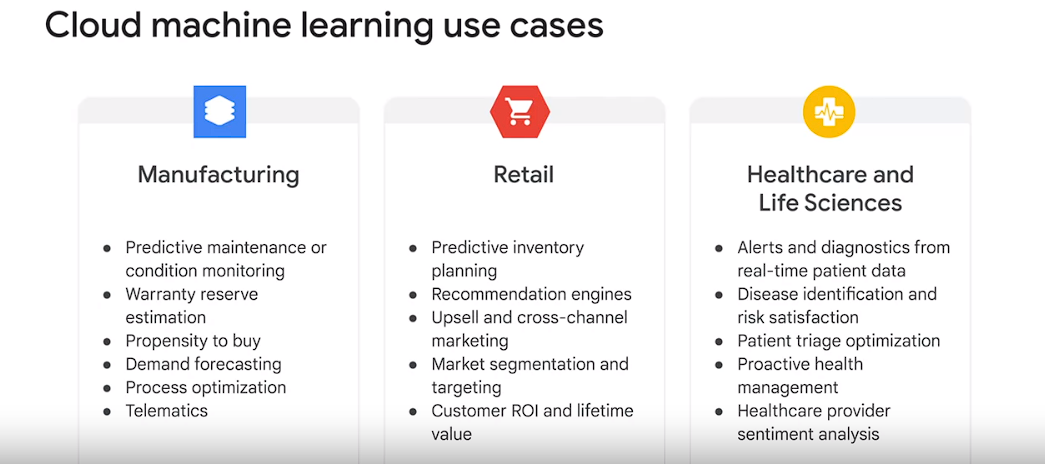
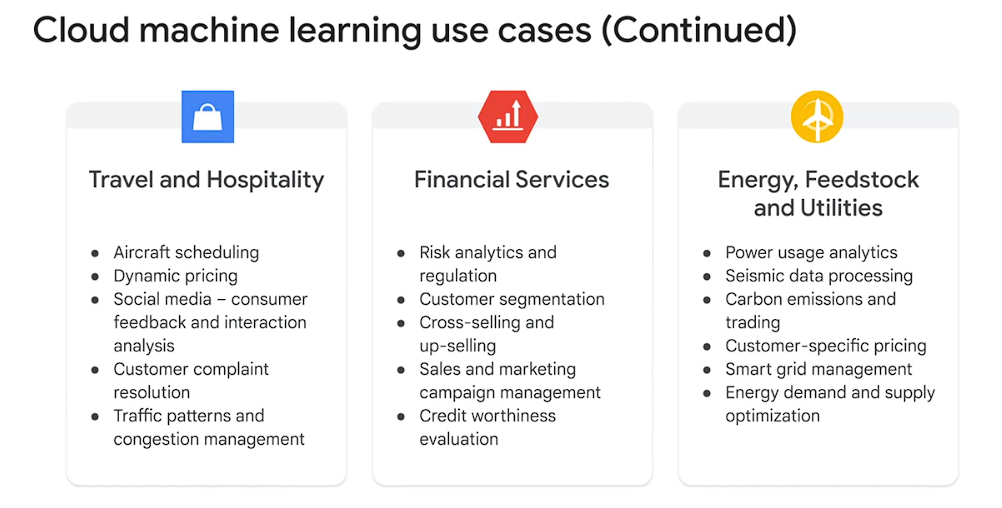
1. **Machine Learning for Predictive Insights**: Machine learning is a method for deriving predictive insights from data using algorithms that are applicable to various datasets.
2. **Backward-Looking Use of Data (Business Intelligence)**: Traditional data analytics often involves looking at historical data to create reports and dashboards for decision-makers, forming the basis of business intelligence.
3. **Need for Predictive Decision Making**: While analyzing historical data can inform decisions, machine learning enables the scalability and repeatability of predictive decision-making processes.
4. **Difference Between AI and Machine Learning**: AI is a broader discipline focused on building machines that think and act like humans, while machine learning is a specific toolset within AI, similar to Newton's laws in physics.
5. **Supervised Learning in Machine Learning**: The form of machine learning discussed is supervised learning, where models are trained using labeled examples consisting of inputs and corresponding true answers.
6. **Adjustable Parameters in Machine Learning Models**: Machine learning models have adjustable parameters that are fine-tuned during the training process to make the model's output as close as possible to the true answers.
7. **Generalization in Machine Learning**: The key to making machine learning models generalize is having a large dataset of labeled examples, allowing the model to learn patterns and make accurate predictions on unseen data.
8. **Two Stages of Machine Learning: Training and Inference**: Machine learning involves two stages - training, where the model learns from labeled examples, and inference, where the trained model is used to make predictions on new, unseen data.
9. **Importance of Operationalizing Machine Learning Models**: It's not enough to focus on the training stage; operationalizing machine learning models, putting them into production, is crucial for running inferences and making real-world predictions.
10. **Neural Networks in Machine Learning**: Neural networks, with multiple layers, are a significant technology in machine learning. Deep neural networks have enabled advancements in challenging tasks like language translation, image classification, and speech understanding.
11. **Google's Use of Machine Learning**: Google extensively uses machine learning in its products, with over 10,000 deep learning models across various services like YouTube, Play, Chrome, Gmail, and more.
12. **Building Multiple Models for Complex Business Problems**: Complex business problems often require breaking them down into smaller problems, each addressed by a specific machine learning model. Avoid the notion of a single, monolithic model solving the entire problem.
13. **Google Photos, Google Translate, and Smart Reply as Examples**: Google Photos, Google Translate, and Smart Reply in Gmail are examples illustrating the application of multiple machine learning models to solve diverse problems, such as image tagging, language translation, and email response generation.

Sure, here's a comprehensive breakdown of how the approach outlined changes the way problems are approached:

1. **Data Collection and Problem Framing**: Instead of trying to anticipate and encode all possible rules for a problem, the focus shifts to collecting data that represents the problem space. In the case of local queries like "coffee near me," data on user preferences, distances to shops, and other relevant factors are collected.
2. **Example-based Learning**: ML involves learning from examples. In the case of local queries, examples consist of user preferences (e.g., whether they liked the recommended coffee shop based on distance). These examples serve as labeled data, where the input is the distance to the shop, and the label indicates whether the user liked the result.
3. **Model Training and Iteration**: Once a sufficient amount of labeled data is collected, a model is trained to make predictions based on new inputs. The model learns to generalize from the examples, finding patterns that optimize for user preferences. It's important to continuously iterate and refine the model as more data becomes available.
4. **Decision Making based on Data**: Instead of relying on predefined rules, decisions are made based on data-driven insights derived from ML models. For example, in the case of the "coffee near me" query, the decision on which coffee shop to recommend is based on a trade-off between factors like distance, quality of coffee, and service time, as learned from user preferences.
5. **Adaptive and Scalable Solutions**: ML-powered solutions are adaptive and scalable. They can continuously improve themselves over time as more data is collected and can handle a wide range of inputs without the need for manual intervention.
6. **Transition from Heuristics to Data-Driven Insights**: While heuristics may be initially used to address a problem, the ultimate goal is to replace them with data-driven insights obtained through ML. This requires a shift in mindset towards trusting data and examples rather than predefined rules.

By embracing this approach, companies can unlock the potential of ML to automate and personalize solutions across various domains, leading to more efficient, effective, and adaptable systems.

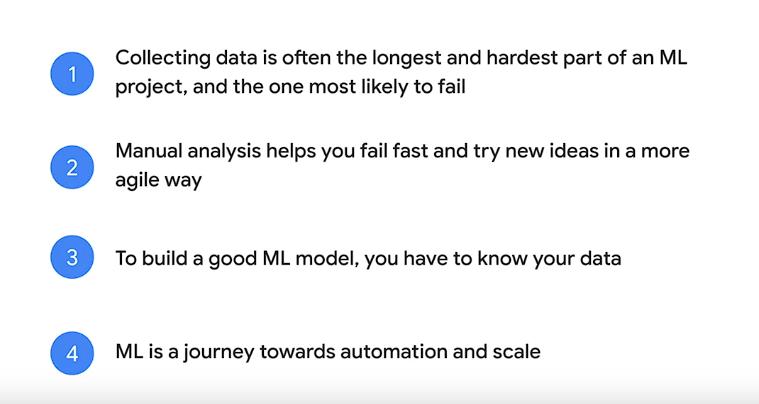




1. **ML Integration into REPs:**
   * Utilize pre-trained models for easy integration.
   * Avoid the need to build custom models when off-the-shelf solutions suffice.
2. **Aucnet's ML Implementation:**
   * Aucnet, the largest real-time car auction service in Japan, implemented ML.
   * ML system detects car model numbers accurately.
   * Provides estimated price ranges for each model.
   * Recognizes the photographed part of the car.
3. **Efficiency Improvements:**
   * Dealers only need to upload photos for classification by the ML system.
   * Eliminates the need for manual specification of car details for each photo.
4. **Ocado's ML Application:**
   * Ocado, the largest online-only grocery in the UK, implemented ML.
   * Used natural language processing (NLP) to process customer emails.
   * Extracts sentiment, entities, and syntax to aid in routing and prioritization.
5. **Shift Towards Conversation Interfaces:**
   * Customers prefer interactive communication over traditional methods.
   * Manual handling of each call is not scalable.
   * Gartner predicts increased investment in conversation interfaces over mobile apps.
6. **Usage of Pre-trained Models by Various Companies:**
   * Giphy employs the vision API for optical character recognition in memes.
   * Social media companies use the vision API to filter inappropriate uploads.
   * Uniqlo utilizes Dialogue Flow to design a shopping chatbot.
7. **Specialization Focus:**
   * Building automated ML models.
   * Utilizing Big Query for ML.
   * Developing custom ML models.

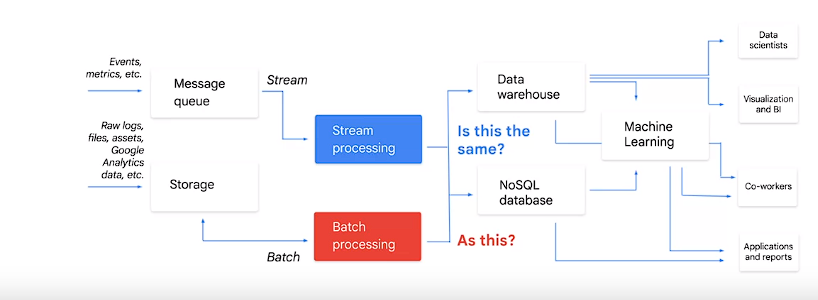
Certainly, here's an expanded summary incorporating all points from the provided text:

1. **Basic Navigation with Google Maps:**
   * Google Maps utilizes a set of rules and algorithms based on road networks, traffic data, and routing algorithms to provide directions.
   * Data collection on roads, traffic conditions, and closures enables efficient route planning.
   * This form of navigation primarily relies on predefined rules and algorithms.
2. **Location-Based Assistance:**
   * In more complex scenarios, such as navigating multi-story buildings, Google Maps employs data from various sources such as WiFi signals, barometric pressure, and walking speed.
   * Machine learning plays a crucial role in analyzing and interpreting diverse data to provide relevant information, such as identifying the floor in a building.
3. **Personalized Recommendations:**
   * Google Maps integrates users' past preferences and context to offer personalized recommendations, like suggesting nearby attractions based on interests and location.
   * Machine learning processes large volumes of data to tailor services according to individual preferences.
4. **Data's Role in Machine Learning:**
   * Data serves as the fuel for machine learning, enabling the transition from generic to personalized services.
   * Increasing data **quantity and variety** enhances the effectiveness of machine learning models.
   * Machine learning enables scalability and automation, transforming manual processes into efficient, data-driven solutions.
5. **Transition to Machine Learning:**
   * Transitioning from manual data analysis to machine learning involves leveraging existing data for successful model development.
   * Manual analysis helps in understanding the data and refining inputs for decision-making.

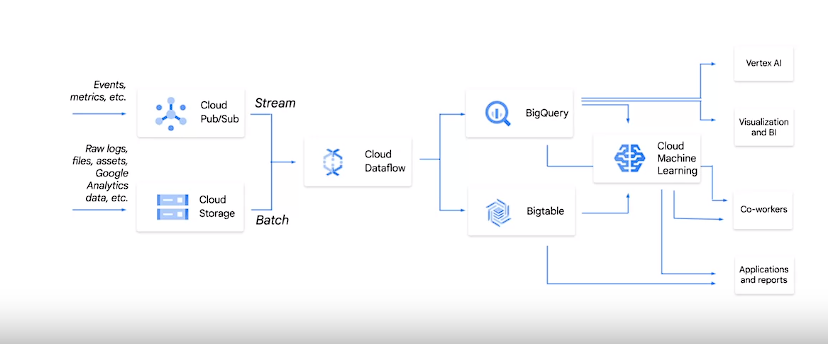


* + Machine learning facilitates automation and scalability, driving efficiency in data analysis and prediction.

1. **Challenges and Solutions in Machine Learning:**

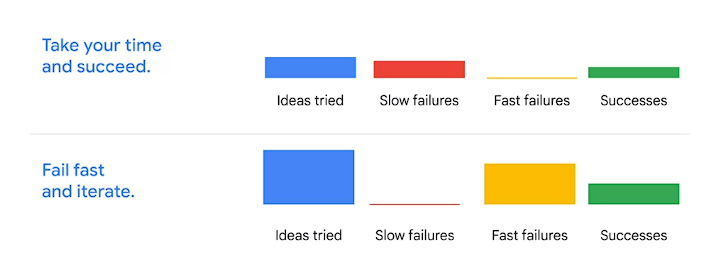


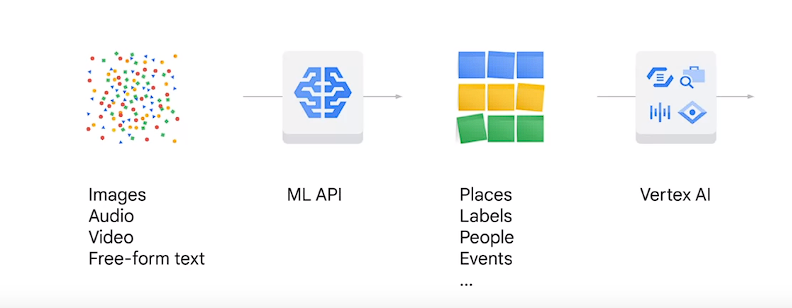
* + Challenges like training-serving skew require consistent data processing pipelines for both training and prediction.
  + Tools like Cloud Dataflow facilitate seamless integration of batch and streaming data processing for machine learning applications.



* + Emphasizing the importance of quantity over complexity in machine learning model development.
  + The performance metrics you care about change between training and predictions as well. During training, the key performance aspect you care about is Scaling to a lot of Data. Distributed training, if you will. During prediction though, the key performance aspect is speed of response, high QPS. This is a key insight behind TensorFlow.

1. **Dealing with Unstructured Data:**
   * While structured data is common, a significant portion of enterprise data is unstructured.
   * Machine learning pipelines help in processing unstructured data through ML APIs, extracting entities and insights for building simpler ML models.

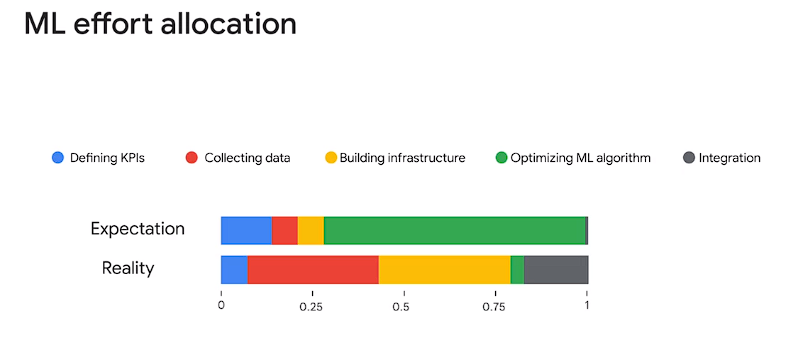




This comprehensive summary highlights the diverse applications of machine learning in Google Maps, ranging from basic navigation to personalized recommendations, and underscores the pivotal role of data in driving these functionalities.

ML

1. Definition of Machine Learning (ML): ML is the process through which a computer writes a program to accomplish a task by analyzing examples and determining the best program to write based on input-output pairs.
2. Comparison with Traditional Software Engineering: In traditional software engineering, a human analyzes the problem and writes code to create a program, while in ML, another computer analyzes examples to determine the best program to write.
3. Framework for Understanding ML: The text provides a simplified framework for understanding ML, emphasizing its practical application in businesses.
4. Effort Allocation in Building ML Systems: Effort in building an end-to-end ML system is distributed across tasks such as defining Key Performance Indicators (KPIs), data collection, infrastructure building, optimizing the ML algorithm, and integrating with existing systems.

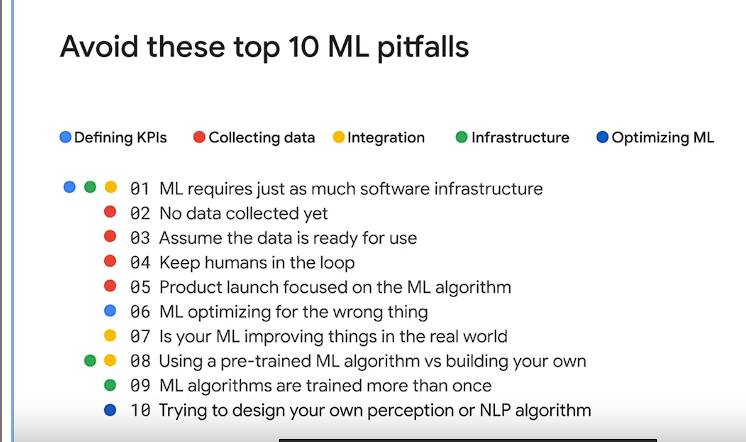


1. Common Focus on Optimizing ML Algorithm: Many practitioners tend to focus solely on optimizing the ML algorithm, paying attention to technical details and tuning hyperparameters.
2. Importance of Data Collection and Infrastructure Building: Data collection and infrastructure building are highlighted as crucial aspects of building successful ML systems, often requiring more effort than optimizing the algorithm itself.
3. Emphasis on Well-Measured User Data in ML: Once an organization reaches the ML stage, well-measured user data reduces the need for extensive KPI definition and organizational effort, providing valuable insights for decision-making.
4. Value of Learning About ML: While ML may not be the sole solution for every problem, the journey toward implementing ML techniques yields significant value, benefiting many aspects of business operations.

Top of Form

Here are the key points extracted from the text:

1. **Secret Source in Machine Learning (ML):**
   * The secret source in ML lies not just in code or algorithms but in organizational know-how gained from managing numerous ML systems.
2. **Importance of Technical ML Skills:**
   * Developing technical ML skills, including software and data handling skills, is essential for becoming proficient ML strategists.
   * Google's experience can help practitioners avoid common pitfalls in ML implementation.
3. **Top 10 Pitfalls in ML Implementation:**



* + Failing to recognize that training an ML algorithm may not be faster than writing traditional software due to additional complexities.
  + Neglecting to collect necessary data before attempting ML implementation.
  + Overlooking the importance of regularly reviewing and maintaining collected data to ensure its quality and relevance.
  + Forgetting to incorporate humans in the loop for reviewing data, handling cases, and curating training inputs.
  + Launching a product solely based on its ML algorithm without considering other valuable features.
  + Optimizing an ML system for the wrong metrics, leading to unintended consequences.
  + Failing to measure the real-world impact of an ML algorithm's performance improvement.
  + Underestimating the effort required to fuse pre-trained ML algorithms with custom-built solutions.
  + Misconception that ML algorithms need to be trained only once, whereas they often require frequent retraining.
  + Attempting to design in-house perception algorithms without considering existing highly tuned alternatives available off the shelf.

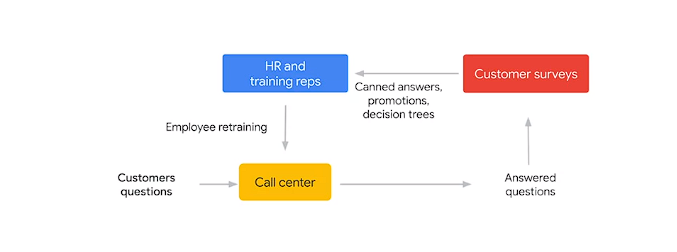
1. **Value of the ML Journey:**
   * Most of the value in the ML journey comes along the way, even if the destination is not fully reached.
   * ML improves various aspects of operations and products, providing users with great experiences that are hard to replicate.
   * Attempting to jump to fully automated ML solutions often leads to suboptimal outcomes, emphasizing the importance of a realistic path.

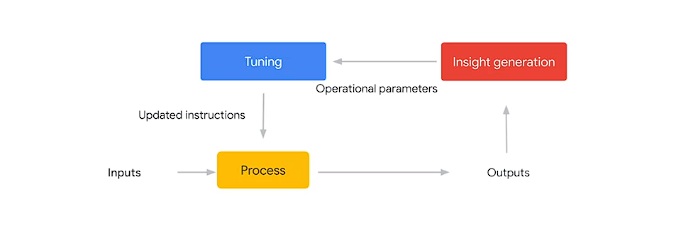
These points highlight the importance of technical skills, data collection, human involvement, and careful consideration of pitfalls in implementing ML systems, along with the value derived from the journey towards ML adoption.

Top of Form

Here are the key points extracted from the text:

1. **Evolution of Business Processes:**
   * Transitioning from a non-ML to an ML solution in an organization involves considering the evolution of business processes.
   * A business process refers to any set of activities a company performs directly or indirectly to serve customers, which must be continually improved.
2. **Importance of Feedback Loops:**
   * Understanding feedback loops within business processes is crucial to understanding the role of ML in large organizations.
   * Almost every business process incorporates a feedback loop, which helps in converting operational expertise into better future outcomes.
3. **Concrete Example:**
   * An example is provided of an internet service provider's call center, where customer inquiries are handled, and feedback from these interactions is utilized to improve future responses.
   * The feedback loop involves collecting customer survey responses, extracting insights, updating operational parameters, and retraining employees.





1. **General View of Feedback Loop:**
   * The text presents a more general view of the feedback loop, involving input and output to a process, regeneration of insights, tuning of the original process with updated instructions, and continuous improvement.

These points emphasize the iterative nature of improving business processes through feedback loops and how ML can enhance this process by generating insights for better future outcomes.

Top of Form

Here's a breakdown of the phases in the path to ML as described in the text:

1. **Individual Contributor Phase:**
   * In this phase, tasks or business processes are performed by a single person.
   * The example given is a receptionist in an office building, who handles phone calls, inquiries, and provides directions.
   * Tasks are not parallelized or scaled, and the process is usually informal.
2. **Delegation Phase:**
   * As tasks become more important to the company, they are delegated to multiple people who perform the same task in parallel.
   * Example: Store checkers in retail stores, where roles are formalized, and rules are established for consistency among individuals.
3. **Digitization Phase:**
   * In this phase, the core repeatable part of the task or business process is automated with computers.
   * Example: ATM machines, which automate cash withdrawal services, providing customers with a high-quality service due to repeatable and well-automated interactions.
4. **Big Data and Analytics Phase:**
   * Organizations use a lot of data to build operational and user insights, optimizing each part of the process for better outcomes.
   * Example: Toyota's lean manufacturing philosophy, where they measure everything about their facilities to improve construction processes and delivery times.
5. **Machine Learning Phase:**
   * Utilizing data from the previous steps, computer processes are automatically improved using machine learning techniques.
   * Example: YouTube recommendations, where algorithms learn from user interactions to personalize video recommendations based on user preferences and behaviors.

These phases represent the progression towards ML-enabled solutions within organizations, starting from individual tasks performed manually to fully automated and optimized processes driven by machine learning algorithms.

Top of Form



Bottom of Form

ChatGPT can make mistakes. Cons

Here are all the extracted points from the provided text:

**Individual Contributor Phase:**

* Prototype opportunities exist in this phase, allowing for experimentation and learning.
* Skipping this phase can be risky as it may lead to incorrect assumptions and difficulty in convincing stakeholders.
* Lingering in this phase too long can result in organizational knowledge loss and scalability issues.

**Delegation Phase:**

* Increasing the number of employees working on a business process allows for gentle investment ramp-up while maintaining flexibility.
* Skipping this step may lead to a lack of formalization and definition of success, hindering future scalability.
* Delegation is a halfway step to formalizing business processes and allows for organizational buy-in for success metrics.
* It provides an opportunity for product learning by analyzing the diversity in human responses.

**Digitization Phase:**

* Digitization involves getting computer systems to perform repetitive parts of the process, aiding in automation.
* Skipping this step can result in difficulties in serving ML at scale and confusion between IT and ML projects.
* It involves a significant upfront investment and comes with its own risks.
* Failure in either an IT or ML project within this phase can lead to project failure and management confusion.

**Big Data and Analytics Phase:**

* Everything about internal operations and external users is measured to generate insights in this phase.
* Skipping this step can make it challenging to train ML algorithms and measure success accurately.
* Reassessing the original definition of success is possible with the wealth of data available in this phase.

**Machine Learning Phase:**

* Completes the feedback loop by automating each block between measuring success and tuning the software algorithm.
* ML systems can outpace human capabilities in handling inputs and nuances, leading to faster answers and more nuanced treatment of details.
* ML algorithms can learn from many disparate interactions and provide nuanced treatment of details.
* ML should be seen as a way to expand or scale the impact of people, not as a replacement for them.

These points outline the progression from manual processes to ML-enabled solutions and highlight the importance of each phase in achieving successful automation and optimization.

Top of Form

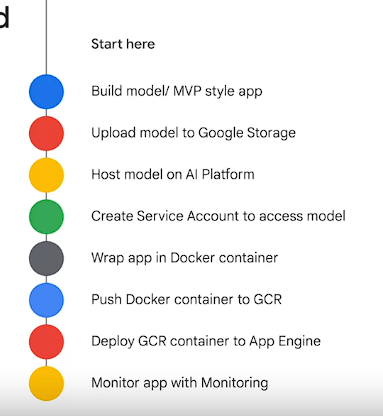
**Experimentation Phase:**

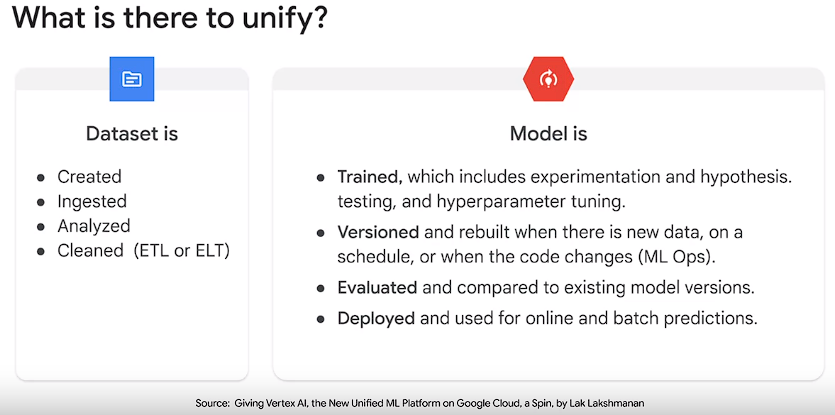
* In the experimentation phase of machine learning development, the process includes framing the problem, preparing the data, experimenting, and evaluating the model.
* Framing the problem involves identifying the use case and minimum business application requirements.
* Preparing the training data may involve using a subset of a larger dataset, performing exploratory data analysis (EDA), improving data quality, and feature engineering.
* Feature engineering involves combining features to create new ones.
* During experimentation, different models are tested and compared using performance metrics such as recall, precision, F1 score, or cross-entropy.
* Different model architectures, input datasets, hyperparameters, and hardware are explored and tested.
* CNNs or convolutional neural networks are used for image classification, object detection and recommend their systems. RNNs or recurrent neural networks are used for sequence modeling, next word prediction, translating sounds to words and human language translation. Sorting and clustering architectures are used for anomaly detection and pattern recognition.
* Play video starting at :2:16 and follow transcript2:16
* GANs or generative adversarial networks are used for anomaly detection, pattern recognition, cybersecurity, self driving cars and reinforced learning.
* Model architectures such as CNNs, RNNs, sorting, clustering, and GANs are used for various tasks.
* Input datasets may include numerical, bivariate, multivariate, categorical, and correlation datasets.
* Different hyperparameters like learning rates, number of layers, num\_estimators, and max depth are adjusted based on the algorithm.
* Hardware types like CPUs, GPUs, and TPUs are utilized for training models.

**Moving to Production:**

* Transitioning from experimentation to production involves packaging, deploying, and monitoring the model.
* Packaging code for production may involve building and installing a Python package for the predictive model.
* Models can be deployed on various platforms such as mobile devices or web services.
* Deploying the model requires setting up endpoints to serve predictions.
* Monitoring the model performance is essential to detect issues like model drift, outliers, or data quality problems.
* Key model performance metrics include model drift, performance, outliers, and data quality.

**Components of ML Workflow:**

* ML development involves various components like dataset creation, model training, evaluation, deployment, and management.
* Vertex AI provides unified definitions and implementations for datasets, training pipelines, models, and endpoints.
* Datasets can be structured or unstructured and stored anywhere on Google Cloud.
* Training pipelines consist of containerized steps for training ML models, aiding in generalization, reproducibility, and auditability.
* Models are ML models with metadata built using a training pipeline or directly loaded.
* Endpoints can be invoked for online predictions and explanations, with one or more models and versions. 



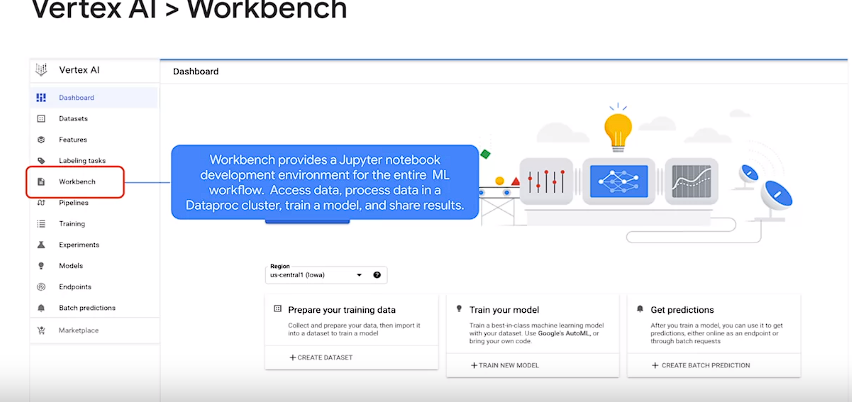
**Vertex AI Features:**

* Vertex AI offers flexibility in training methods, including AutoML for minimal technical effort and custom training for complete control.
* AutoML enables quick prototyping and exploration of datasets before investing in development.
* Custom training allows complete control over training application functionality for targeted outcomes.
* Vertex AI aims to streamline ML workflow with fast experimentation, accelerated deployment, and simplified model management.

These points provide insights into the experimentation phase of ML development and the transition to production, as well as the features offered by Vertex AI for managing the ML workflow.

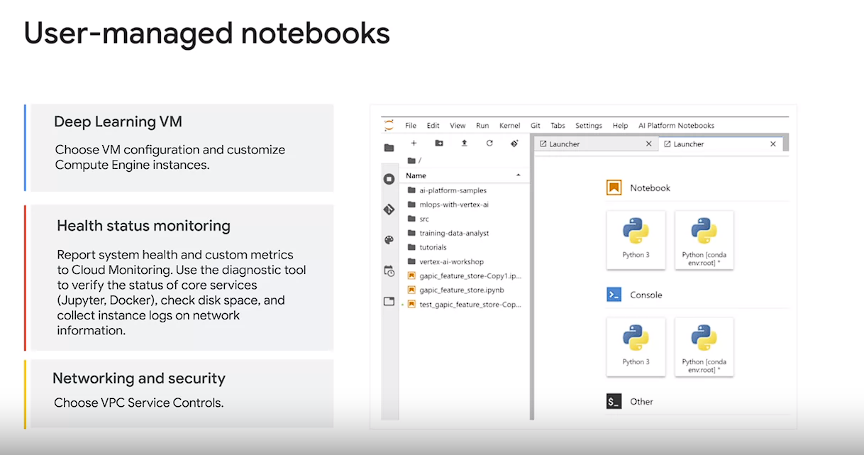
Top of Form

* Overview of Vertex AI Components provided in the lesson, emphasizing a unified approach to machine learning for transitioning ML projects from experimentation to production.
* Explanation of the navigation bar in the Vertex AI dashboard, highlighting various components:



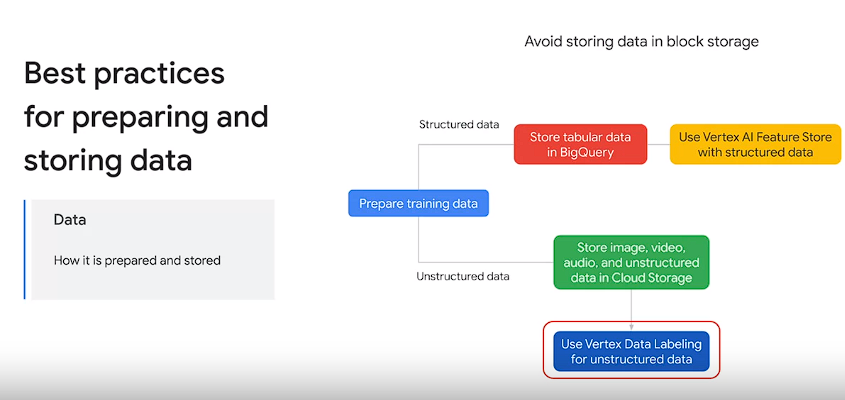
* 1. **Datasets**: Managed by Vertex AI after data loading from Cloud storage or BigQuery, can be linked to models for further analysis and training.
  2. **Vertex AI Feature Store**: Fully managed repository facilitating ingestion, serving, and sharing of ML feature values within the organization, offering scalable storage and compute resources.
  3. **Data Labeling Tasks**: Enables requesting human labeling for datasets (e.g., video, image, text) to prepare for custom machine learning model training, ensuring accurate annotations through user-provided instructions.
  4. **Vertex AI Workbench**: Jupyter notebook-based development environment supporting the entire data science workflow, from data access and processing to model training, evaluation, and result sharing, all within the Jupyter lab interface.
  5. **Vertex AI Pipelines**: Aids in automating, monitoring, and governing ML systems by orchestrating workflows in a serverless manner, while utilizing Vertex ML metadata to store artifacts, ensuring traceability and reproducibility.
  6. For example, an ML models lineage may include the training data, hyperparameters, and code that we use to create the model. The key takeaways are that, pipelines allow you to automate and monitor and experiment with interdependent parts of an ML workflow. ML pipelines are portable, scalable, and based on containers. Each individual parts of your pipeline workflow, for example, creating a dataset or training a model is defined by code. This code is referred to as a component. Each instances of a component is called a step
  7. **Training Models on Vertex AI**: Offers options such as AutoML for automated training or custom training for greater customization, providing features like distributed training, hyperparameter tuning, and GPU acceleration for enhanced model performance.
  8. **Vertex AI Experiments**: Incorporates Vertex Vizier for hyperparameter tuning in complex ML models, facilitating the comparison of different studies using TensorBoard for efficient experimentation.
  9. **Model Management and Deployment**: Utilizes Vertex AI model resources for managing models on Google Cloud, enabling deployment, prediction generation, and hyperparameter tuning, regardless of where the models were trained.
  10. **Deployment Options**: Allows models to be deployed on Vertex AI for serving predictions, with batch prediction functionality available for processing group prediction requests efficiently.

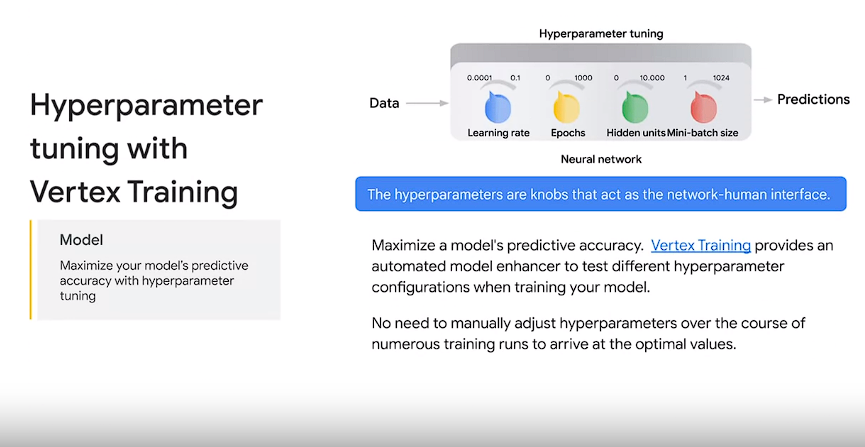
Use batch prediction when you don't require an immediate response and want to process accumulated data with a single request.

* 1. **Vertex ML Metadata**: Stores artifacts and metadata for pipelines run using Vertex AI Pipelines, capturing essential information such as training data, hyperparameters, model accuracy, and batch prediction results, facilitating model performance analysis and improvement.
* Key Takeaway: Vertex AI components streamline automation, monitoring, and experimentation in the ML workflow, providing scalability, portability, and container-based infrastructure for efficient model development and deployment.
* **Vertex AI Workbench Options**:
  1. **Managed Notebooks**: Google-managed environments offering integrated features for an end-to-end notebook-based production environment.
  2. **User-Managed Notebooks**: Deep Learning VM instances providing extensive customization and control over the environment.
* **Features of Managed Notebooks**:
  1. (both) Pre-packaged with JupyterLab and deep learning packages, supporting TensorFlow and PyTorch frameworks.
  2. (both) GPU accelerator support and GitHub repository sync.
  3. (both) Protected by Google Cloud authentication and authorization.
  4. Suitable for data exploration, analysis, modeling, and end-to-end data science workflows.
  5. Allows workflow-oriented tasks within JupyterLab interface, with integrations and features for implementing data science workflows.
  6. Control over hardware and frameworks directly from JupyterLab.
  7. Ability to scale hardware resources without leaving JupyterLab, enabling quick testing and scaling up for larger datasets.
  8. Support for custom Docker container images alongside pre-installed frameworks.
  9. Access to data via Cloud Storage and BigQuery extensions within JupyterLab.
  10. Integration with Dataproc for processing data quickly.
  11. Automated shutdown for idle instances to manage costs effectively.
* **Features of User-Managed Notebooks**:
  1. Customizable Deep Learning VM instances allowing users to specify machine type and framework during creation.
  2. Flexibility to change machine type after creation, though framework changes require instance restart.
  3. Option to make manual modifications such as software updates and package versions.
  4. Health status monitoring tools including a built-in diagnostic tool for verifying core services status and disk space usage.
  5. Networking and security customization options, including VPC Service Controls and built-in networking features.
  6. Ability to configure instances manually to meet specific networking and security requirements.
  7. 

Top of Form

* **Best Practices for Data Preparation and Storage**:



* + Extract data from source systems and convert to ML training optimized format.
  + Store tabular data in BigQuery, and intermediate processed data for speed.
  + For maximum speed, it's better to store materialize data instead of using views, or subqueries for training data.
  + Store unstructured data like images, videos, and audio in large container formats in cloud storage.
  + Use Vertex data labeling for human labeling tasks on unstructured data.
  + Avoid storing data in block storage or virtual machine hard disks.
  + Utilize Vertex AI feature store for structured data, fetching existing features or creating new ones.
  + Set up periodic jobs to compute updated feature values and ingest them into Vertex AI feature store.
* **Best Practices for Training a Model with Vertex AI**:
* 
  + Use Notebooks instance for small datasets, and **Vertex training** service for larger or distributed training.
  + Vertex training service provides pre-built algorithms and supports custom code for fully managed training.
  + Package and push the training application with your model to a cloud storage bucket.
  + Utilize pre-built containers in Vertex Training for simplicity.
  + Utilize Vertex Explainable AI for insights into model predictions and feature attributions.
  + Maximize predictive accuracy using hyperparameter tuning provided by Vertex Training.
  + Use Workbench Notebooks with tools like What-If Tool (WIT) and Language Interpretability Tool (LIT) for model evaluation and bias analysis.
  + Utilize Vertex AI TensorBoard for tracking and comparing experiment metrics, visualizing model graphs, and projecting embeddings.

These best practices ensure efficient and effective implementation of machine learning on Vertex AI, covering data preparation, model training, and evaluation.

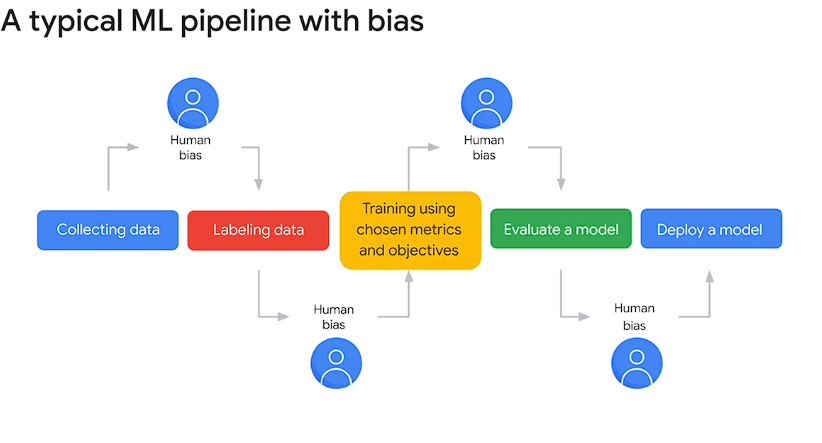
**Best Practices for Data Preprocessing**:

* Use BigQuery for processing tabular data and Dataflow for unstructured data.
* Leverage BigQuery ML in BigQuery for machine learning tasks.
* Consider using managed datasets in Vertex AI to link data to models, providing descriptive statistics and automatic/manual splitting into train, test, and validation sets.
* Managed datasets offer clear linkage between data and custom-trained models, but they are optional if more control over data splitting or lineage is desired.
* For large volumes of unstructured data, use Dataflow with the Apache Beam programming model **to convert data into binary formats like TFRecord**, improving training performance.
* Use a **combination of Dataflow and the Pandas library for transformations not expressible in Cloud SQL or for streaming data.**
* For TensorFlow model development, utilize **TensorFlow Extended (TFX) to prepare data for training.**
* **TensorFlow Transform, a TFX component, enables defining and executing preprocessing functions to transform data efficiently.**
* **Best Practices for Setting Up Machine Learning Environment**:
  + Use notebooks for experimentation and development tasks, including coding, job starting, query running, and status checking.
  + Create a separate notebook instance for each team member to facilitate collaboration and manage dependencies effectively.
  + Stop notebook instances when not in use to optimize cost and resource utilization.
  + Implement security measures to protect Personally Identifiable Information (PII) in notebooks, following guidelines provided in the Notebook Security Blueprint and accompanying deployable blueprint on GitHub.
  + Store prepared data and models in the same Google Cloud project to ensure accessibility and reproducibility, especially when dealing with multiple datasets or projects.
  + Consider customizing Google Cloud properties like network configurations, access management, and software settings for container-associated notebooks.
  + Enhance performance and reduce costs by optimizing machine learning workloads, although detailed strategies for this are beyond the scope of the course.

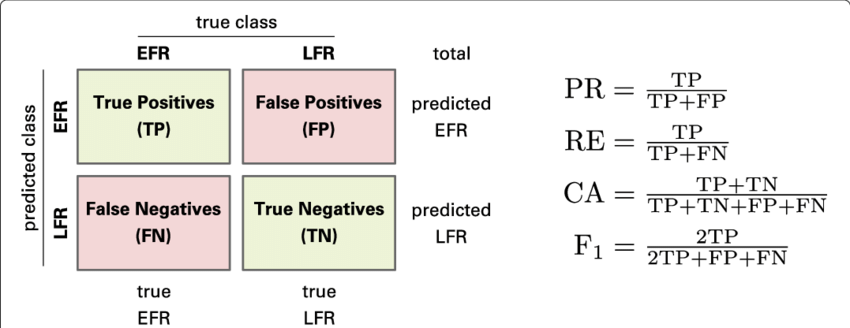
Top of Form

**Tools Interacting with Vertex AI**:

* **Cloud Console**: Deploy models to the cloud and manage datasets, models, endpoints, and jobs through a user-friendly interface integrated with Google Cloud services like Cloud Logging and Cloud Monitoring.
* **Client Libraries**: Vertex AI provides client libraries for various programming languages, offering an optimized developer experience with language-specific conventions and styles for making calls to the Vertex AI API.
* **Google API Client Libraries**: Alternatively, you can use Google API Client Libraries to access the Vertex AI API in languages such as Dart. These libraries simplify the process by allowing you to build representations of API resources and objects, reducing the need for direct HTTP requests.
* **Vertex REST API**: Offers RESTful services for managing jobs, models, and endpoints, as well as making predictions with hosted models on Google Cloud Platform.
* **Deep Learning VM Images**: Optimized virtual machine images designed for data science and machine learning tasks, pre-installed with key ML frameworks and tools. Available with support for various combinations of frameworks and processors, including TensorFlow Enterprise, TensorFlow PyTorch, and generic high-performance computing.
* **Deep Learning Containers**: Docker containers with pre-installed data science frameworks, libraries, and tools, providing performance-optimized and consistent environments for quickly prototyping and implementing workflows.
* **Origin of Bias in Machine Learning Models**:
  + Explains the origins of bias in machine learning models and emphasizes the importance of building inclusive systems.
* **Tradeoffs in Machine Learning Systems**:
  + Discusses the tradeoffs inherent in machine learning systems and how these tradeoffs correspond to evaluation metrics.
* **Equality of Opportunity**:
  + Introduces the concept of equality of opportunity, which aims to achieve equal chances of correct classification regardless of sensitive attributes in machine learning systems.
* **Understanding Data for Inclusive Machine Learning**:
  + Highlights the significance of comprehensively understanding data for achieving inclusive machine learning systems.
* **Facets: Visualization Tool for Machine Learning Data**:
  + Showcases Facets, an open-source visualization tool for exploring and assessing the inclusiveness of training data in machine learning.

1. Bias in machine learning is introduced through exposure to human biases during the learning process.
2. Individuals may have biases towards certain types of shoes, which can be analogous to biases in machine learning.
3. Machine learning involves computers learning solutions by identifying patterns in data, contrasting with traditional programming where solutions are manually coded.
4. Despite being based on data, machine learning is not neutral, as human biases inevitably become embedded in the technology.
5. Various forms of bias in machine learning are highlighted, including interaction bias and selection bias.
6. Latent bias For example, if you were training a computer on what a physicist looks like and you're using pictures of past physicists, your algorithm will end up with a latent bias skewing towards men
7. Interaction bias occurs when the data used for training reflects existing biases in society.
8. Selection bias arises when the data chosen for training is not representative of all groups.
9. Efforts are being made to address bias in technology, such as preventing offensive or misleading content from appearing in search results.
10. Tools are provided for users to flag inappropriate suggestions, contributing to the mitigation of bias in technology.
11. Awareness and proactive measures are emphasized as crucial in addressing bias in machine learning to ensure technology works for everyone.
12. Unconscious biases exist in data, which can be categorized into two forms: biases existing in the world and biases introduced during data collection and labeling. These biases can manifest in various properties such as gender, race, and sexual orientation, among others.
13. Human biases manifest in data samples through reporting bias and selection bias. Reporting bias occurs when subjects choose to disclose only certain aspects about themselves or their opinions, leading to incomplete or skewed data representation. Selection bias arises when the subjects included in the data sample represent only a privileged type of user, neglecting other demographic groups or perspectives.
14. Reporting bias occurs when subjects only reveal certain aspects about themselves or their opinions.
15. Selection bias arises when only a privileged type of user is represented in the data sample.
16. Biases in data collection and labeling procedures include confirmation bias and automation bias. Confirmation bias involves seeking data that confirms hypotheses, potentially overlooking contradictory evidence. Automation bias occurs when only easily automatable data is used, leading to a biased dataset that may not accurately represent the full spectrum of available information.
17. Confirmation bias involves seeking data that confirms hypotheses.(Women do not wear a tie??
18. Automation bias occurs when only easily automatable data is used.
19. Biases in data collection and labeling affect the entire machine learning pipeline, from data preprocessing to model development and evaluation. These biases can impact the performance and fairness of machine learning models.
20. Biases can appear at different stages of the machine learning pipeline, including data collection, data labeling, and model development. It's essential to address biases at each stage to mitigate their impact on the final model's outcomes.
21. Google has announced seven AI principles to guide their work and address biases in AI development and usage. These principles serve as concrete standards to govern research, product development, and business decisions related to AI technologies.
22. ML models may learn or amplify problematic biases existing in real-world data. Despite good intentions, machine learning models trained on biased data may perpetuate or exacerbate existing societal biases, such as those related to race, gender, religion, or other characteristics.
23. A checklist is provided to identify situations where bias-related issues may arise in use cases or products. This includes considerations about the type of data being used, its potential correlation with personal characteristics, and the potential negative impacts on individuals or communities.
24. Google provides tools to diagnose fairness issues in data, labels, and predictions, such as the **What If Tool,** available within TensorBoard. These tools enable developers and researchers to analyze and address biases in their machine learning models, promoting fairness and inclusivity in AI systems.
25. 
26. **Introduction to Unconscious Biases in Data:**
    * Unconscious biases exist in data, stemming from human biases present in the world, such as gender, race, and sexual orientation.
    * These biases manifest in data samples through reporting bias (subjects revealing only certain aspects) and selection bias (privileged representation of users).
27. **Examples of Bias in Machine Learning:**
    * In the context of building a fraud detection model, biases can arise from limited data representation, such as only card transactions being considered fraudulent.
    * Human biases can also affect data collection and labeling procedures, including confirmation bias (seeking data confirming hypotheses) and automation bias (relying on easily automated data).
28. **Impact of Biases in Data Collection and Labeling:**
    * Biases affect the entire machine learning pipeline, from data collection to model output, potentially resulting in biased outcomes.
    * Biases can appear in data collection (availability of data), labeling (human annotator biases), and model development (objectives disadvantaging certain groups).
29. **Google's Approach to Responsible AI:**
    * Google has established seven AI principles to guide responsible AI development and usage.
    * These principles actively govern research, product development, and business decisions to ensure ethical and fair AI practices.
30. **Checklist for Bias-Related Issues:**
    * Considerations include the use of personal data like biometrics, race, religion, etc., and their correlation with other characteristics.
    * Tools are available to diagnose fairness issues in data, labels, and predictions.
31. **Confusion Matrix and Inclusive Machine Learning:**
    * A confusion matrix helps evaluate machine learning performance, particularly for classification problems.
    * It breaks down performance into true positives, false negatives, false positives, and true negatives.
    * Focus on evaluation metrics like true positive rate, false positive rate, precision, and recall to identify bias in model predictions.
    * Different applications may prioritize minimizing false positives or false negatives based on trade-offs and impact considerations.
32. **Visualizing and Evaluating Sub-Group Performance:**
    * Visualizing evaluation metrics across sub-groups helps identify disparities and ensure inclusivity.
    * By evaluating metrics across diverse demographics, machine learning systems can be made more equitable and inclusive.
33. **Conclusion:**
    * Evaluating metrics is crucial for measuring the inclusivity of machine learning systems and ensuring ethical AI practices.
    * Consideration of trade-offs between false positives and false negatives is essential in making informed decisions about model performance.

Top of Form



1. **Understanding Model Errors:**
   * Recognizing that ML systems will make mistakes.
   * Importance of understanding and assessing these errors and their impact on user experience.
2. **Introduction to Equality of Opportunity:**
   * Discussion on evaluating inclusion during ML model development.
   * Explanation of equality of opportunity: ensuring all users have an equal chance of a desirable outcome.
3. **Equality of Opportunity in Loan Approval Model:**
   * Illustration using a toy classifier to predict loan repayment likelihood.
   * Use of credit score as a proxy for multiple variables, with varying distributions across groups.
   * Introduction of threshold setting to determine loan approval based on credit score.
4. **Considerations in Threshold Selection:**
   * Trade-offs in threshold selection: balancing correct decisions with financial implications.
   * Importance of setting thresholds that maximize profit while minimizing incorrect decisions.
5. **Formal Definition of Equality of Opportunity:**
   * Formal setup: protected attribute (A), outcome (Y), predictor (Y hat), and thresholds based on A.
   * Goal: Equal true positive rate across groups, ensuring equal chance of positive classification.
6. **Application of Equality of Opportunity in Loan Predictor:**
   * Scenario with two groups (blue and orange) and conditions for loan approval.
   * Evaluation of thresholds to ensure equal performance across groups, despite differences in distributions.
7. **Challenges and Solutions:**
   * Challenges of group-unaware approach and mathematical implications.
   * Importance of optimizing thresholds based on equality of opportunity to ensure fairness and efficiency.
8. **Takeaway and Conclusion:**
   * Importance of using equality of opportunity criteria to clarify core issues in ML model development.
   * Transfer of burden of uncertainty from groups with most uncertainty to model creators, incentivizing investment in better classifiers.

Here are the points covered in the passage about Facets, an open-source data visualization tool:

1. **Understanding Data for Better Results:**
   * Importance of understanding large datasets for optimal machine learning model performance.
   * Challenge posed by datasets with numerous data points and features.
2. **Introduction to Facets:**
   * Overview of Facets, an open-source data visualization tool developed by Google.
   * Consists of two main components: Overview and Dive.
3. **Facets Overview:**
   * Screenshot example from the UCI census data showcasing Facets Overview.
   * Provides quick insights into feature distribution and common data issues.
   * Allows comparison between multiple datasets, highlighting discrepancies and issues such as unexpected feature values and missing values.
4. **Facets Overview Examples:**
   * Visualization of numeric features (e.g., capital gain and capital loss) highlighting non-uniformity and potential issues (e.g., high percentage of zero values).
   * Visualization of categorical features (e.g., target feature representing salary) highlighting discrepancies between training and test datasets.
5. **Facets Dive:**
   * Introduction to Facets Dive, offering an intuitive interface for exploring relationships between data points across different features.
   * Customizable features such as position, color, and visual representation of each data point based on its feature values.
6. **Facets Dive Examples:**
   * Animation demonstrating exploration of UCI Census test dataset by coloring data points based on a feature (e.g., relationship) and faceting by age and marital status.
   * Example from a research-based image dataset showcasing confusion matrix view to analyze misclassifications.
   * Identification of incorrectly labeled data through visual inspection, enhancing model evaluation and data understanding.
7. **Conclusion:**
   * Importance of tools like Facets in discovering insights and improving machine learning models' accuracy and inclusivity.