Google Cloud

Partner Certification Academy

Professional Machine Learning Engineer

The information in this presentation is classified:

Google confidential & proprietary

⚠️ This presentation is shared with you under NDA.

- Do **not** record or take screenshots of this presentation.
- Do **not** share or otherwise distribute the information in this presentation with anyone **inside** or **outside** of your organization.

Thank you!

Google Cloud

Source Materials

Some of this program's content has been sourced from the following resources:

- Google Cloud certification site
- Google Cloud documentation
- Google Cloud console
- Google Cloud courses and workshops
- Google Cloud white papers
- Google Cloud Blog
- Google Cloud YouTube channel
- Google Cloud samples
- Google codelabs
- Google Cloud partner-exclusive resources

Google Cloud

Google Cloud Skills Boost for Partners

- Professional Machine Learning Engineer Certification
- Cloud Skills Boost for Partners Professional Machine Learning Engineer Learning Path
- Partner Learning Services Instructor-Led PMLE Curriculum

Google Cloud Partner Advantage

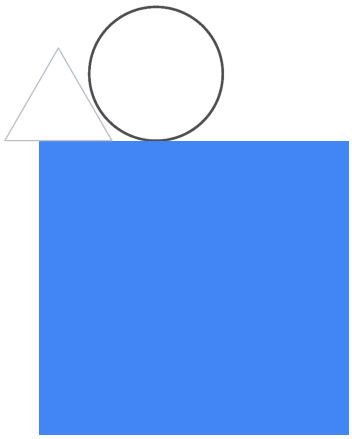
- Best practices for implementing machine learning on Google Cloud
- Artificial Intelligence
- End-to-End MLOps Go-to-Market Kit

Session Logistics

- When you have a question, please:
  - Click the Raise hand button in Google Meet.
  - Or add your question to the Q&A section of Google Meet.
  - Please note that answers may be deferred until the end of the session.
- These slides are available in the Student Lecture section of your Qwiklabs classroom.
- The session is **not recorded.**
- Google Meet does not have persistent chat.
  - If you get disconnected, you will lose the chat history.
  - Please copy any important URLs to a local text file as they appear in the chat.

Google Cloud Partner Learning Programs

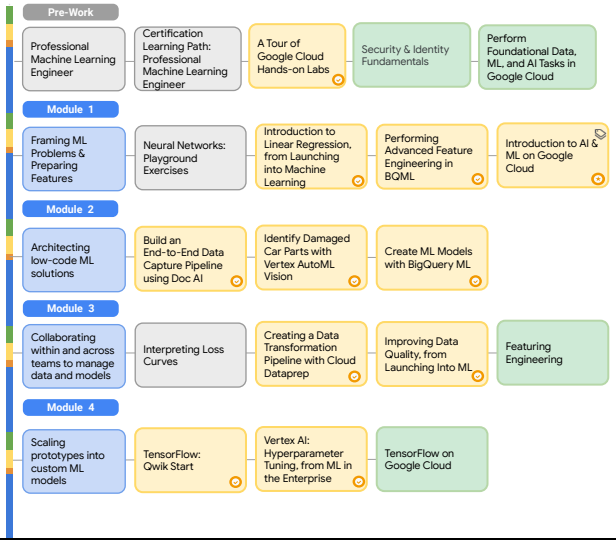
- Partner Certification Academy
- Partner Delivery Readiness Index (DRI)
- Cloud Skills Boost for Partners
- Partner Advantage



PARTNER CERTIFICATION ACADEMY

Professional Machine Learning Engineer

On-demand Course
Hands-on Skill Badge
Hands-on Lab
Resource
Live Virtual Workshop
required to earn exam voucher



A Professional Machine Learning Engineer builds, evaluates, productionizes, and optimizes ML models by using Google Cloud technologies and knowledge of proven models and techniques. The ML Engineer:

- handles large, complex datasets and creates repeatable, reusable code.
- considers responsible AI and fairness throughout the ML model development process, and collaborates closely with other job roles to ensure long-term success of ML-based applications.
- has strong programming skills and experience with data platforms and distributed data processing tools.
- is proficient in the areas of model architecture, data and ML pipeline creation, and metrics interpretation.
- is familiar with foundational concepts of MLOps, application development, infrastructure management, data engineering, and data governance.
- makes ML accessible and enables teams across the organization.

By training, retraining, deploying, scheduling, monitoring, and improving models, the ML Engineer designs and creates scalable, performant solutions.

Recommended candidate:

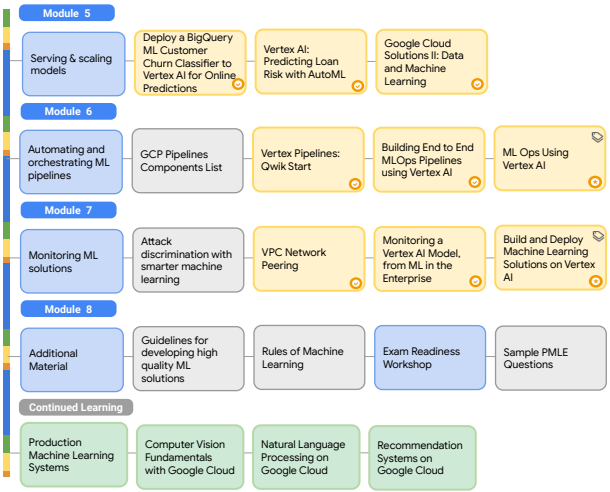
- Has in-depth experience setting up cloud environments for an organization
- Has experience deploying services and solutions based on business requirements

Google Cloud

PARTNER CERTIFICATION ACADEMY

Professional Machine Learning Engineer

On-demand Course
Hands-on Skill Badge
Hands-on Lab
Resource
Live Virtual Workshop
required to earn exam voucher



A Professional Machine Learning Engineer builds, evaluates, productionizes, and optimizes ML models by using Google Cloud technologies and knowledge of proven models and techniques. The ML Engineer:

- handles large, complex datasets and creates repeatable, reusable code.
- considers responsible AI and fairness throughout the ML model development process, and collaborates closely with other job roles to ensure long-term success of ML-based applications.
- has strong programming skills and experience with data platforms and distributed data processing tools.
- is proficient in the areas of model architecture, data and ML pipeline creation, and metrics interpretation.
- is familiar with foundational concepts of MLOps, application development, infrastructure management, data engineering, and data governance.
- makes ML accessible and enables teams across the organization.

By training, retraining, deploying, scheduling, monitoring, and improving models, the ML Engineer designs and creates scalable, performant solutions.

Recommended candidate:

- Has in-depth experience setting up cloud environments for an organization
- Has experience deploying services and solutions based on business requirements

Google Cloud

Learner Commitment

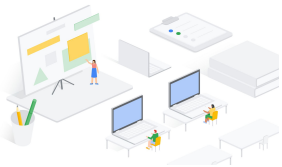
Each week, learners are to complete the learning path's course content, Cloud Skills Boost for Partner Quests/Challenge Labs and material that the mentor has recommended that will support learning.

- **Workshop Day:** Meet for the cohort's weekly 'general session'. **(≅ 2 hours)**
- **During the week:** Complete the week's course, perform hands-on labs, review any additional material suggested material for the week. **(≅ 8 - 16 hours)**

➔ **Important:** Learners must allocate time between each weekly session to study and familiarize themselves with any prerequisite knowledge they may lack. It is also recommended that learners complete the next week's course prior to the scheduled workshop.

Path to Service Excellence

Certification



Advanced Solutions Training

Delivery Readiness Index

Certification is just one step on your professional journey. Google Cloud also offers our partners access to advanced solutions training, and a new quality-focused program called Delivery Readiness Index (DRI) to help you achieve service excellence with your customers.

Benchmark your skills with DRI

DRI helps to benchmark partner proficiency and capability at any point during the customer journey however should be used primarily as a lead measure to predict and prepare for partner delivery success.

DRI assesses and analyzes Partner Consultant GCP proficiency by creating a DRI Profile inclusive of their GCP knowledge, skills, and experience.

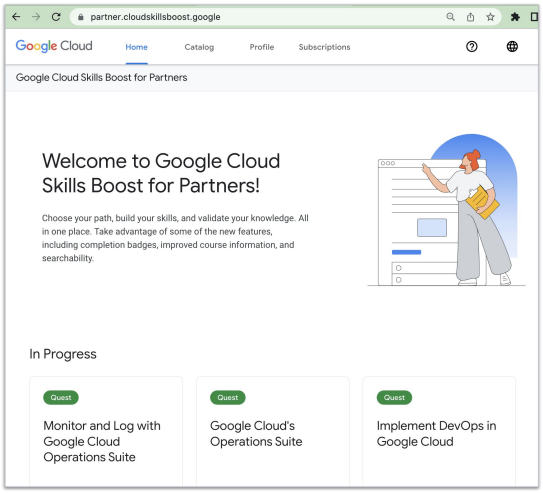
With the DRI insights, we can prescriptively advise the partner project team on the ground and bridge niche capability gaps.

DRI also takes action. For partner consultants, DRI generates a tailored L&D plan that prescribes personalized learning, training, and skill development to build GCP proficiency.

Google Cloud Skills Boost for Partners

https://partner.cloudskillsboost.google/

- On-demand course content
- Hands-on labs
- Skill Badges
- **FREE** to Google Cloud Partners!

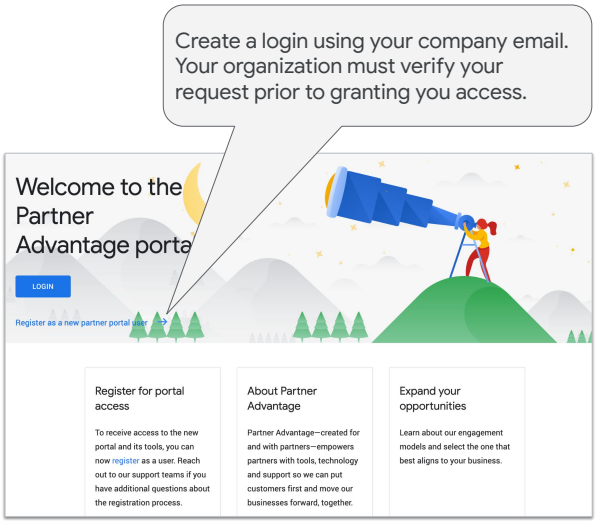


Proprietary + Confidential

Google Cloud

Partner Advantage

- Resources for Google Cloud partner organizations:
  - Recent announcements
  - Solutions/role-based training
  - Live/pre-recorded webinars on various topics
    - Partner Advantage Live Webinars
- Complements the certification self-study material presented on Google Cloud Skills Boost for Partners
- Helpful Links:
  - Getting started on Partner Advantage
  - Join Partner Advantage
  - Get help accessing Partner Advantage



https://www.partneradvantage.goog

Google Cloud

The getting started link:
https://support.google.com/googlecloud/topic/9198654#zippy=

Note the top section, **"Getting Started & User Guides"** and two key documents → Direct Partners to this if they need to enroll into Partner Advantage

1. Logging in to the Partner Advantage Portal - Quick Reference Guide
2. Enrolling in the Partner Advantage Program - Quick Reference Guide

Focus from this point on:

Some context on enrolling in PA:
Access to Partner Portal is given in 2 ways

- Partner Admin Led: Partner Administrator at Partner Company can set up users
- User Led: User can go through Self Registration
  - https://www.partneradvantage.goog/GCPPRM/s/partneradvantageportallogin?language=en_US
  - Or directly to the User Registration Form, https://www.partneradvantage.goog/GCPPRM/s/partnerselfregistration?language=en_US

Please Note

- After a user self-registers, they receive an email that essentially states:

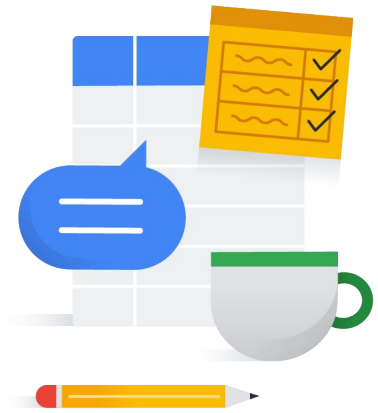
    - “Hi {Partner Name}, you are one step away from joining the Google Cloud Partner Advantage Community. Please click to continue with the user registration process. See you in the cloud, The Partner Advantage Team
- Once registered, they can access limited content until their **Partner Administrator approves the user**
- Their Partner Administrator also receive an email notifying them that a member of their organization has registered themselves on their organization’s Google Cloud Partner Advantage account.
    - It also states that this user has limited access to the portal
    - They are provided instructions on how to review and provision the appropriate access for the user that has registered
- Once their admin approves the user, they receive an email that states:
    - Hi {User Name}, Your Partner Administrator has updated your access to the Google Cloud Partner Advantage portal. You have been granted edit access to additional account information on the portal on behalf of your organization to help build your business. For additional access needs, please work with your Partner Administrator. See you in the cloud, The Partner Advantage Team

The net takeaway is, on the Support Page (the first link on this slide) Google Cloud Partner Advantage Support, there’s a section **“Issue accessing Partner Advantage Portal? Click here for troubleshooting steps”**

- The source of their issue can be related to the different items shown
- Additionally, there’s a Partner Administrator / Partner Adminstrator Team at their partner organization that has to approve their access.. Until that step is completed, they will have access issues/limitation. They will need to identify who this person or team is at their organization

Program issues or concerns?

- Problems with **accessing** Cloud Skills Boost for Partners
  - cloud-partner-training@google.com
- Problems with **a lab** (locked out, etc.)
  - support@qwiklabs.com
- Problems with accessing Partner Advantage
  - https://support.google.com/googlecloud/topic/9198654



- Problems with accessing **Cloud Skills Boost for Partners**
  - cloud-partner-training@google.com
- Problems with a **lab** (locked out, etc.)
  - support@qwiklab.com
- Problems with accessing **Partner Advantage**
  - https://support.google.com/googlecloud/topic/9198654

Module 3

Collaborating within and across teams to manage data and models

Module Agenda

01 Exploring and preprocessing organization-wide data

02 Model prototyping using Jupyter notebooks

03 Understanding Model Metrics

Exploring and preprocessing organization-wide data

Dataproc is managed Hadoop on Google Cloud Platform.

- Fast, easy, managed way to run Hadoop and Spark/Hive/Pig on Google Cloud.
- Create clusters in 90 seconds or less on average.
- Scale clusters up and down even when jobs are running.

Apache Hadoop is an open-source framework for big data. It is based on the MapReduce programming model, which Google invented and published. The MapReduce model, at its simplest, means that one function -- traditionally called the “map” function -- runs in parallel across a massive dataset to produce intermediate results; and another function -- traditionally called the “reduce” function -- builds a final result set based on all those intermediate results. The term “Hadoop” is often used informally to encompass Apache Hadoop itself and related projects, such as Apache Spark, Apache Pig, and Apache Hive.

Dataproc is a fast, easy, managed way to run Hadoop, Spark, Hive, and Pig on Google Cloud. All you have to do is to request a Hadoop cluster. It will be built for you in 90 seconds or less, on top of Compute Engine virtual machines whose number and type you can control. If you need more or less processing power while your cluster’s running, you can scale it up or down. You can use the default configuration for the Hadoop software in your cluster, or you can customize it. And you can monitor your cluster using Stackdriver.

When should we use Dataproc?

- Easily migrate on-premises Hadoop jobs to the cloud.
- Quickly analyze data (like log data) stored in Cloud Storage; create a cluster in 90 seconds or less on average, and then delete it immediately.
- Use Spark/Spark SQL to quickly perform data mining and analysis.
- Use Spark Machine Learning Libraries (MLlib) to run classification algorithms.

Running on-premises Hadoop jobs requires a hardware investment. On the other hand, running these jobs in Dataproc allows you to pay only for hardware resources during the life of the ephemeral customer you create. You can further save money using preemptible instances for batch processing.

You can also save money by telling Dataproc to use preemptible Compute Engine instances for your batch processing. You have to make sure that your jobs can be restarted cleanly if they're terminated and you get a significant break in the cost of the instances. At the time this video was made, preemptible instances were around 80% cheaper. Be aware that the cost of the Compute Engine instances isn't the only component of the cost of a Dataproc cluster, but it's a significant one.

Once your data is in a cluster, you can use Spark and Spark SQL to do data mining, and you can use MLlib, which is Apache Spark's Machine Learning Libraries, to discover patterns through machine learning.

Dataflow offers managed data pipelines

- Processes data using Compute Engine instances.
  - Clusters are sized for you.
  - Automated scaling, no instance provisioning required.
- Write code once and get batch and streaming.
  - Transform-based programming model.

Dataproc is great when you have a dataset of known size, or when you want to manage your cluster size yourself. But what if your data shows up in realtime? Or it's of unpredictable size or rate? That's where Dataflow is a particularly good choice. It's both a unified programming model and a managed service, and it lets you develop and execute a big range of data processing patterns: extract-transform-and-load, batch computation, and continuous computation. You use Dataflow to build data pipelines, and the same pipelines work for both batch and streaming data.

Dataflow is a unified programming model and a managed service for developing and executing a wide range of data processing patterns including ETL, batch computation, and continuous computation. Dataflow frees you from operational tasks like resource management and performance optimization.

Dataflow features:

Resource Management: Dataflow fully automates management of required processing resources. No more spinning up instances by hand.

On Demand: All resources are provided on demand, enabling you to scale to meet your business needs. No need to buy reserved compute instances.

Intelligent Work Scheduling: Automated and optimized work partitioning which can dynamically rebalance lagging work. No more chasing down “hot keys” or pre-processing your input data.

Auto Scaling: Horizontal auto scaling of worker resources to meet optimum throughput requirements results in better overall price-to-performance.

Unified Programming Model: The Dataflow API enables you to express MapReduce like operations, powerful data windowing, and fine grained correctness control regardless of data source.

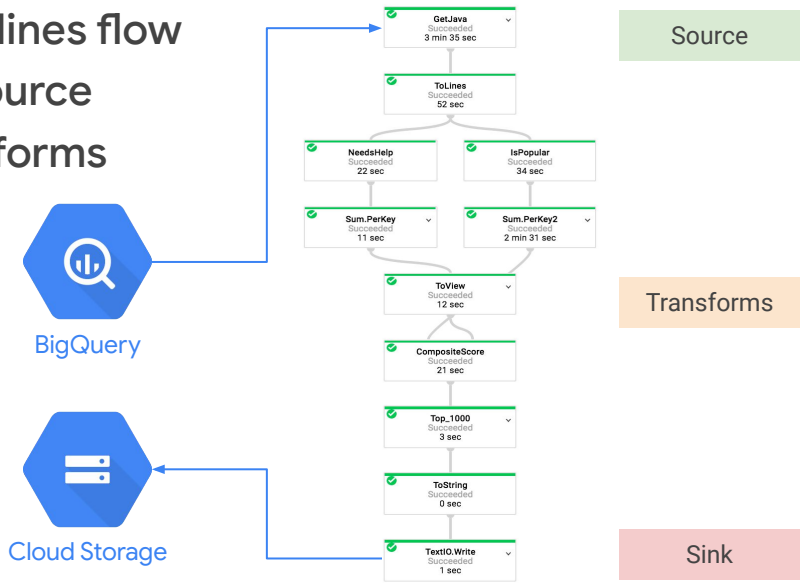
Open Source: Developers wishing to extend the Dataflow programming model can fork and or submit pull requests on the Java-based Dataflow SDK. Dataflow pipelines can also run on alternate runtimes like Spark and Flink.

Monitoring: Integrated into the Cloud Console, Dataflow provides statistics such as pipeline throughput and lag, as well as consolidated worker log inspection—all in near-real time.

Integrated: Integrates with Cloud Storage, Pub/Sub, Datastore, Cloud Bigtable, and BigQuery for seamless data processing. And can be extended to interact with others sources and sinks like Apache Kafka and HDFS.

Reliable & Consistent Processing: Dataflow provides built-in support for fault-tolerant execution that is consistent and correct regardless of data size, cluster size, processing pattern or pipeline complexity.

Dataflow pipelines flow data from a source through transforms to a sink



This example Dataflow pipeline reads data from a BigQuery table (the “source”), processes it in various ways (the “transforms”), and writes its output to Cloud Storage (the “sink”). Some of those transforms you see here are map operations, and some are reduce operations. You can build really expressive pipelines.

Each step in the pipeline is elastically scaled. There is no need to launch and manage a cluster. Instead, the service provides all resources on demand. It has automated and optimized work partitioning built in, which can dynamically rebalance lagging work. That reduces the need to worry about “hot keys” -- that is, situations where disproportionately large chunks of your input get mapped to the same custer.

When should we use Dataflow?

- *ETL* (extract/transform/load) pipelines to move, filter, enrich, shape data.
- *Data analysis*: batch computation or continuous computation using streaming.
- *Orchestration*: create pipelines that coordinate services, including external services.
- Integrates with Google Cloud services like Cloud Storage, Pub/Sub, BigQuery, and Cloud Bigtable.
  - Open source Java and Python SDKs.

People use Dataflow in a variety of use cases. For one, it serves well as a general-purpose ETL tool.

And its use case as a data analysis engine comes in handy in things like these: fraud detection in financial services; IoT analytics in manufacturing, healthcare, and logistics; and clickstream, Point-of-Sale, and segmentation analysis in retail.

And, because those pipelines we saw can orchestrate multiple services, even external services, it can be used in real time applications such as personalizing gaming user experiences.

BigQuery is a fully managed data warehouse and SQL query engine

- Provides near real-time interactive analysis of massive datasets (hundreds of TBs).
- Query using SQL syntax (SQL 2011).
- No cluster maintenance is required.

If, instead of a dynamic pipeline, you want to do ad-hoc SQL queries on a massive dataset, that is what BigQuery is for. BigQuery is Google's fully managed, petabyte scale, low cost analytics data warehouse.

BigQuery is Google's fully managed, petabyte scale, low cost analytics data warehouse. BigQuery is NoOps: there is no infrastructure to manage and you don't need a database administrator, so you can focus on analyzing data to find meaningful insights, use familiar SQL, and take advantage of our pay-as-you-go model. BigQuery is a powerful big data analytics platform used by all types of organizations, from startups to Fortune 500 companies.

BigQuery's features:

Flexible Data Ingestion: Load your data from Cloud Storage or Datastore, or stream it into BigQuery at 100,000 rows per second to enable real-time analysis of your data.

Global Availability: You have the option to store your BigQuery data in European locations while continuing to benefit from a fully managed service, now with the option of geographic data control, without low-level cluster maintenance.

Security and Permissions: You have full control over who has access to the data stored in BigQuery. If you share datasets, doing so will not impact your cost or performance; those you share with pay for their own queries.

Cost Controls: BigQuery provides cost control mechanisms that enable you to cap your daily costs at an amount that you choose. For more information, see Cost Controls.

Highly Available: Transparent data replication in multiple geographies means that your data is available and durable even in the case of extreme failure modes.

Super Fast Performance: Run super-fast SQL queries against multiple terabytes of data in seconds, using the processing power of Google's infrastructure.

Fully Integrated In addition to SQL queries, you can easily read and write data in BigQuery via Dataflow, Spark, and Hadoop.

Connect with Google Products: You can automatically export your data from Google Analytics Premium into BigQuery and analyze datasets stored in Google Cloud Storage, Google Drive, and Google Sheets.

BigQuery can make Create, Replace, Update, and Delete changes to databases, subject to some limitations and with certain known issues.

BigQuery runs on Google's high-performance infrastructure

- Compute and storage are separated with a terabit network in between.
- You only pay for storage and processing used.
- Automatic discount for long-term data storage.

It's easy to get data into BigQuery. You can load from Cloud Storage or Datastore, or stream it into BigQuery at up to 100,000 rows per second.

BigQuery is used by all types of organizations, from startups to Fortune 500 companies. Smaller organizations like BigQuery's free monthly quotas. Bigger organizations like its seamless scale and its available 99.9% service level agreement.

Long term storage pricing is an automatic discount for data residing in BigQuery for extended periods of time. When the age of your data reaches 90 days in BigQuery, Google will automatically drop the price of storage from $0.02 per GB per month down to $0.01 per GB per month.

For more information on the architecture of BigQuery, see:
https://cloud.google.com/blog/big-data/2016/01/bigquery-under-the-hood

Pub/Sub is scalable, reliable messaging

- Supports many-to-many asynchronous messaging.
  - Application components make push/pull subscriptions to topics.
- Includes support for offline consumers.
- Based on proven Google technologies.
- Integrates with Dataflow for data processing pipelines.

Pub/Sub is a fully managed real-time messaging service that allows you to send and receive messages between independent applications. You can leverage Pub/Sub's flexibility to decouple systems and components hosted on Google Cloud or elsewhere on the internet. By building on the same technology Google uses, Pub/Sub is designed to provide "at least once" delivery at low latency with on-demand scalability to 1 million messages per second (and beyond).

Pub/Sub features:

Highly Scalable
Any customer can send up to 10,000 messages per second, by default—and millions per second and beyond, upon request.

Push and Pull Delivery
Subscribers have flexible delivery options, whether they are accessible from the internet or behind a firewall.

Encryption
Encryption of all message data on the wire and at rest provides data security and protection.

Replicated Storage
Designed to provide "at least once" message delivery by storing every message on multiple servers in multiple zones.

Message Queue
Build a highly scalable queue of messages using a single topic and subscription to support a one-to-one communication pattern.

End-to-End Acknowledgement
Building reliable applications is easier with explicit application-level acknowledgements.

Fan-out
Publish messages to a topic once, and multiple subscribers receive copies to support one-to-many or many-to-many communication patterns.

REST API
Simple, stateless interface using JSON messages with API libraries in many programming languages.

When should we use Pub/Sub?

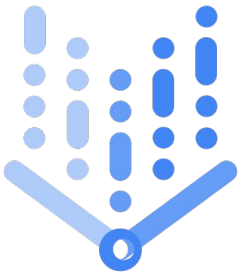
- Building block for data ingestion in Dataflow, Internet of Things (IoT), Marketing Analytics.
- Foundation for Dataflow streaming.
- Push notifications for cloud-based applications.
- Connect applications across Google Cloud (push/pull between Compute Engine and App Engine).

Pub/Sub builds on the same technology Google uses internally. It's an important building block for applications where data arrives at high and unpredictable rates, like Internet of Things systems. If you're analyzing streaming data, Dataflow is a natural pairing with Pub/Sub.

Pub/Sub also works well with applications built on Google Cloud's compute platforms. You can configure your subscribers to receive messages on a "push" or a "pull" basis. In other words, subscribers can get notified when new messages arrive for them, or they can check for new messages at intervals.

Vertex AI Workbench is a notebook service to get your projects up and running in minutes

- Managed JupyterLab experience.
- Secure development and controlled user access.
- Advanced networking.
- Support for data science frameworks and optimized for machine learning.
- Git support.
- Bring your own container

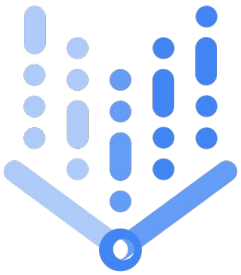


Pub/Sub builds on the same technology Google uses internally. It's an important building block for applications where data arrives at high and unpredictable rates, like Internet of Things systems. If you're analyzing streaming data, Dataflow is a natural pairing with Pub/Sub.

Pub/Sub also works well with applications built on Google Cloud's compute platforms. You can configure your subscribers to receive messages on a "push" or a "pull" basis. In other words, subscribers can get notified when new messages arrive for them, or they can check for new messages at intervals.

When should we use Vertex AI Workbench?

- Get up and running fast. Deploy new JupyterLab instances with one click.
- Instances are preconfigured with optimized versions of popular data science and ML libraries.
- Scale on demand.
- Seamless experience.
- Can schedule runs used Scheduled Executions
- Can run as part of Vertex AI Pipelines



Pub/Sub builds on the same technology Google uses internally. It's an important building block for applications where data arrives at high and unpredictable rates, like Internet of Things systems. If you're analyzing streaming data, Dataflow is a natural pairing with Pub/Sub.

Pub/Sub also works well with applications built on Google Cloud's compute platforms. You can configure your subscribers to receive messages on a "push" or a "pull" basis. In other words, subscribers can get notified when new messages arrive for them, or they can check for new messages at intervals.

Cloud Composer is a fully managed workflow orchestration service built on Apache Airflow

- Hybrid and multi-cloud
- Open source
- Easy orchestration in Python
- Built-in integration with BigQuery, Dataflow, Dataproc, Cloud Storage, Pub/Sub, and more
- Allows for branching DAGs to handle task failures & other conditionals
- Requires a persistent cluster

Pub/Sub builds on the same technology Google uses internally. It's an important building block for applications where data arrives at high and unpredictable rates, like Internet of Things systems. If you're analyzing streaming data, Dataflow is a natural pairing with Pub/Sub.

Pub/Sub also works well with applications built on Google Cloud's compute platforms. You can configure your subscribers to receive messages on a "push" or a "pull" basis. In other words, subscribers can get notified when new messages arrive for them, or they can check for new messages at intervals.

When should we use Cloud Composer?

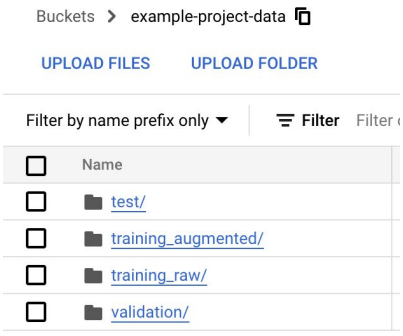
- When we have lots of small orchestration tasks to run using Python, for example loading data from APIs.
- Don't use for a one-off project. Only use if the persistent cluster will be used frequently enough to be worthwhile

Pub/Sub builds on the same technology Google uses internally. It's an important building block for applications where data arrives at high and unpredictable rates, like Internet of Things systems. If you're analyzing streaming data, Dataflow is a natural pairing with Pub/Sub.

Pub/Sub also works well with applications built on Google Cloud's compute platforms. You can configure your subscribers to receive messages on a "push" or a "pull" basis. In other words, subscribers can get notified when new messages arrive for them, or they can check for new messages at intervals.

Organize Unstructured Data

When using unstructured data (images, audio, video, etc.) stored in Cloud Storage, design a directory structure to keep your data splits and any processing done on them clear and cleanly separated.



When using unstructured data (images, audio, video, etc.) stored in Cloud Storage, design a directory structure to keep your data splits and any processing done on them clear and cleanly separated.

More ideas at [ML Best Practices > Machine Learning Development]

Masking Sensitive Data

GCP offers tools to intelligently de-identify sensitive data in BigQuery or Cloud Storage using a number of techniques:

- Redaction: Deletes all or part of a detected sensitive value.
- Replacement: Replaces a detected sensitive value with a specified surrogate value.
- Masking: Replaces a number of characters of a sensitive value with a specified surrogate character, such as a hash (#) or asterisk (*).
- Crypto-based tokenization: Encrypts the original sensitive data value using a cryptographic key. Sensitive Data Protection supports several types of tokenization, including transformations that can be reversed, or "re-identified."
- Bucketing: "Generalizes" a sensitive value by replacing it with a range of values. (For example, replacing a specific age with an age range, or temperatures with ranges corresponding to "Hot," "Medium," and "Cold.")
- Date shifting: Shifts sensitive date values by a random amount of time.
- Time extraction: Extracts or preserves specified portions of date and time values.

De-Identifying data does require retraining a model based on it from scratch.
Transformations reference: https://cloud.google.com/dlp/docs/transformations-reference

You can also create inspection jobs to scan for sensitive data or re-identification potential:
https://cloud.google.com/dlp/docs/concepts-job-triggers

More ideas at ML Best Practices > Machine Learning Development

We split data into training, validation, and test sets so that at different phases of the project, we can estimate how the model will perform on new data.

Question to the class:
Why not just training and test? How would you describe what the validation set does for us?

Test Set Validity

Make sure that your test set meets the following two conditions:

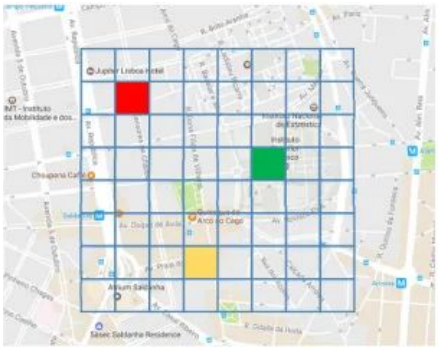
- It is large enough to yield statistically meaningful results.
- It is representative of the data set as a whole. In other words, don't pick a test set with different characteristics than the training set.

From ML Crash Course > Training and Test Sets

Strategies for Splitting Train/Val/Test Data

How can we split data so that as more data is added, recurring or dependant examples do not later appear in other splits?

For example, when planning a data split for detecting objects in satellite imagery, we may assign rounded GPS coordinates to appear only in training, validation, or test.



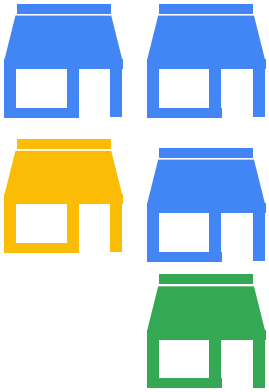
How can we split data such that recurring or dependant examples from training do not later reappear in validation or test?
When detecting instances in satellite or city imagery, for example, we may split on rounded GPS coordinates in random ‘stripes’ across the city.
For example if you have a model that detects potholes from satellite imagery, you wouldn’t want the same pothole from different satellite passes to appear in training multiple times and then again in validation or test. By assigning data split by GPS coordinates (possibly rounded), you can ensure the same examples don’t appear across the split.

Relevant resources & image source:
https://cloud.google.com/vertex-ai/docs/general/ml-use
https://cloud.google.com/vertex-ai/docs/tabular-data/data-splits

Strategies for Splitting Train/Val/Test Data

How can our test set best represent data on which we would like to predict, but haven't seen?

For example, if we are using some stores' data as training data for a larger chain of stores, we could use 80% of our available stores in training, and put 10% of our stores in validation and 10% in test to understand if some stores (our training data) will be predictive of unseen stores (our validation and test stores).



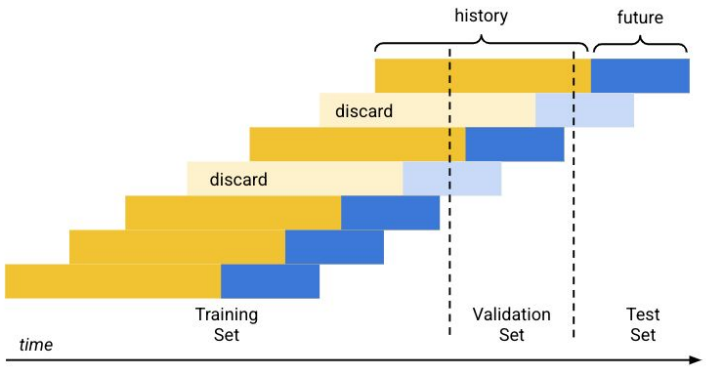
How can our test set best represent data on which we would like to predict, but haven't seen?
For example, if we are using some stores' data as training data for a larger chain of stores, we could use 80% of our available stores in training, and put 10% of our stores in validation and 10% in test to understand if some stores (our training data) will be predictive of unseen stores (our validation and test stores). *We could compare this to a model where past data is predictive of future results within one store to see if we should be training on data specific to each store.*

Relevant resources & image source:
https://cloud.google.com/vertex-ai/docs/general/ml-use
https://cloud.google.com/vertex-ai/docs/tabular-data/data-splits

Strategies for Splitting Train/Val/Test Data

Which data should be predictive of which other data?

In time series, past data should be predictive of the future.



What data should be predictive of what other data?
In time series, past data should be predictive of the future.
(not a random sampling across the full timeline being predictive of gaps that need to be filled in.)

Relevant resources & image source:
https://cloud.google.com/vertex-ai/docs/general/ml-use
https://cloud.google.com/vertex-ai/docs/tabular-data/data-splits

Handling Imbalanced Data

For Imbalanced Data:

Explanation:
Downsampling and upsampling are techniques used to adjust the class distribution of a dataset.
They're especially crucial when dealing with imbalanced datasets, where one class significantly outnumbers the other.

1. Downsampling (or Undersampling)
Downsampling involves randomly removing instances from the over-represented class to balance the class distribution.

Advantages:
Reduces the size of the dataset, which can lead to faster training times.

Disadvantages:
Information loss since instances from the over-represented class are removed.
Can lead to underfitting if not done correctly.

Business Use cases:

1) Bank Fraud Detection: Say out of 1,000,000 transactions, only 500 are

1) fraudulent. Training a model on such data will most likely predict everything as non-fraudulent because of the huge class imbalance. To avoid this, one might downsample the non-fraudulent transactions so that the model can learn a better decision boundary between fraudulent and non-fraudulent transactions.

2) Predictive Maintenance: In industries, machine failures might be a rare event. If you're trying to predict these rare failures, you might have a huge amount of 'machine working fine' data and very few 'machine failure' data. Downsampling the 'working fine' instances can help create a balanced dataset for modeling.

2. Upsampling (or Oversampling)
Upsampling involves adding more instances to the under-represented class. This can be achieved either by duplicating instances or by generating synthetic instances.

Advantages:
No loss of information.
Can lead to a better representation of the minority class.

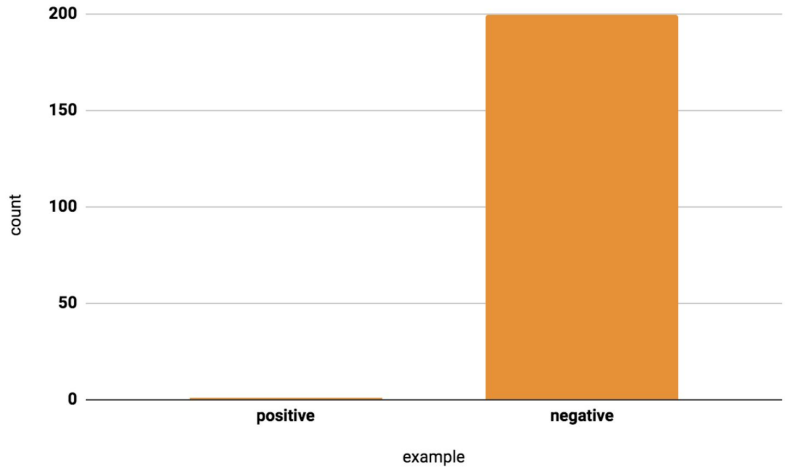
Disadvantages:
Can increase the size of the dataset, leading to longer training times.
Risk of overfitting, especially if synthetic data is not representative of true instances.
Real-world business cases:

Business Use cases:

1) Medical Diagnostics: Consider a situation where you're diagnosing a rare disease. The number of positive cases might be much smaller than the negative ones. Using upsampling, you can increase the number of positive cases (either by duplication or synthetic data generation) to balance the dataset. This ensures that the model has enough data to learn about the characteristics of the rare disease.

2) Churn Prediction: If a business has a very low churn rate (which is good!), it might result in an imbalanced dataset where the 'churn' instances are far fewer than 'non-churn'. In such cases, upsampling the 'churn' instances can help the model understand the patterns leading to customer churn more effectively.

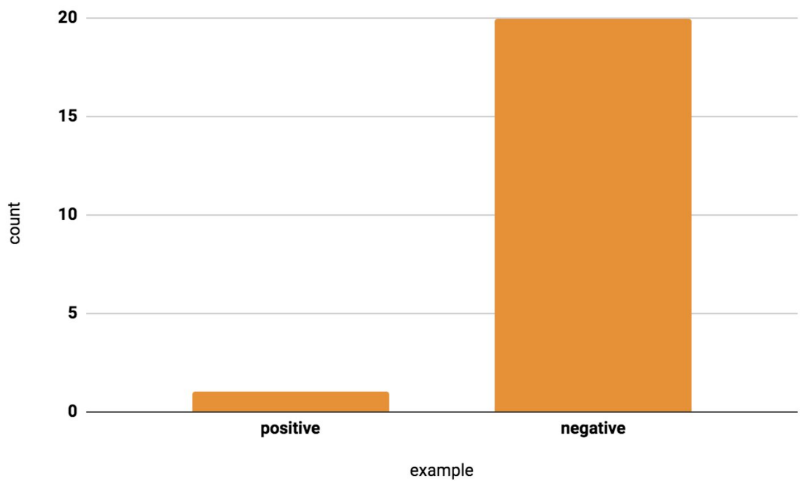
Handling Imbalanced Data



For Imbalanced Data:

Consider the following example of a model that detects fraud. Instances of fraud happen once per 200 transactions in this data set, so in the true distribution, about 0.5% of the data is positive.

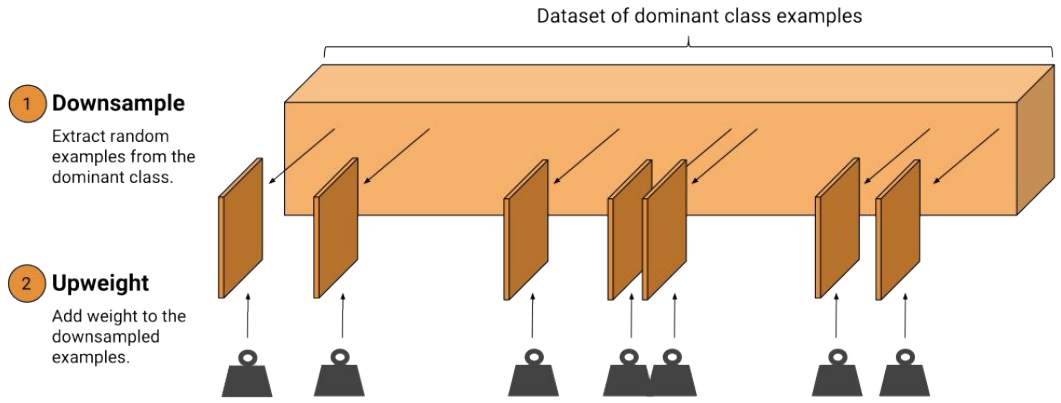
Handling Imbalanced Data



For Imbalanced Data:

Step 1: Downsample the majority class. Consider again our example of the fraud data set, with 1 positive to 200 negatives. Downsampling by a factor of 10 improves the balance to 1 positive to 20 negatives (5%). Although the resulting training set is still *moderately imbalanced*, the proportion of positives to negatives is much better than the original *extremely imbalanced* proportion (0.5%).

Handling Imbalanced Data



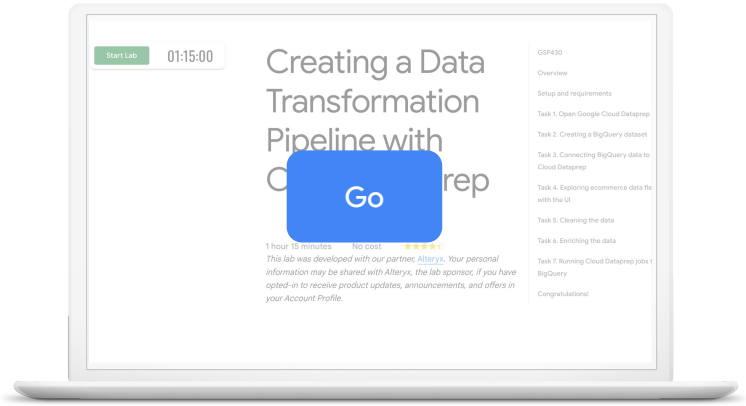
For Imbalanced Data:

Step 2: Upweight the downsampled class: The last step is to add example weights to the downsampled class. Since we downsampled by a factor of 10, the example weight should be 10.

An example of assigning class weights is here: https://www.tensorflow.org/tutorials/structured_data/imbalanced_data#train_a_model_with_class_weights

Recommended Lab

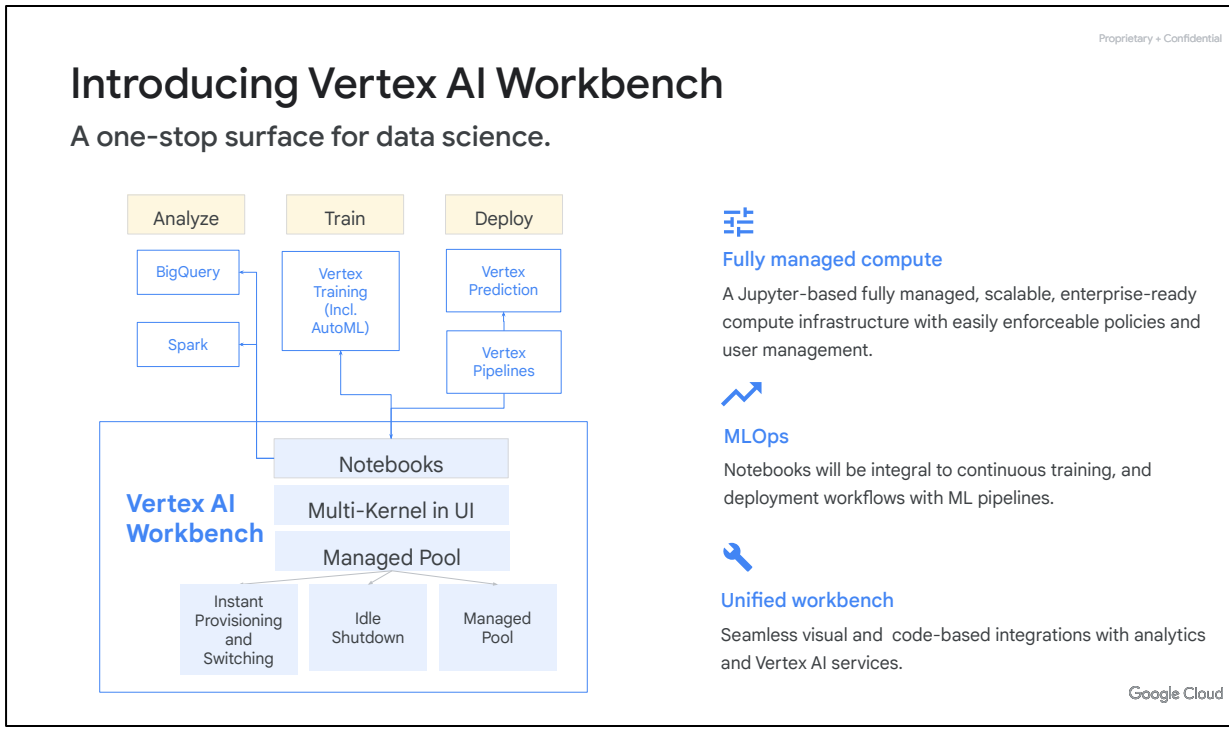
Creating a Data Transformation Pipeline with Cloud Dataprep



From the Course
Transform and Clean your Data with Dataprep by Alteryx on Google Cloud

Working with Cloud Dataprep on Google Cloud
(part of Transform and Clean your Data with Dataprep by Alteryx on Google Cloud)

Model prototyping using Jupyter notebooks



Vertex AI Workbench is a single development environment for the entire data science workflow.

You can use Vertex AI Workbench's notebook-based environment to query and explore data, develop and train a model, and run your code as part of a pipeline.

For example, Vertex AI Workbench lets you:

- Access and explore your data from within a Jupyter notebook by using BigQuery and Cloud Storage integrations.
- Automate recurring updates to your model by using scheduled executions of your notebook's code that run on Vertex AI.
- Process data quickly by running a notebook on a Dataproc cluster.
- Run a notebook as a step in a pipeline by using Vertex AI Pipelines.

Vertex AI Workbench offers a managed notebooks option with built-in integrations that help you to set up an end-to-end notebook-based production environment. For users who need full control over their environment, Vertex AI Workbench provides a user-managed notebooks option.

Vertex AI Workbench instances are prepackaged with JupyterLab and have a

preinstalled suite of deep learning packages, including support for the TensorFlow and PyTorch frameworks. You can configure either CPU-only or GPU-enabled instances.

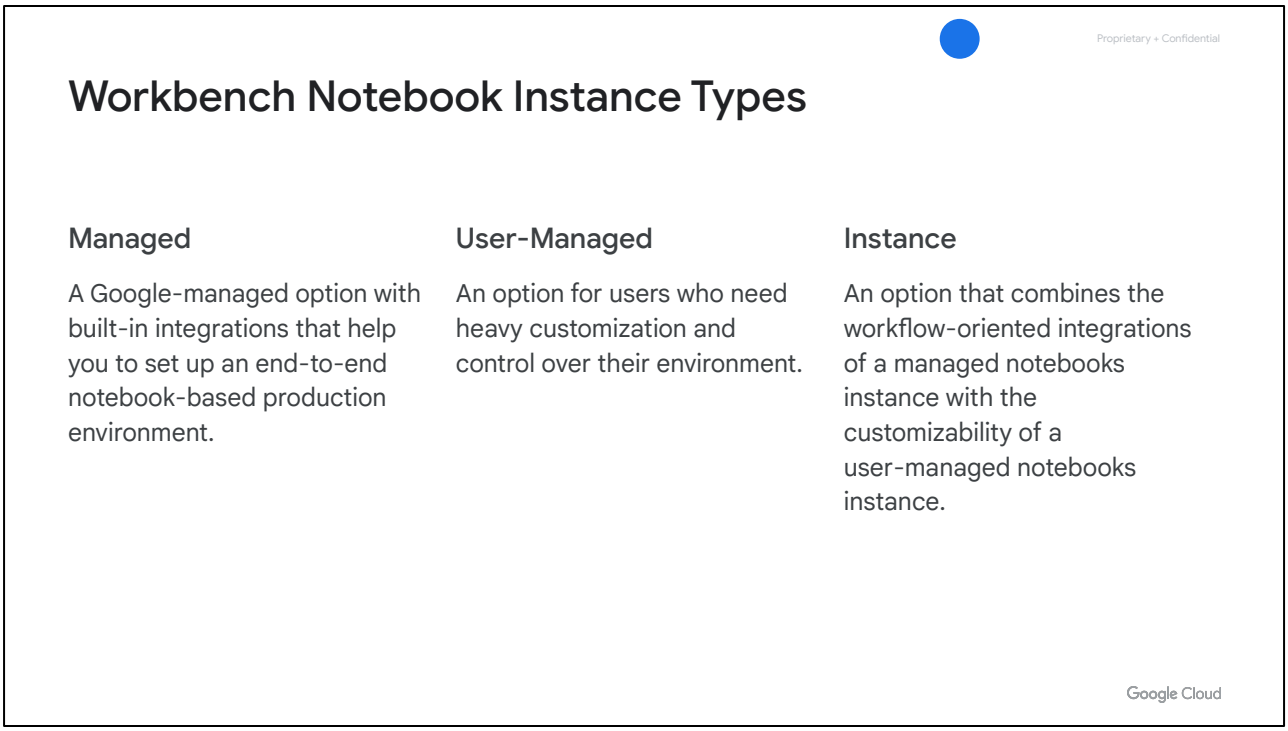
Vertex AI Workbench instances support the ability to sync with a GitHub repository. Vertex AI Workbench instances are protected by Google Cloud authentication and authorization.

Your Vertex AI Workbench notebook instances are protected by Google Cloud authentication and authorization.

You can also create notebooks with Spark kernels using Dataproc Serverless Spark.

Source: MLOps Architectures with CI/CD (slides) | Y22
https://docs.google.com/presentation/d/1OiGZOviGJf6XX7E91-41PGBy3aoRPZcibmF9dtH1Cms/edit?resourcekey=0-0LF8xS6di73GKb_vAVDZjw#slide=id.g1043fd8a725_0_2031

Source:
https://cloud.google.com/vertex-ai/docs/workbench/instances/create



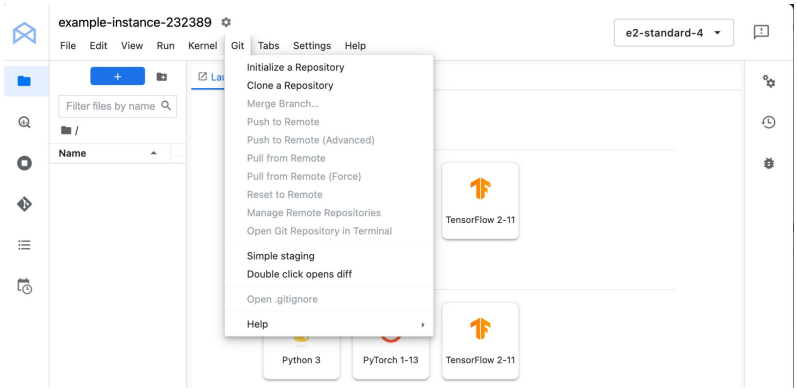
- Vertex AI Workbench instances: An option that combines the workflow-oriented integrations of a managed notebooks instance with the customizability of a user-managed notebooks instance.
- Vertex AI Workbench managed notebooks: A Google-managed option with built-in integrations that help you to set up an end-to-end notebook-based production environment.
- Vertex AI Workbench user-managed notebooks: An option for users who need heavy customization and control over their environment.

All instance types are prepackaged with JupyterLab and have a preinstalled suite of deep learning packages, including support for the TensorFlow and PyTorch frameworks. You can use CPU-only or GPU-enabled instances. All instance types also integrate with GitHub so that you can sync your notebook with a GitHub repository.

All Vertex AI Workbench instance types are protected by Google Cloud authentication and authorization.

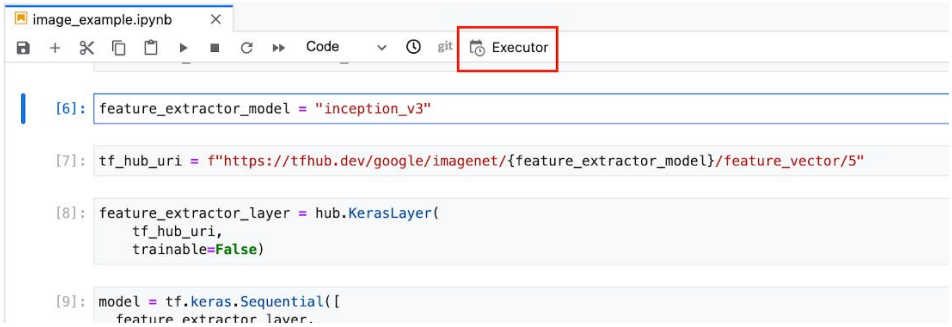
From: https://cloud.google.com/vertex-ai/docs/workbench/introduction
More info: https://cloud.google.com/vertex-ai/docs/workbench/notebook-solution

Sync Notebooks with GitHub



Check in your notebook environment to source control to collaborate with others and build tools each other can use.

Schedule Notebook Runs



You can schedule notebooks to run.

You can also run a notebook as a step in a pipeline by using Vertex AI Pipelines.

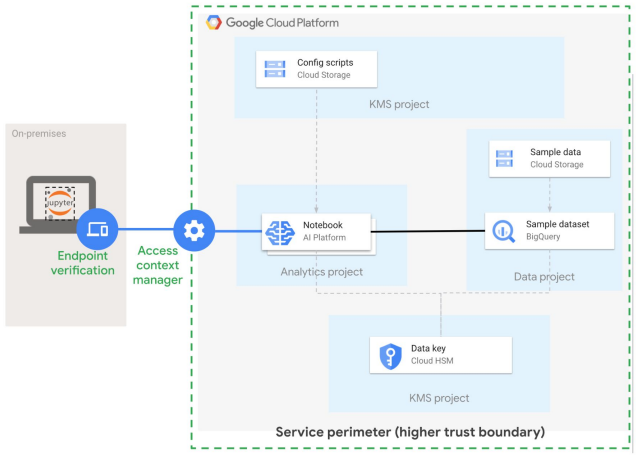
Workbench Notebook Roles

Creating notebooks requires roles outside of Vertex AI Administrator. You'll need:

- Notebooks Administrator
- Service Account User (on the project or Compute Engine service account)

Proprietary + Confidential

Protecting confidential data in Vertex AI Workbench user-managed notebooks



Google Cloud

You can use a VPC (virtual private cloud) to restrict access to certain users using Identity-Aware Proxy (IAP) and Endpoint Verification to ensure only approved devices may access your GCP resources & data.

Security Pattern: [Protecting confidential data in Vertex AI Workbench user-managed notebooks]

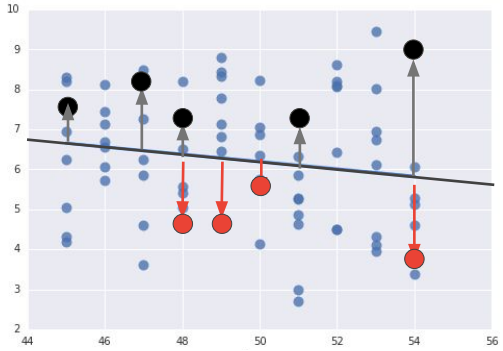
Understanding Model Metrics

We evaluate our models with a loss function that calculates our error for each example and aggregates those 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 together?



One measure of the quality of the prediction at a single point in your data set is simply the signed difference between the prediction and the actual value, or *label*. This difference is called the error.

How might you put a bunch of error values together? The simplest way to compose them is a SUM.

If you were to use the sum function to compose your error terms, the resulting model would treat error terms of opposite sign as cancelling each other out. And while your model *does* need to cope with contradictory evidence, it's not the case that a model that splits the difference between positive and negative errors has found a perfect solution.

You'd like to reserve the "perfect" designation for a model in which the predictions match the label for all points in your dataset, not for a model that makes signed errors that cancel each other out.

Performance Metrics for a Regression Model

MAE (Mean Absolute Error)
Average of the absolute difference between the actual and predicted values in the dataset.

MSE (Mean Squared Error)
Average of the squared difference between the original and predicted values in the data set. Squaring increases the penalty of larger errors.

RMSE (Root Mean Squared Error)
Square root of the MSE. This returns the error to the same units as the dependent variable.

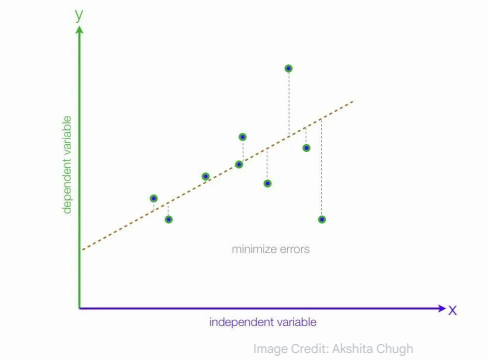


Image from this Medium post:
https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e

In statistics, mean absolute error (MAE) is a measure of errors between paired observations expressing the same phenomenon. Examples of Y versus X include comparisons of predicted versus observed, subsequent time versus initial time, and one technique of measurement versus an alternative technique of measurement. MAE is calculated as the sum of absolute errors divided by the sample size

Alternative formulations may include relative frequencies as weight factors. The mean absolute error uses the same scale as the data being measured. This is known as a scale-dependent accuracy measure and therefore cannot be used to make comparisons between predicted values that use different scales.[2] The mean absolute error is a common measure of forecast error in time series analysis,[3] sometimes used in confusion with the more standard definition of mean absolute deviation. The same confusion exists more generally.

Let's calculate 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
```

4. Take the square root of the mean.

```
1.58
```

RMSE (Root Mean Squared Error)
Square root of the MSE. This returns the error to the same units as the dependent variable while increasing the influence of further outliers by squaring.

The sum of the absolute values of the errors seems like a reasonable alternative, but there are problems with this method of composing data as well, which will be tackled shortly.
Instead, what is often used is what is called the Mean Squared Error. The MSE is computed by taking the set of errors from your dataset, taking their squares (to get rid of the negatives), and computing the average of the squares.

The MSE is a perfectly valid loss function, but it has one problem. Although errors might be in pounds, or kilometers, or dollars, the square error will be pounds-squared, kilometers-squared, or dollars-squared. That can make the MSE hard to interpret.
As a result, it is common practice to take the square-root of that mean squared error to get to units that can be understood. RMSE is the root of the mean squared error.

The bigger the RMSE, the worse the quality of the predictions. So, what you want to do is to minimize the RMSE.
The notation here is to use a little hat symbol on top of the Y that represents

your model's prediction, and to use a plain Y to represent the label.

Colab RMSE:
https://colab.research.google.com/github/datacoe-publicissapient/risingai2020/blob/master/notebooks/RMSE_MAPE_Forecasting.ipynb?authuser=0&pli=1

https://www.codecogs.com/eqnedit.php?latex=sqrt{\frac{1}{n}&space;\times&space;\sum_{i=1}^{n}(\hat{Y}_{i}-Y_{i})^{2}}

Lower RMSE indicates a better performing model

RMSE=.145 **RMSE=.149**

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

Now you have a metric to compare two points in parameter-space--two sets of parameter values for your linear model--formally.

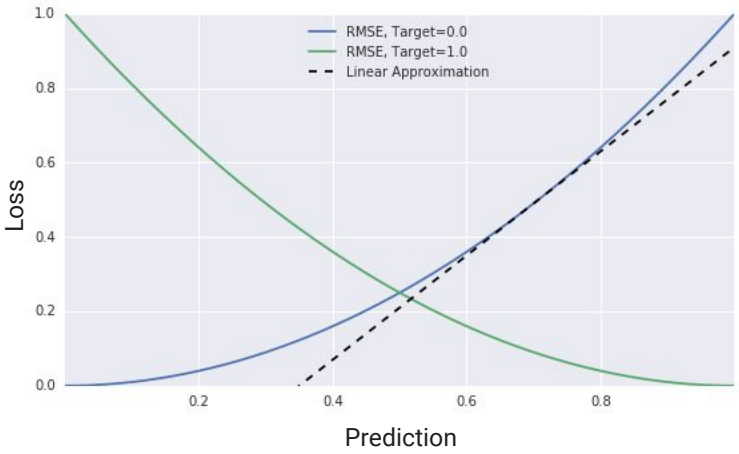
Take a look at these two scatterplots and regression lines for Baby Weight vs Mother's Age for mothers above 39. It can be incredibly hard to visually spot which line is a better fit to the underlying data. That's where your loss metrics aid in deciding which model is better.

The model on the left has an RMSE of .145, and the model on the right has an RMSE of .149. Therefore, the loss function indicates that the values for weight and bias on the left-hand side are better than on the right-hand side.

Colab RMSE:
https://colab.research.google.com/github/datacoe-publicissapient/risingai2020/blob/master/notebooks/RMSE_MAPE_Forecasting.ipynb?authuser=0&pli=1

Problem: RMSE doesn't work for classification

RMSE doesn't penalize bad classifications appropriately.



Although RMSE works fine for linear regression problems, it doesn't work as a loss function for classification. Remember, classification problems are ones in which the label is a categorical variable. The problem with using RMSE for classification has to do with how these categorical variables are represented in your model. As discussed, categorical variables are often represented as binary integers.

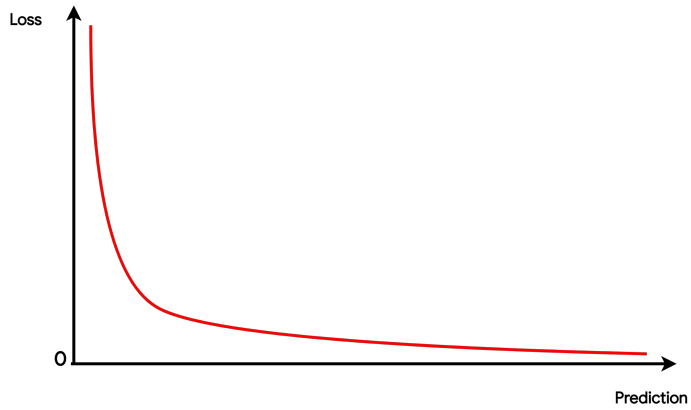
For an intuition as to why this presents a problem, look at the loss curves depicted. The domain, on the X axis, represents the prediction. The range, on the Y axis, represents the loss, given that prediction. Color here denotes the label: green indicates the label was 1, blue indicates the label was 0.

What's wrong with this curve?

The problem is, it fails to capture an intuitive belief that predictions that are really bad should be penalized much more strongly. Note how a prediction of 1 when the target is 0 is about 3 times worse than a prediction of 0.5 for the same target.

Instead of RMSE then, you need a new loss function, one that penalizes in accordance to your intuitions.

Solution: Classification gets its own metrics, like cross entropy



One of the most commonly used loss functions for classification is called Cross-Entropy or Log Loss. Here you have similar graph to what you saw on the last slide, only instead of showing the loss for RMSE, the value of a new loss function, called Cross-Entropy, is shown. Note that unlike RMSE, Cross-Entropy penalizes bad predictions very strongly, even in this limited domain.

Cross Entropy:
Cross-entropy is a measure used in machine learning and statistics to quantify the difference between two probability distributions. In the context of machine learning, it's often used as a loss function to train models like classifiers.

Here's a simple way to understand it:
Imagine you have a set of events, and for each event, there are two things:

- The true probabilities: what actually happens in reality.
- The predicted probabilities: what your model thinks will happen.

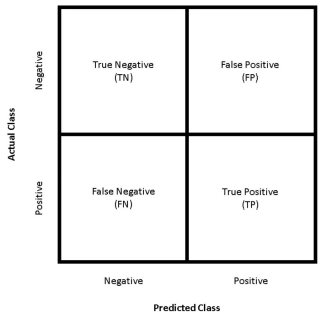
For example, let's say you're trying to predict whether it will rain tomorrow.

The true probability might be 0.8 (80% chance of rain, maybe because it's the middle of the rainy season), and your model predicts a probability of 0.6 (60% chance of rain).

- Cross-entropy is particularly useful in classification tasks because it's sensitive to differences between the predicted probabilities and the actual class labels (which in practice are often "one-hot" encoded, i.e., have probability 1 for the true class and 0 for all other classes). So when you're training a model and you want the predicted probabilities to match the actual class labels as closely as possible, minimizing the cross-entropy helps achieve this.

Colab: Cross-Entropy: https://colab.research.google.com/github/tensorchiefs/dl_book/blob/master/chapter_04/nb_ch04_02.ipynb

Measuring the performance of a Classification Model with a confusion matrix



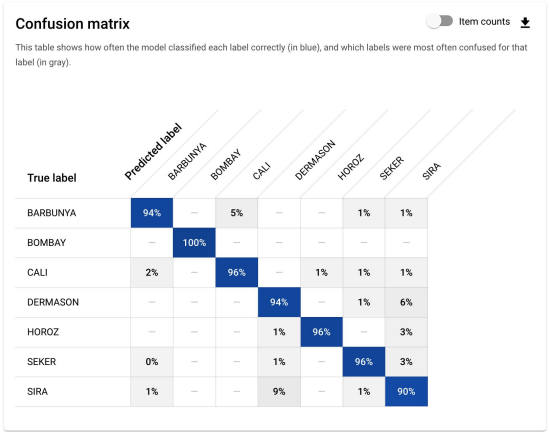
In classification, we predict a class. So don't get a result where we can calculate the difference between it and a ground truth value.

So we often use confusion matrices to understand true positives (correct classifications), false positives (incorrectly guessed to be the class of interest), false negatives (incorrectly missed examples), and true negatives (correctly predicted to not be a member of the class).

Multi-class classification is basically creating an individual prediction for each class, then choosing the class with the highest confidence.

[Replace with screenshot from Vertex AI model.]

Confusion matrices also work for multiclass classification

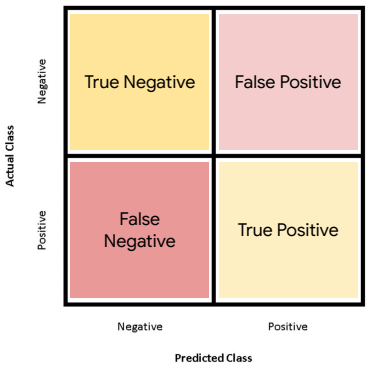


Multiple classes can be compared as well to see which classes are being confused for each other.

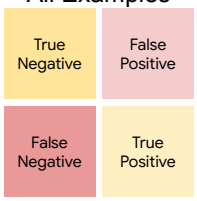
You can see percentages or item counts.

Accuracy

What percentage of our total predictions did we get correct?



$$\frac{\text{True Positive} + \text{True Negative}}{\text{All Examples}}$$

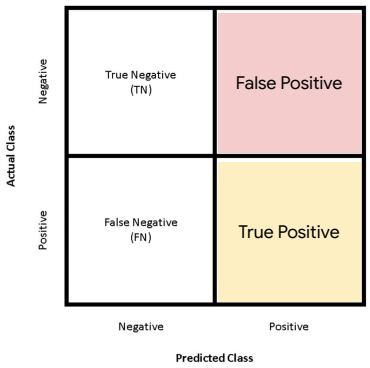


Explain confusion matrix

[Replace with screenshot from Vertex AI model.]

Precision

What percentage of our positive predictions are correct?



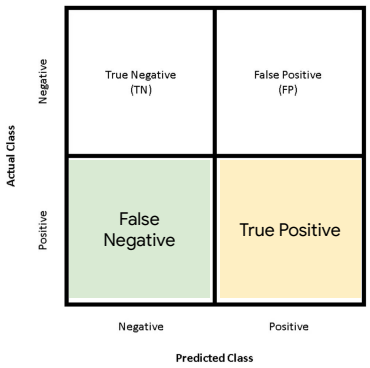
$$\frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

Explain confusion matrix

[Replace with screenshot from Vertex AI model.]

Recall

What percentage of the total positive examples did we find?

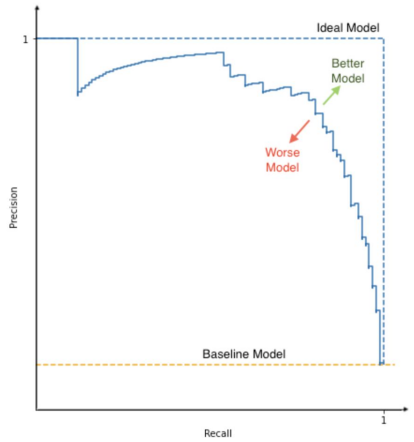


$$\frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

Explain confusion matrix

[Replace with screenshot from Vertex AI model.]

Precision-Recall Curve



Plotting Precision against recall at various thresholds allows you to determine the best tradeoff for your use case.

Try to think in real world examples to weigh the cost of missed examples (if higher, prefer Recall) vs the cost of an incorrect prediction (if higher, prefer Precision).

F1 Score: The Harmonic Mean of Precision & Recall

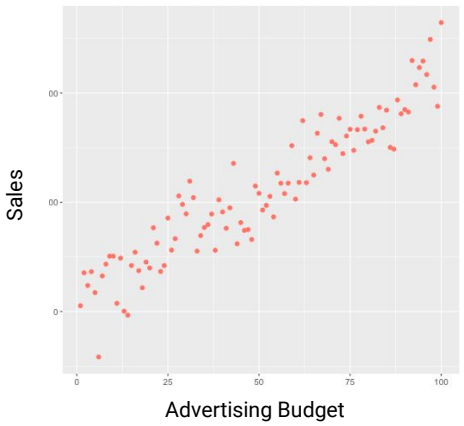
$$\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

If we need to replace considerations of Precision and Recall with a single score, we can use an F1 score.

Can be nice to have one number to compare models regardless of threshold, but less real-world in terms of our understanding of the performance of the model.

Suppose we want to predict sales based on advertising budget

What is the error measure to optimize?



Let's consider a model to predict sales based on advertising budget.

What do you observe about the pattern you see in the data? It looks very strongly correlated (the more budget gained, the longer the higher the sale, which intuitively makes sense).

To model this behavior and prove a correlation, what model do you typically want to call on first? A simple linear regression model

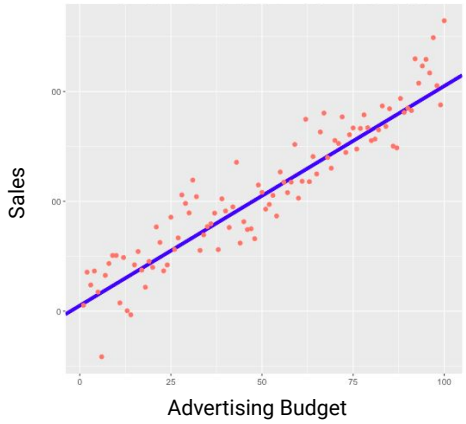
And as we covered, for regression problems, the loss metric you want to optimize is typically Mean Squared Error (MSE) or Root Mean Squared Error (RMSE).

Demo: Advertising budget of a product vs Sales, A real world simple Linear Regression Example: https://colab.research.google.com/drive/19L6GfHnDU4255BnjdP8jH4LrSrHpil90?usp=sharing

Model 1 is a linear model using linear regression

Red = training examples
Blue = trained model

Model 1 RMSE on Training Data: 2.224



The Mean Squared Error tells you how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line (these distances are the “errors”) and squaring them. The squaring is necessary to remove any negative signs. MSE also gives more weight to larger differences.

Taking the square root of the MSE gives us the RMSE, which is simply the distance, on average, of a data point from the fitted line, measured along a vertical line. The RMSE is directly interpretable in terms of measurement units, and so is a better measure of goodness of fit than a correlation coefficient.

For both error measures a lower value indicates a better performing model. The closer the error is to zero, the better.

Here we’re using a Linear Regression model, which simply draws the line of “best fit” to minimize the error. Our final RMSE is 2.224. Let’s say that, for our problem, this is pretty good.

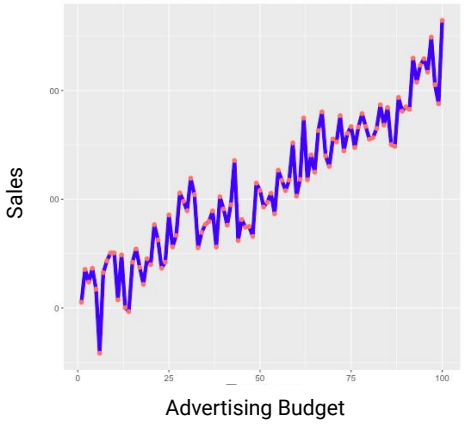
Model 2 has more free parameters

Red = training examples
Blue = trained model

Model 1 RMSE on Training Data: 2.224

Model 2 RMSE on Training Data: 0

But which model is better?



But look at this! What if we use a more complex model? A more complex model has more free parameters. In this case, these free parameters let us capture every squiggle of the dataset. Doing so ...

We reduced our RMSE all the way down to 0 -- the model is now perfectly accurate. Are we done? Is this the best model?

People intuitively feel that there is something fishy about Model 2.

But how can we tell? In ML, we often have lots of data, and no such intuition. Is an NN with 8 nodes better than an NN with 12 nodes? The 12-nodes version has lower RMSE … so should we pick it? But what if we try 16 nodes, and it has even lower RMSE?

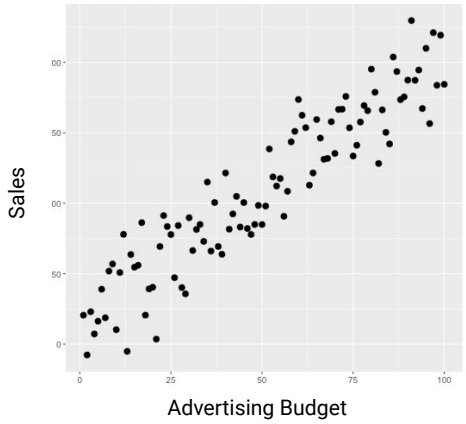
The example here might be a polynomial of 100th order or a neural network with lots of nodes. As you saw in the spiral example at the end of the last lecture on Optimization, a more complex model has more parameters that can be optimized. While this can help it fit more complex data, it might also help it memorize simpler datasets.

At what point do we stop and say a model is now simply memorizing the data and overfitting?

Does the model generalize to new data?

Our meaningful metric is not the one calculated on training data, but the one calculated on data the model has not seen in training.

These black dots indicate new data the model hasn't seen.



One of the best ways to assess the quality of a model is to see how well it performs against a new set of data that it has not seen before. Then we can determine whether the model "generalizes" well across new data points.

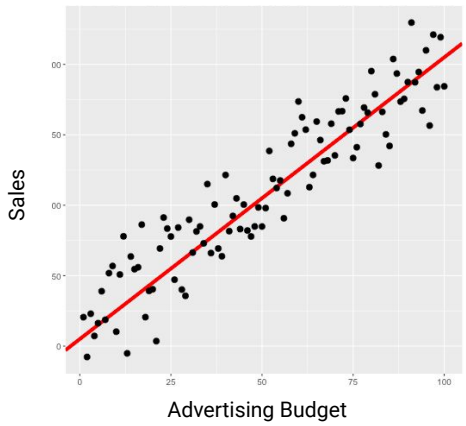
Let's check back on the linear regression model and neural network models and see how they are doing…

Model 1 generalizes well

Model 1 RMSE on Training Data: 2.224
Model 1 RMSE on New Data: 2.198

Pretty similar = good

These black dots indicate new data the model hasn't seen.



Our linear regression model on the new data points is generalizing well. The new Root Mean Squared Error is comparable to what we saw before (no surprises is a good thing; we want consistent performance out of our models).

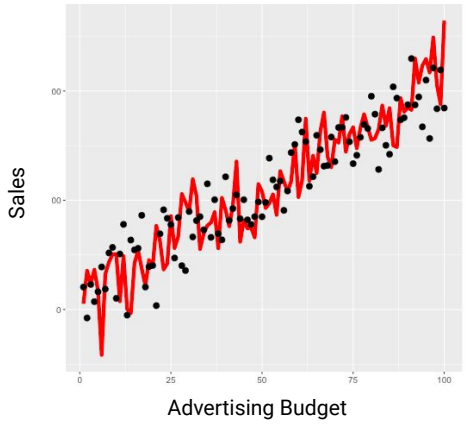
Model 2 does not generalize well

Model 2 RMSE on Training Data: 0

Model 2 RMSE on New Data: 3.2

This is a red flag!

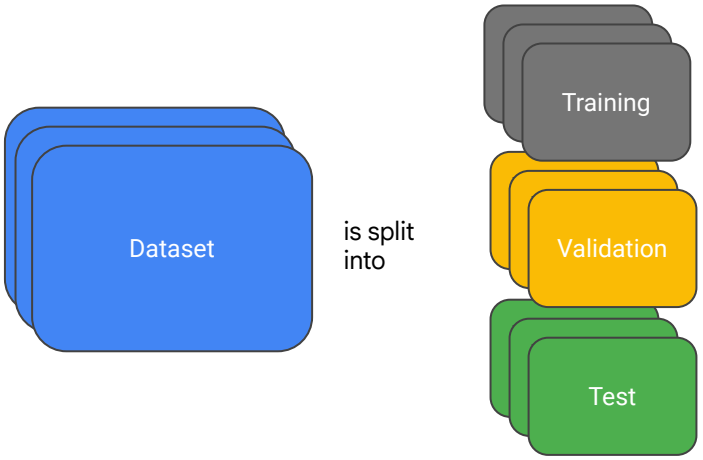
These black dots indicate new data the model hasn't seen.



Looking back at model 2, we can see that it does not generalize well at all on the new training dataset. The RMSE jumped from 0 (perfect accuracy) to 3.2, which is a big problem.

The model was coampletely overfitting on the training dataset it was provided, and that proved to be too brittle (not generalizable) when new data was provided.

Split the dataset



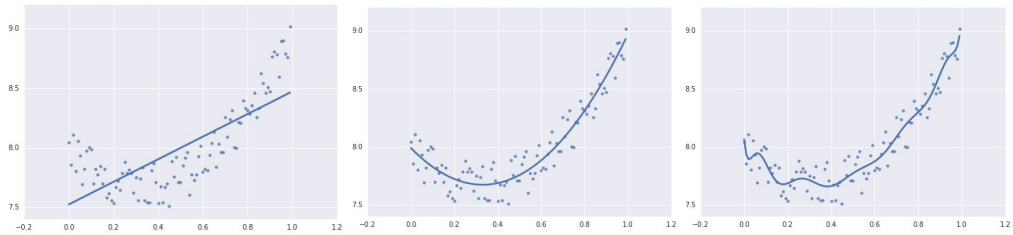
Now you may be asking, how can I be sure that my model is not overfitting? How do I know when to stop training?

The answer is simple: Split your data!

By dividing your original dataset into completely separate and isolated groups, you can iteratively train your model on your training dataset and then compare its performance against an independent validation dataset.

Models that generalize well have similar error values across training and validation. As soon as you start seeing your models not perform well against your validation data (for example, if your loss metrics start to increase), it's time to stop.

Beware of overfitting as you increase model complexity



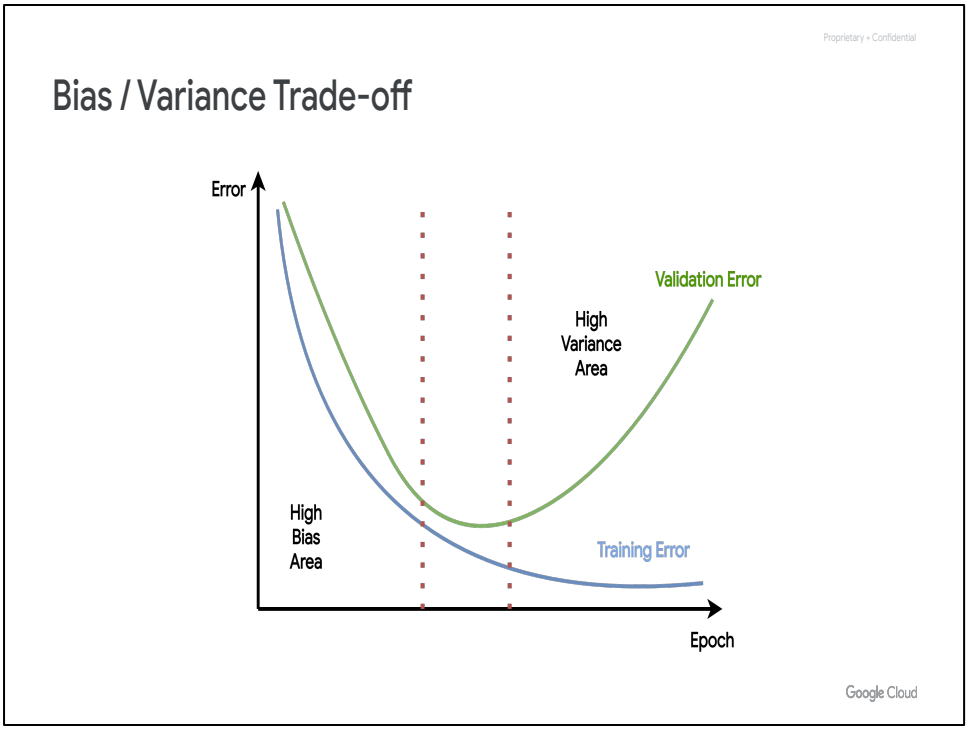
Underfit | Fit | Overfit

High Bias | High Variance

Training and evaluating an ML model is an experiment with finding the right generalizable model that fits your training dataset but doesn't memorize it.

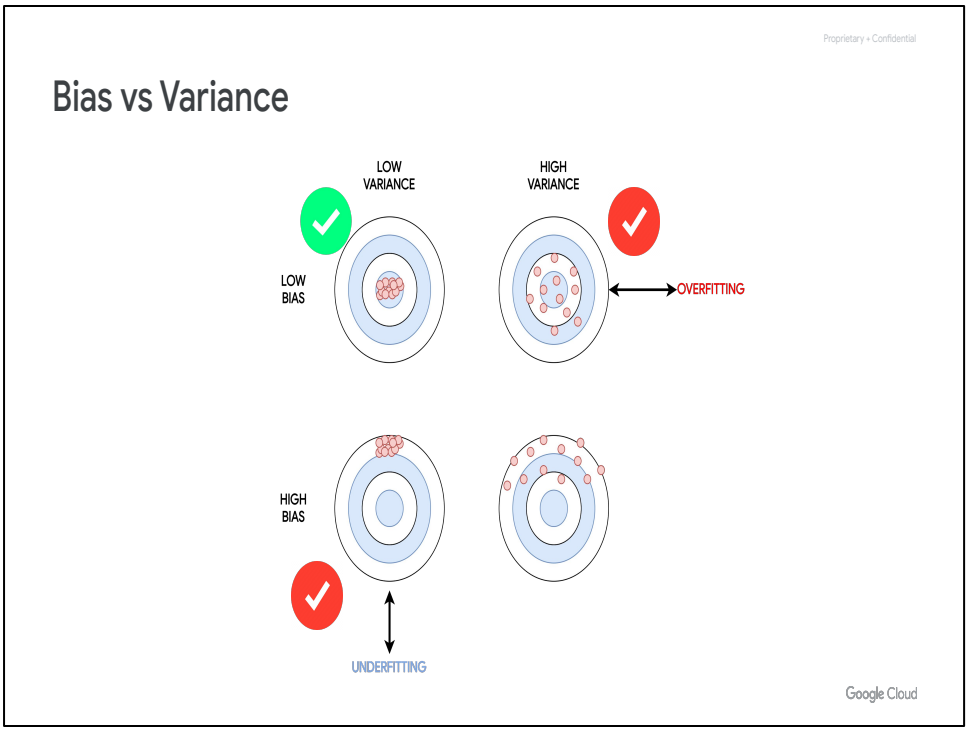
- **Underfitting:** As you see here, we have an overly simplistic linear model that doesn't fit the relationships in the data. You'll be able to see how bad this is immediately by looking at your loss metric during training (and visually on this graph here as there are quite a few points outside the shape of trend line). **This is called underfitting.**

- **On the opposite end of the spectrum is overfitting**, as shown on the right extreme. Here we greatly increased the complexity of our linear model and turned it into an n-th order polynomial which seems to model the training dataset really well -- almost too well. This is where the evaluation dataset comes in -- you can use the evaluation dataset to determine if the model parameters are leading to overfitting. Overfitting or memorizing your training dataset can be far worse than having a model that only adequately fits your data.



- **Bias:** refers to the error caused by the models simplification of the problem. A model with high bias will not do a good job capturing the patterns in the data. It underfits the data.

- **Variance:** refers to the sensitivity of our model to the specifics of the training data. A model with high variance will overfit to the training data. It learns the noise and does not generalize well.



Explanation:

- This visually shows the bias and variance tradeoff take shape in linear models.
- The bullseye is the true value we want to predict and the pink dots are what the model actually predicts

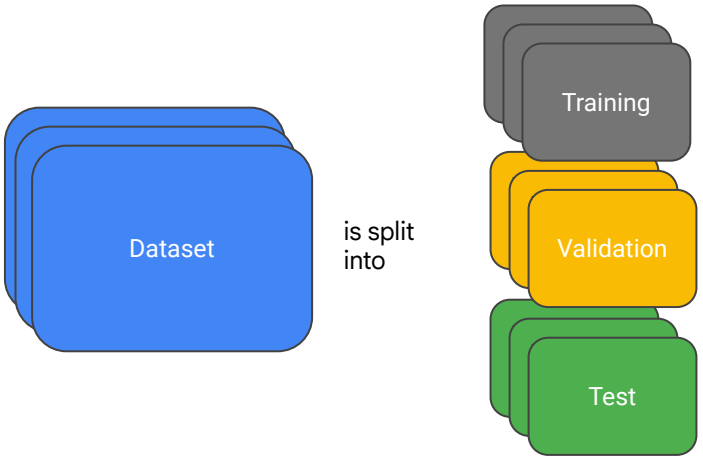
the aim is to visually show the bias and variance tradeoff take shape in linear models.

GOOD: Low bias and low variance:
This is the ideal situation. It means our model is complex enough to capture the patterns in the data (low bias), but not too complex that it overfits (low variance). A model with low bias and variance will have strong predictive performance on both training and new data.

Approaches to solve this problem?
Techniques such as **cross-validation, regularization, boosting,** can be used to manage this trade-off. Through these techniques, one can aim to achieve models that have both low bias and low variance.

Ensure you are evaluating on a separate dataset.

Validation as you train. Test for your final results.



Validation

- assess how well a model will generalize to an independent dataset.

You can use a technique called cross-validation which randomizes the training and validation data each epoch of training. The upside is that you get to use all the data, but you have to train lots more times.

So here's what you have to remember:

- If you have lots of data, use the approach of having a completely independent, held-out validation dataset.
- If you don't have that much data, use the cross-validation approach.

Always additionally have a test dataset for final performance evaluation.

Validação

- avaliar quão bem um modelo será generalizado para um conjunto de dados independente.

Você pode usar uma técnica chamada validação cruzada, que randomiza os dados de treinamento e validação em cada época do treinamento. A vantagem é que você consegue usar todos os dados, mas precisa treinar muito mais vezes.

Então aqui está o que você deve lembrar:

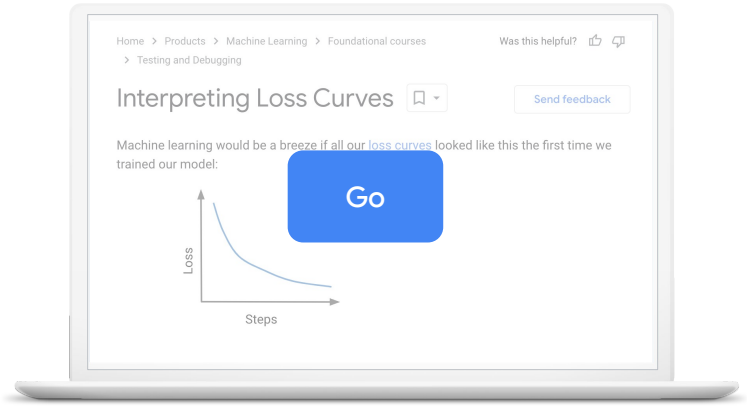
- Se você tiver muitos dados, use a abordagem de ter um conjunto de dados de validação totalmente independente e retido.
- Se você não tiver tantos dados, use a abordagem de validação cruzada.

Além disso, sempre tenha um conjunto de dados de teste para avaliação final de desempenho.

Interpreting Training Loss Curves

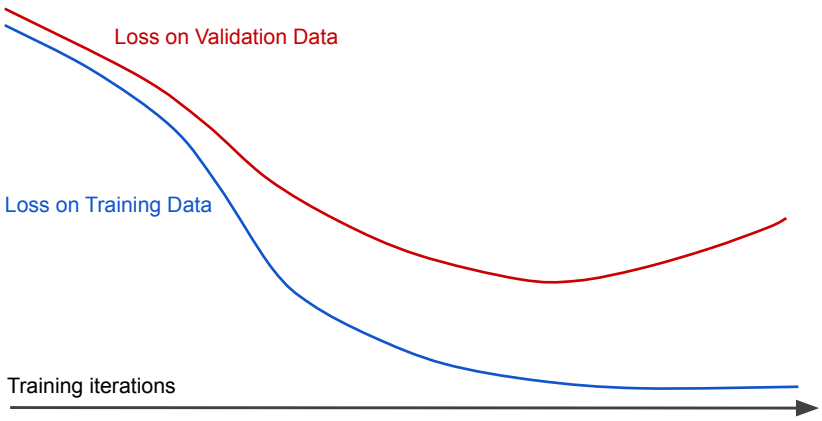
Reference

Interpreting Loss Curves



This could be a good quick demo to share the screen and go through together.

What is happening here? How can we address this?



Remember our goal while training a model is to minimize the loss value. If you graphed the loss curve both on training and test data, it may look something like this. The graph shows Loss on the y axis vs. Time on the x axis.

Notice anything wrong here?

Yeah, the loss value is nicely trending down on the training data but shoots upwards at some point on the test data. That cannot be good!

Clearly, some amount of overfitting is going on here. Seems to be correlated with the number of training iterations.

How could we address this?

We could reduce number of training iterations and stop earlier. Early stopping is definitely an option, but there must be better ones...

Here is where Regularization comes into the picture!

A plot of loss as a function of the number of training iterations. The following plot shows a typical loss curve:

Loss curves can help you determine when your model is converging or overfitting.

- Loss curves can plot all of the following types of loss:
    - **Training loss:** is a critical metric that quantifies how well a model's predictions on the training data align with the actual outcomes

  Here's a breakdown of what **training loss** means and its importance:

  Definition: The training loss, often simply called "loss", is a value calculated using a loss function (or cost function). This function measures the difference between the model's predictions and the true values for the data points in the training dataset.
  Purpose: The primary goal during training is to minimize this loss. As the loss decreases, the model's predictions on the training data become more accurate. In other words, the model is "learning" when the training loss reduces.
  Loss Functions: The specific formula used to compute the loss depends on the task:
    - **For regression tasks** (predicting a continuous value), Mean

- Squared Error (MSE) is a common loss function.
- **For binary classification** (determining one of two classes), Binary Cross-Entropy (or log loss) is often used.
- **For multi-class classification** (determining one of multiple classes), Categorical Cross-Entropy is common

- **Validation loss:** is the value of the loss function for the validation set. It gives an estimate of the model's performance on unseen data, under the current state of training

- **Test loss:** specifically refers to the value of the loss function when it's evaluated on the test set.

Regularization!

The simpler the better!

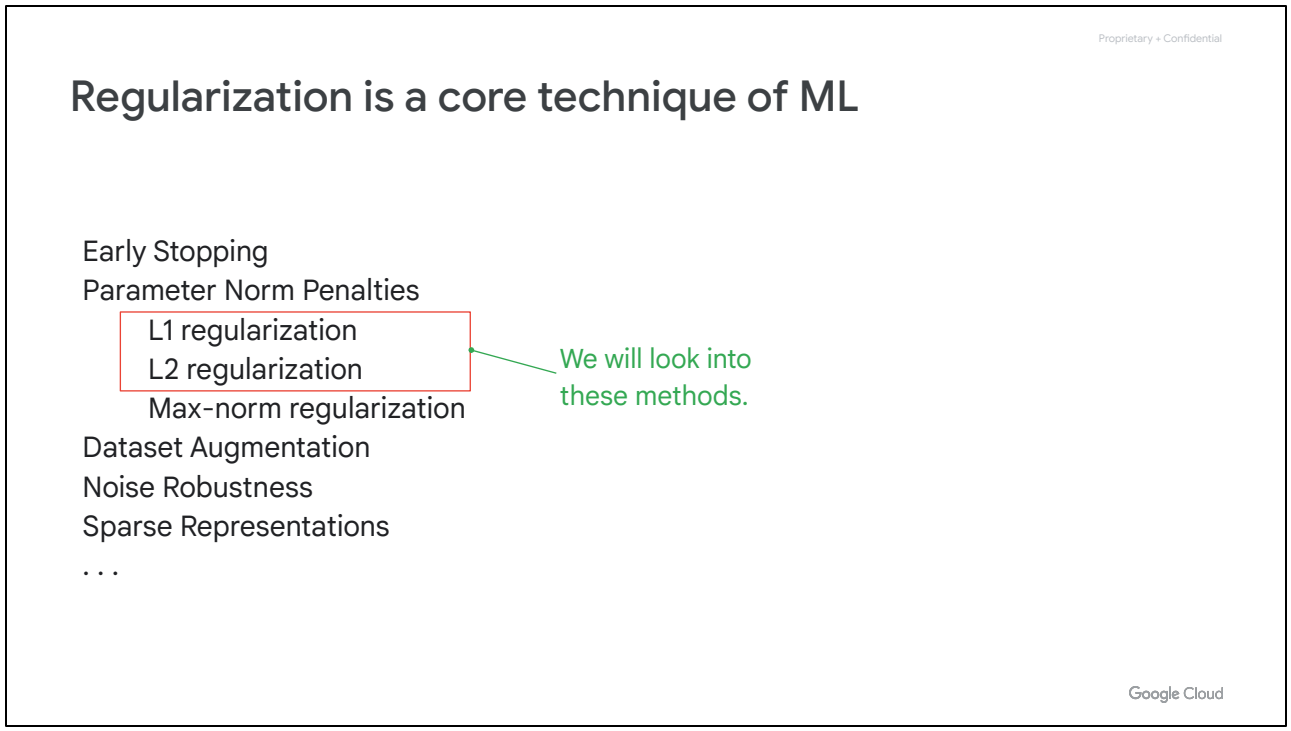
Regularization is used when...

- You have too many features in your model
- You don't know which feature should be discarded
- You want to reduce overfitting

Don't cook with every spice in the spice rack!

Image Source: https://pixabay.com/en/spice-rack-cooking-spices-1650049/ (cc0)

We concluded that simpler models are usually better; we don't want to cook with every spice in the spice rack!



Regularization is one of the major fields of research within Machine Learning.

L1 and L2 regularization are techniques used to prevent overfitting in machine learning models. They add penalties to the loss function to constrain model complexity.

- **L1 regularization** penalizes the absolute size of the weights, and encourages sparsity. Many weights end up being 0.
- **L2 regularization** penalizes the square of the weights, and distributes the penalty more evenly. Most weights end up small, but non-zero.

There are many published techniques and more to come:

- We already mentioned “Early Stopping”
- We also started exploring the group of methods under the umbrella of “Parameter Norm Penalties”.
- There is also “Dataset Augmentation” methods, “Noise Robustness”, “Sparse Representations” and many more.

But, before we do that, let’s quickly remind ourselves what problem “Regularization” is solving for us:

- Regularization refers to any technique that helps generalize a model.
- A generalized model performs well not just on training data, but also on never-seen test data.

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

Question
Which of these is important when performing logistic regression?

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

Answer

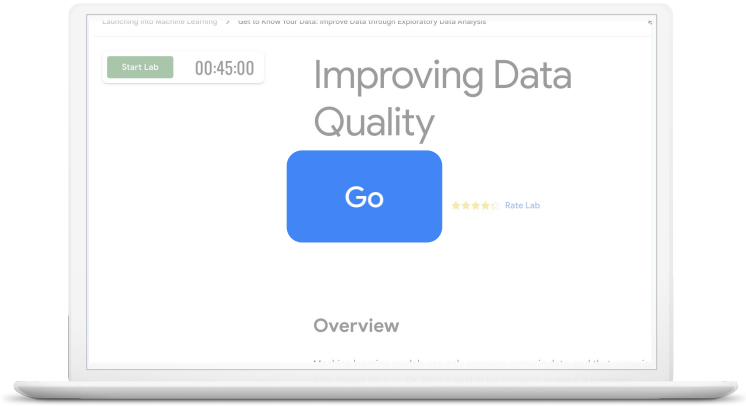
The correct answer is both A & B. Adding regularization to logistic regression helps keep the model simpler by having smaller parameter weights. This penalty term added to the loss function makes sure that cross entropy through gradient descent doesn't keep pushing the weights from going closer and closer to plus or minus infinity and casuing numerical issues. Also, with now smaller logits, we can now stay in the less flat portions of the sigmoid function making our gradients less close to zero and thus allowing weight updates and training to continue.

C is incorrect and therefore so is E because regularization does not transform the outputs into a calibrated probability estimate. The great thing about logistic regression is that it already outputs a calibrated probability estimate since the sigmoid function is the cumulative distribution function of the logistic probability distribution. This allows us to actually predict probabilities instead of just binary answers like yes or no, true or false, buy or sell, etc.

Recommended Lab

Improving Data Quality

From the Course
Launching Into Machine Learning



Working with Cloud Dataprep on Google Cloud
(part of Transform and Clean your Data with Dataprep by Alteryx on Google Cloud)

Questions and answers

Thank you for attending this training!

We love your feedback! Please take a minute to complete the survey and help us improve our courses.


Google Cloud