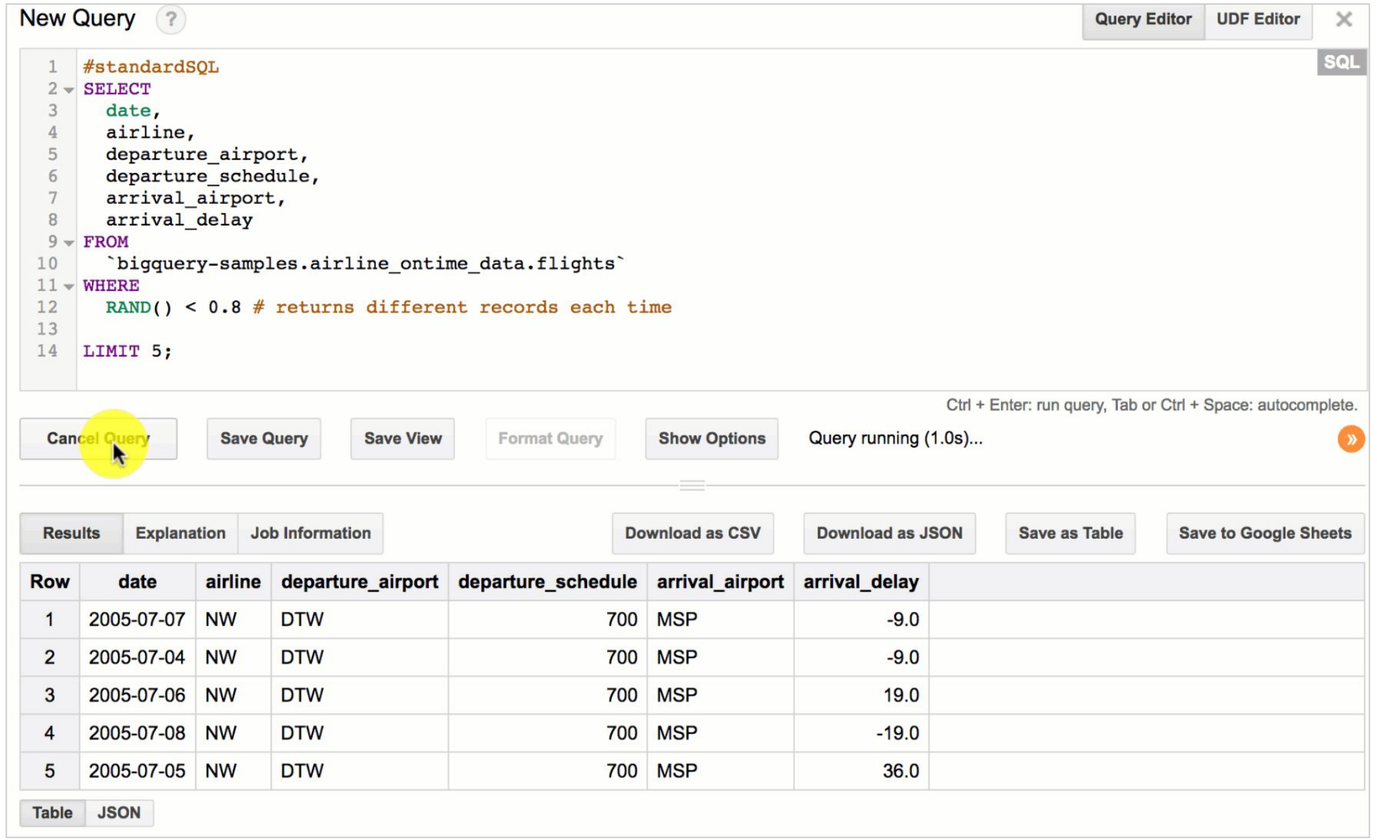
You need to know which specific data was involved in training, validation, and testing.

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 →



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

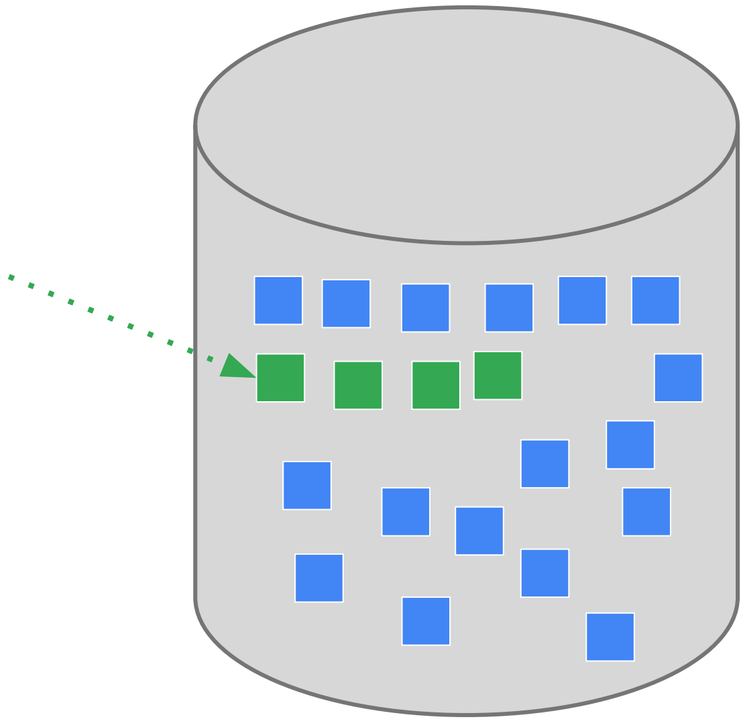
We hypothesize that flight delay depends on the carrier, time of day, weather, and airport characteristics (# of runways, etc.) We want to predict flight delays. What field should we split our data on?

- 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

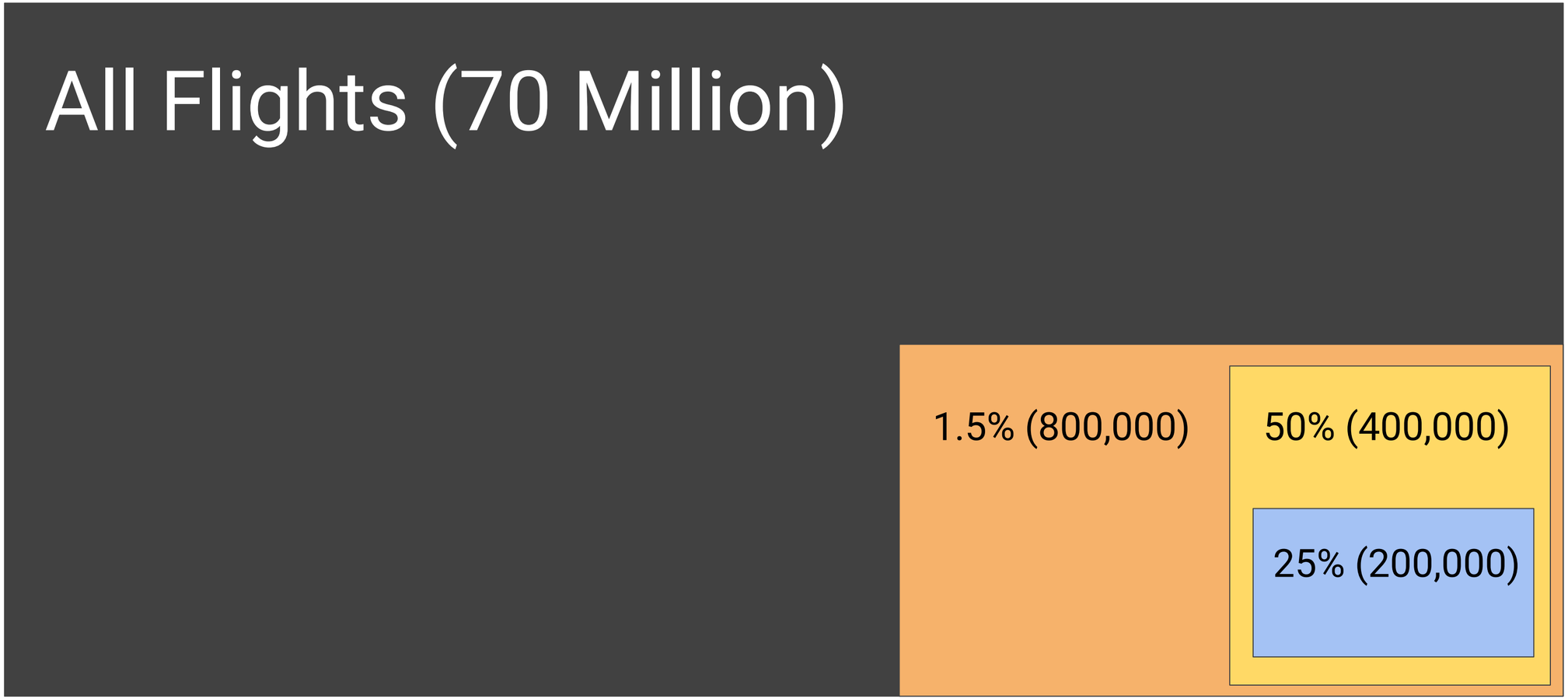
```
#standardSQL
SELECT
  date,
  airline,
  departure_airport,
  departure_schedule,
  arrival_airport,
  arrival_delay
FROM
 `bigquery-samples.airline_ontime_data.flights`
WHERE
  MOD(ABS(FARM_FINGERPRINT(date)),70) = 0
     AND
  MOD(ABS(FARM_FINGERPRINT(date)),10) < 8
```

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!)

How we want to split our data



We can extend this to creating 3 splits

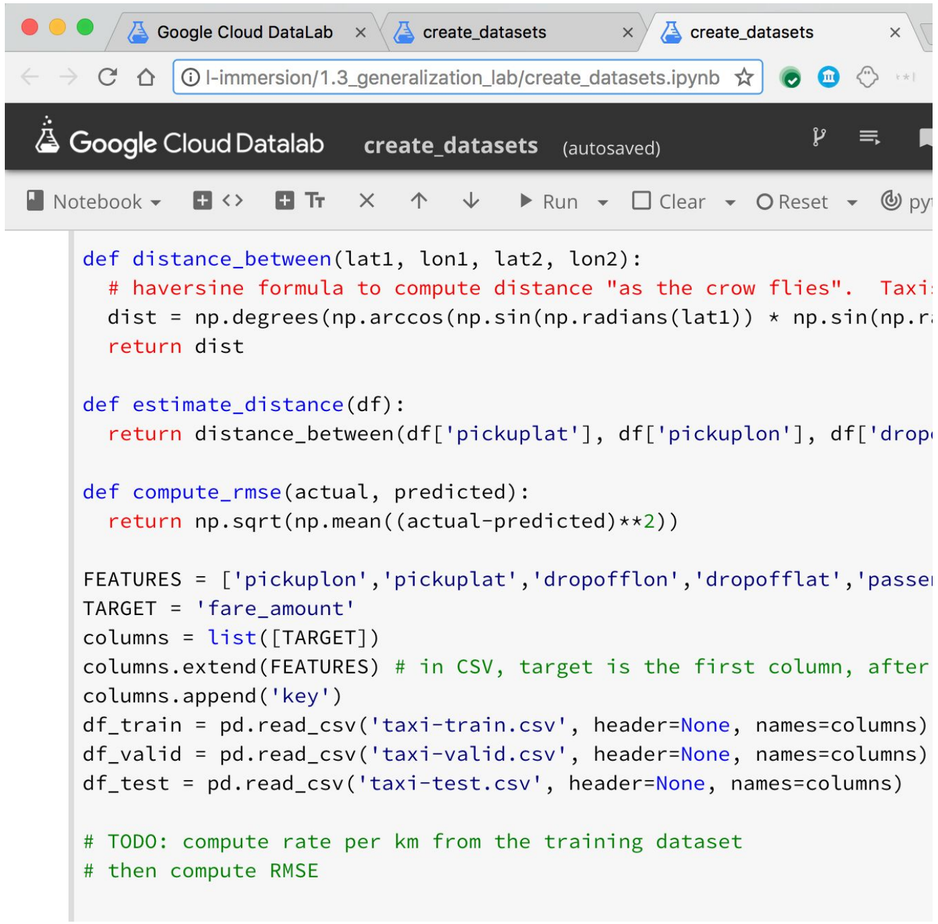
```
#standardSQL
SELECT
  date,
  airline,
  departure_airport,
  departure_schedule,
  arrival_airport,
  arrival_delay
FROM
 `bigquery-samples.airline_ontime_data.flights`
WHERE
  MOD(ABS(FARM_FINGERPRINT(date)),70) = 0
     AND
  MOD(ABS(FARM_FINGERPRINT(date)),700) >= 350
    AND
  MOD(ABS(FARM_FINGERPRINT(date)),700)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

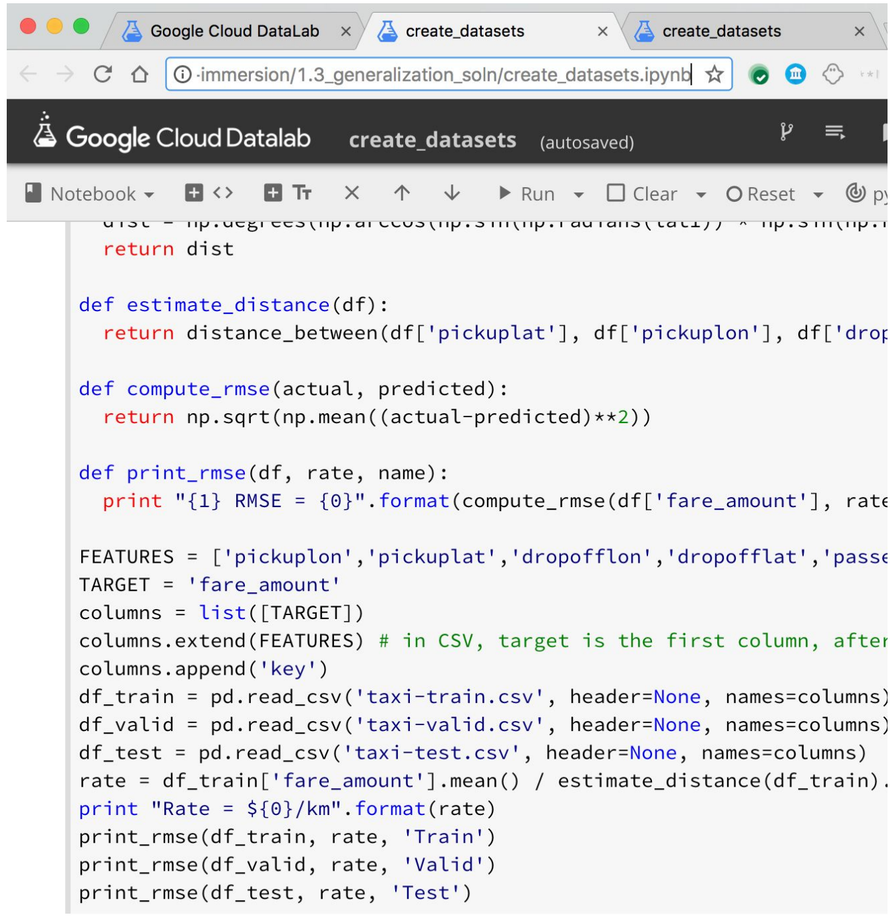
Then take 1 in 70 flights.

Ignore the 50% of the dataset (training).

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

Solutions to all the labs exist in the repository





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?

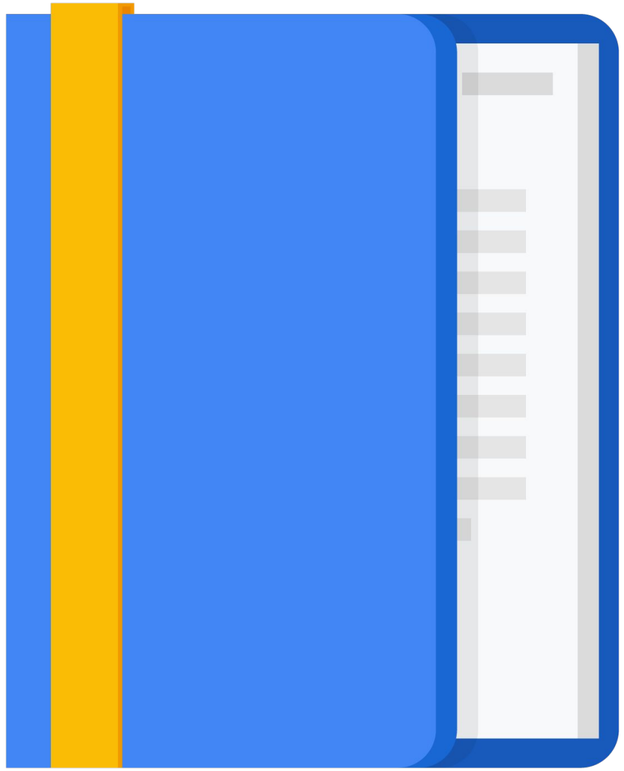
Agenda

Generalization

Sampling

Lab: Maintaining Consistency in Training with Repeatable Datasets

Lab: Explore and Create Datasets



Lab Intro

Maintaining Consistency in Training with Repeatable splits

Google Cloud

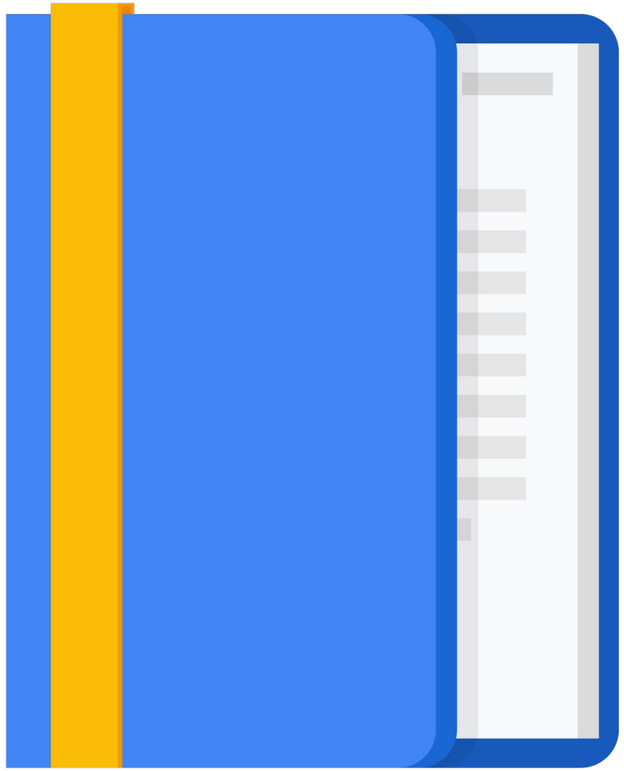
Agenda

Generalization

Sampling

Lab: Maintaining Consistency in Training with Repeatable Datasets

Lab: Explore and Create Datasets

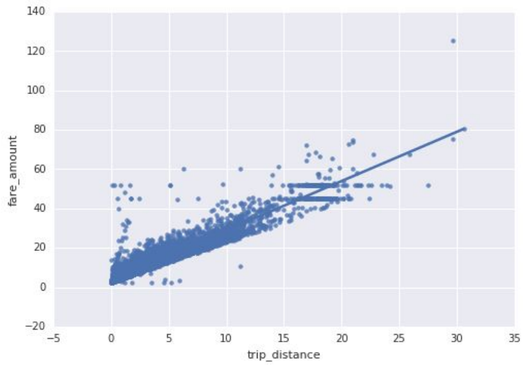


Lab Intro

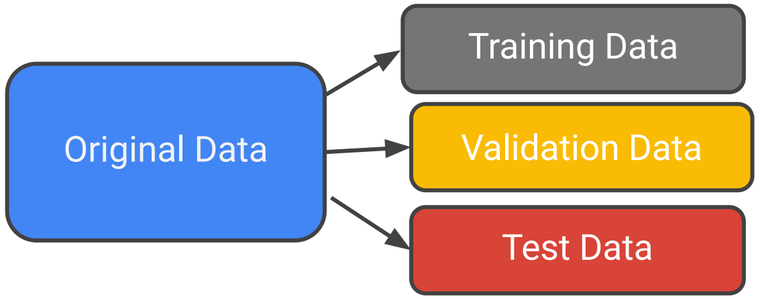
Explore and Create Datasets

Google Cloud

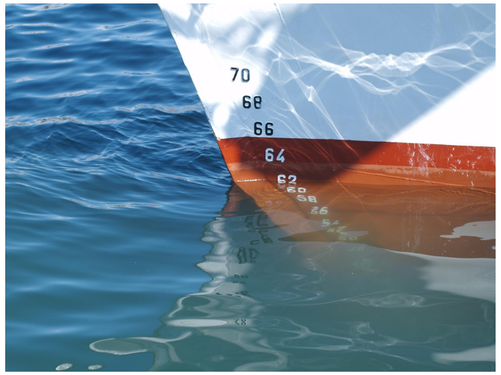
Lab steps



1. Explore



2. Create ML Datasets

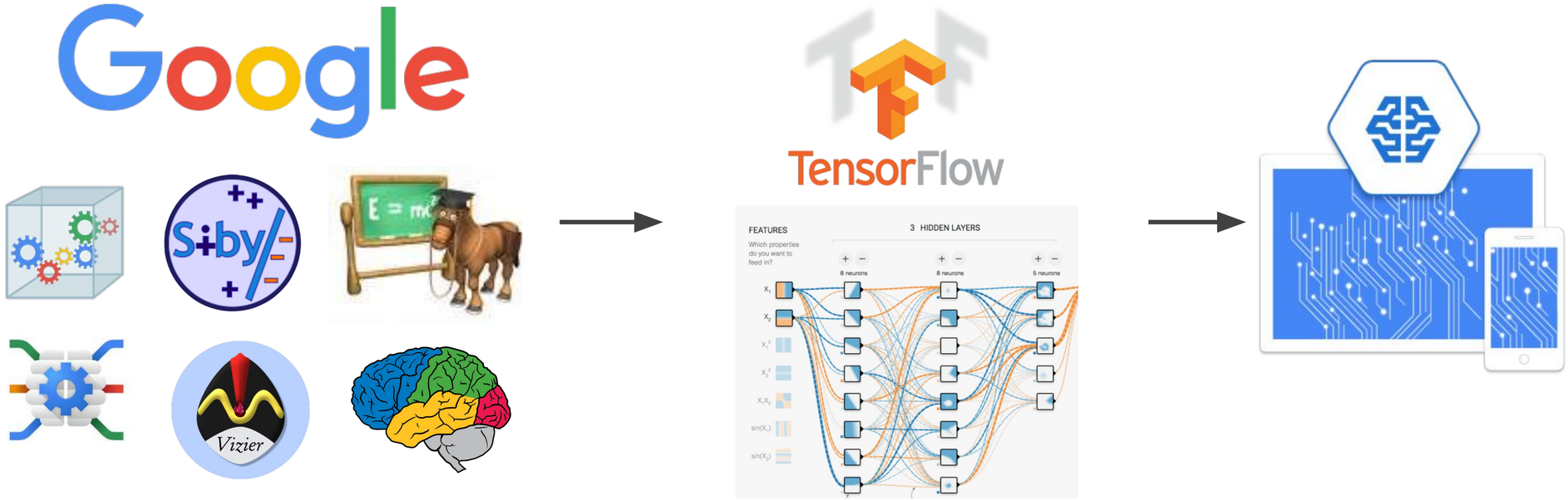


3. Benchmark

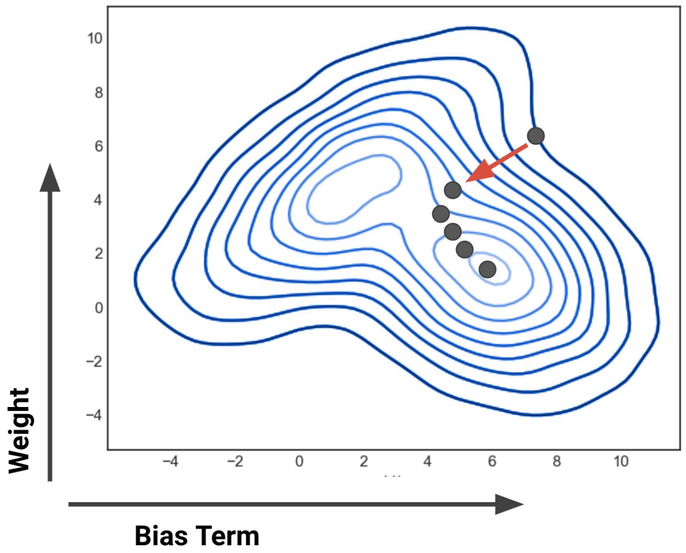
Google Cloud

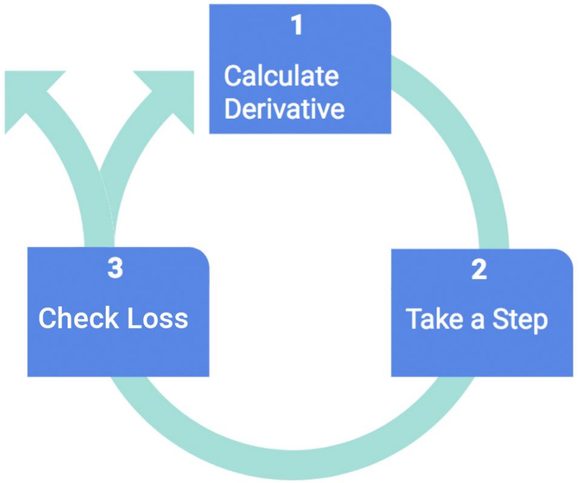
Launching into ML:
Course summary

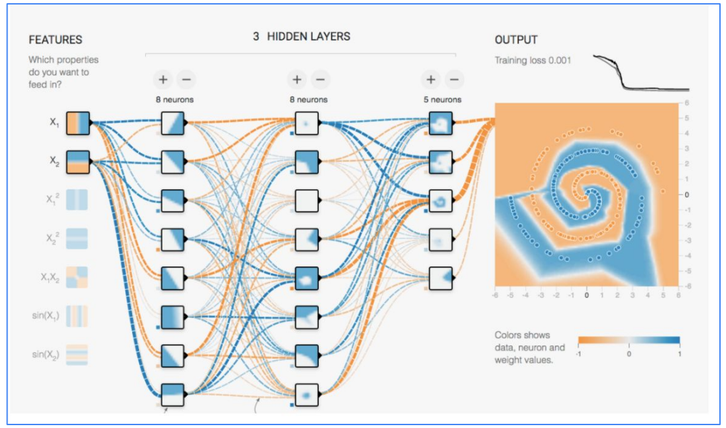
TensorFlow and AI Platform are built on the past experiences of Google production ML systems



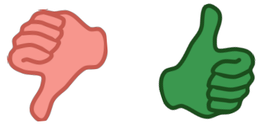
Optimize your ML models with gradient descent to minimize error



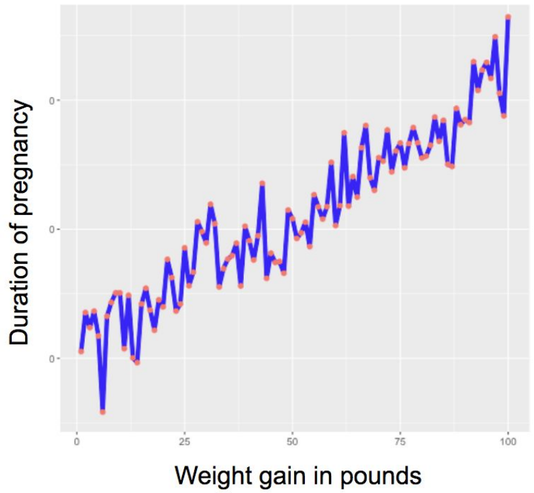




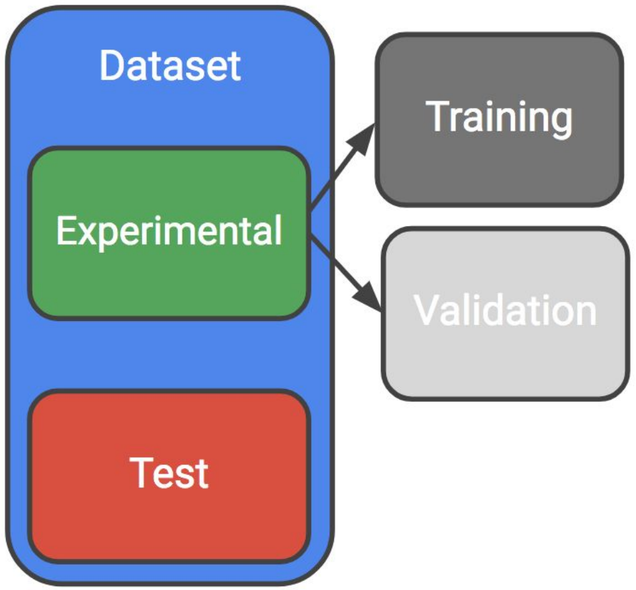
RMSE and Cross-Entropy



Generalization and sampling create repeatable datasets that can be used for training, validation, and testing



Overfitting



Split your data

Estimating Taxi Fare

Machine Learning on Google Cloud learning path

1. How Google Does Machine Learning
2. Launching into ML
3. Introduction to TensorFlow
4. Feature Engineering
5. The Art and Science of ML


Google Cloud