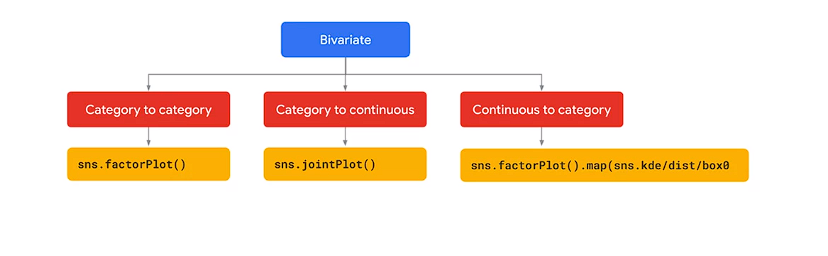
1. **Definition of EDA:**
   * EDA is an approach to analyzing data sets to summarize their main characteristics, often using visual methods.
   * It involves using graphics and basic sample statistics such as mean, median, or standard deviation to understand what information might be obtainable from the dataset.
2. **Purpose of EDA:**
   * EDA allows analysts to quickly look at data for trends, outliers, and patterns.
   * The goal is to **obtain theories** that can later be tested in the modeling step.
3. **Techniques Used in EDA:**
   * EDA employs a variety of techniques, mostly graphical, to maximize insight into a dataset.
   * Techniques include scatter plots, box plots, histograms, etc.
4. **Comparison with Other Data Analysis Approaches:**
   * Three popular data analysis approaches are classical analysis, Exploratory Data Analysis, and Bayesian analysis.
   * . These three approaches are similar in that they all start with a general science engineering problem, and all yield science engineering conclusions.
   * While all three start with a general problem and yield conclusions, they differ in the sequence and focus of the intermediate steps.
5. **Classical Analysis:**
   * In classical analysis, data collection is followed by the imposition of a model, and analysis focuses on the parameters of that model.
6. **Exploratory Data Analysis (EDA):**
   * In EDA, data collection is not followed by a model imposition. Analysis is done immediately to infer what model would be appropriate.
   * EDA allows the data to suggest admissible models that best fit the data.
7. **Bayesian Analysis:**
   * In Bayesian analysis, analysts attempt to answer research questions about unknown parameters using probability statements based on prior data.
   * The purpose is to determine posterior probabilities based on prior probabilities and new information.
8. **Application in Real-World Data Analysis:**
   * Real-world data analysts often mix elements of all three approaches and others.
   * The distinctions between the approaches were made to emphasize their major differences.
9. **EDA Techniques:**
   * EDA techniques are generally graphical and include scatter plots, box plots, histograms, etc.

Posterior probabilities is the probability, and event will happen after all evidence, or background information has been taken into account. Prior probability is the probability and event will happen before you taken any new evidence into account.

1. **Integration of Approaches:**
   * Real-world data analysts freely mix elements of all three approaches and other approaches as well.
   * The distinctions were made to emphasize the major differences among the three approaches.
2. **EDA Approach:**
   * EDA does not impose deterministic or probabilistic models on the data. Instead, it allows the data to suggest admissible models that best fit the data.
3. **Focus of EDA:**
   * In EDA, the focus is on the data, its structure, outliers, and models suggested by the data.
4. **Methods Used in EDA:**
   * **Univariate Analysis:**
     + This is the simplest form of analyzing data, where the data has only one variable.
     + It doesn't deal with causes or relationships like regression; its purpose is to describe and summarize the data and find patterns.
     + For categorical features, numerical EDA can be performed using pandas' crosstab function and visual EDA using Seaborn's countplot function.
     + For continuous features, numerical EDA can be performed using pandas' describe function, and visualization can be done using boxplots, distribution plots, and kernel density estimation (KDE) plots using Matplotlib or Seaborn.



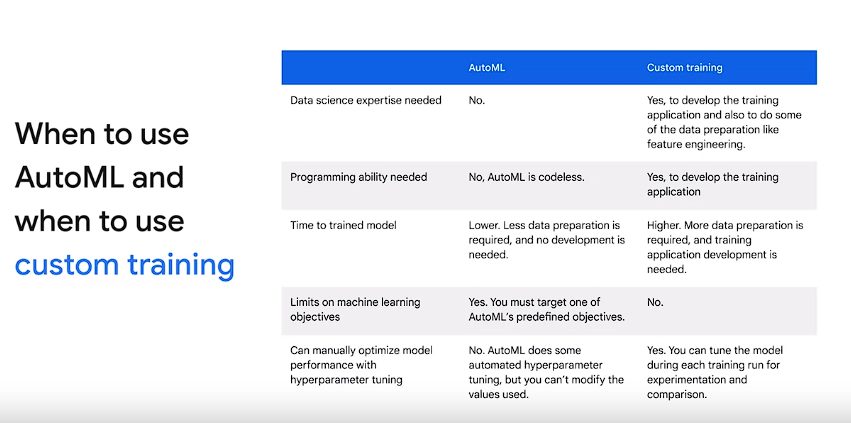
* + **Bivariate Analysis:**
    - Bivariate analysis involves the analysis of two sets of values to find out if there is a relationship between them.
    - It usually involves variables x and y and is one of the simplest forms of statistical analysis.
    - Bivariate and multivariate data can be analyzed in Python using Matplotlib or Seaborn, among other tools.

1. **Visualization Tools:**
   * Seaborn is mentioned as a powerful tool for EDA, with features like conditional plots and the ability to easily build various types of plots to visualize relationships between variables.
   * Seaborn's factor plot method allows for drawing categorical plots on a facet grid, while the jointPlot function can draw plots of two variables with bivariate and univariate graphs.
   * Seaborn's regplot function can visualize linear relationships between two sets of features, as well as identify outliers present in the data.

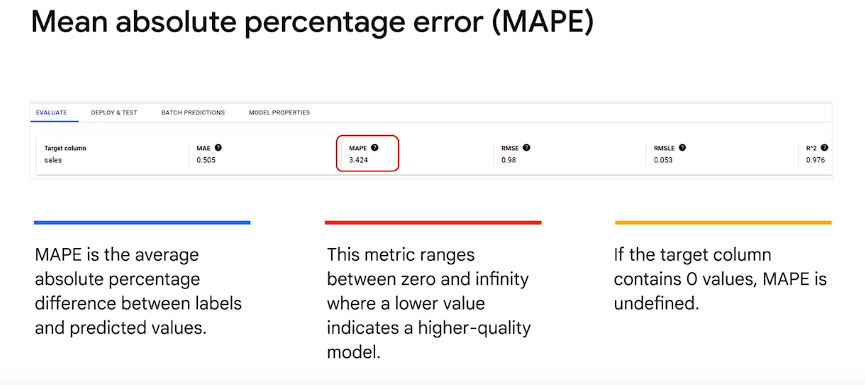
he purpose of Exploratory Data Analysis (EDA) is to find insights that will serve for data cleaning preparation or transformation, ultimately used in a machine learning algorithm. Data analysis and visualization are used at every step of the machine learning process, including data exploration, data cleaning, model building, and presenting results.

Here's a breakdown of the key points mentioned:

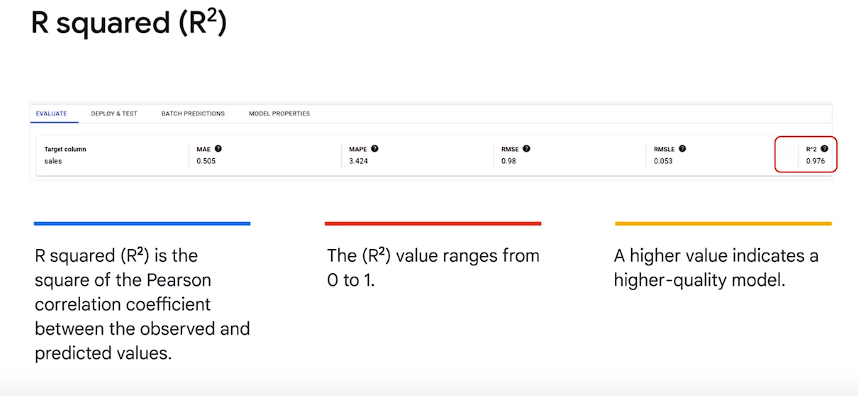
1. **Purpose of EDA:**
   * To find insights that aid in data cleaning, preparation, or transformation for machine learning algorithms.
2. **Use of Data Analysis and Visualization:**
   * Data analysis and visualization are integral at every step of the machine learning process.
   * These steps, including data exploration, data cleaning, model building, and presenting results, can all be part of one notebook.
3. **Examples of Visualization Techniques:**
   * **Histograms:** Graphical display of data using bars of different heights to show the distribution and spread of continuous sample data.
   * **Scatter Plots:** Graph where values of two variables are plotted against two axes to reveal any correlation present.
   * **Heat Maps:** Graphical representation of data using color coding to represent different values, commonly used **to visualize correlations** between features in a dataset.
4. **Importance of EDA:**
   * EDA helps gain maximum insight into the dataset and its underlying structure.
   * It helps identify outliers or anomalies in the data.
   * It is crucial for identifying the most influential features in the dataset.
5. **Summary:**
   * Data analysis, the second step in the ML pipeline, is crucial for preparing the data before model training.
   * The purpose of EDA is to gain insight into the dataset, identify outliers, anomalies, and influential features.
   * There are many more ways to explore, analyze, and plot data, which can be explored to enhance understanding and insights.
6. The team at XYZ Company has defined a business use case and wants to deliver an ML model to production, their first ML project.
7. The team consists of a software developer, a data analyst, and a data scientist.
8. The goal is to predict the consumer spending score and determine which variable contributes most to the spending score.
9. The company has structured data and comma-separated value files, unstructured data with texts and images, and streaming data from sensors, but the business case is to predict a credit score, which is structured data.
10. The team will use Vertex AI's AutoML, which provides a point-and-click solution for training, evaluating, and deploying machine learning models.
11. Vertex AI offers features and services in a dashboard format for various tasks in the machine learning pipeline.
12. The team will use Tabular data type and regression classification as their objective in Vertex AI's AutoML.
13. AutoML tables automatically defines the problem and model based on the data type of the target column.
14. The team uploads the dataset into Vertex AI's AutoML, where the data analyst uploads the file from her laptop.
15. AutoML supports various data types such as image, tabular, text, and video, but the team requires a tabular structured dataset.
16. AutoML tables automatically selects the model type based on the data type of the target column; regression models predict numerical values, while classification models predict categories.
17. AutoML simplifies the process by performing automated hyperparameter tuning, eliminating the need for manual modification.
18. AutoML is codeless, allowing users to create and train models with minimal technical effort.
19. Custom training in Vertex AI requires data science expertise for developing the training application and data preparation.
20. AutoML saves time by requiring less data preparation and development compared to custom training.
21. AutoML targets predefined objectives such as regression, classification, and forecasting, while custom training offers more flexibility in objective selection.
22. AutoML and custom training have limits on managed datasets, and data size limitations vary depending on the type of datasets.



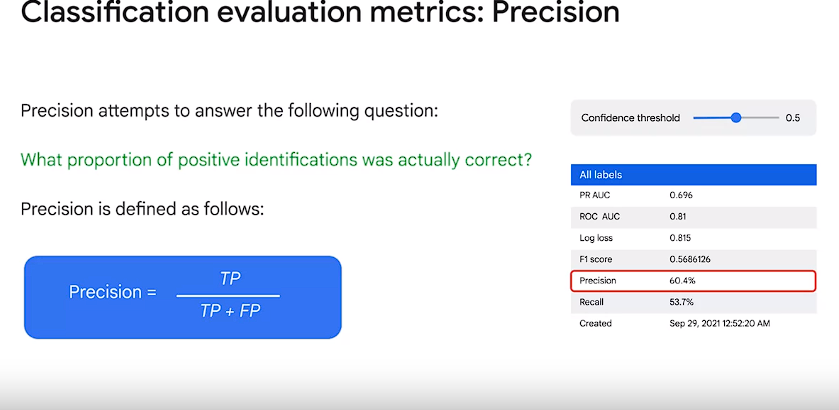
1. For unmanaged datasets, such as those from Google Cloud Storage or BigQuery, there is no limit on data size.
2. Vertex AI manages various stages in the ML workflow, including dataset creation, model training, evaluation, hyperparameter tuning (custom training only), model storage, deployment, and more.
3. Model evaluation metrics provide quantitative measurements of how the model performed on the test set.
4. Interpretation and usage of metrics depend on business needs and the problem the model is trained to solve.
5. AutoML tables split the dataset into training, validation, and testing sets by default, with 80% for training, 10% for validation, and 10% for testing.
6. The training set is used for learning model parameters, the validation set for tuning hyperparameters, and the test set for evaluating model performance on new data.
7. Model evaluation metrics for regression include Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), Root Mean Squared Logarithmic Error (RMSLE), and R-squared.
8. Mean Absolute Error (MAE) measures the average magnitude of errors without considering direction, while MAPE measures the average absolute percentage difference between labels and predictions.

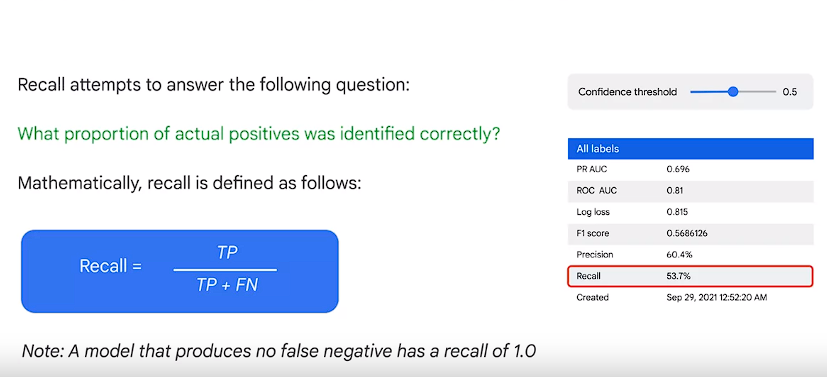


1. RMSE is more sensitive to outliers than MAE and penalizes large errors more heavily.
2. RMSLE penalizes underpredictions more heavily than overpredictions and is useful when not penalizing large differences more heavily than small ones.
3. R-squared indicates the proportion of variance in the dependent variable that is predictable from the independent variables.(0-1)



1. Model feature attributions show the impact of each feature on the model, aiding in understanding feature importance.
2. Classification metrics include PRAUC, ROC AUC, log loss(cross entropy between model prediction and target value[0->infinity]) , F1 score(harmonic mean of preci and recall), precision, recall, and confusion matrix.
3. Precision and recall help understand how well the model captures information and what it leaves out, with F1 score providing a balance between the two.





1. Confusion matrix assesses the accuracy of a predictive model for classification tasks.
2. Evaluation metrics can be accessed via web UI, REST, command line, or Python.
3. Testing the model with new data and deploying it via endpoints for batch or online predictions are essential steps before using the model in production.
4. AutoML provides solutions for various tasks such as vision and natural language processing.
5. The team at XYZ company has defined a business use case and established success criteria for delivering an ML model to production, which will be their second project.
6. The team consists of a software developer proficient in Java, a data analyst knowledgeable in SQL but lacking ML expertise, and a data scientist with domain knowledge but limited experience in putting ML models into production.
7. The team previously deployed a successful model into production using Vertex AI AutoML and now seeks the flexibility of a custom model.
8. They need to choose between continuing with Vertex AI AutoML and improving their coding skills, using a custom-based Python framework, or considering BigQuery ML.
9. Their data requirements include structured data larger than 100 gigabytes, potentially containing fewer than 1,000 rows.
10. Vertex AI AutoML has limitations on data set size and minimum row requirements, making it unsuitable for their current project.
11. BigQuery ML offers an easy-to-use method for invoking ML models on structured data using SQL.
12. Custom modeling without BigQuery ML introduces complexity and requires multiple tools, slowing down the process.
13. BigQuery ML accelerates time to production, simplifies development, and automates several steps in the ML workflow.
14. It streamlines tasks such as importing and preprocessing data, model building, and deployment.
15. BigQuery ML serves as a middle ground between using pretrained models and building custom TensorFlow models.
16. Working with BigQuery ML involves four major steps: extracting training data with SQL, creating and evaluating the model, and predicting using the model.
17. Since the team's data resides in BigQuery and a member is proficient in SQL, development will be faster.
18. BigQuery ML automates common ML tasks and performs hyperparameter tuning.
19. An example use case involves building a model within BigQuery using a public dataset of taxi rides from New York City to predict taxi fares.
20. The process involves selecting relevant data, creating a model with a few lines of code, evaluating it, and serving predictions.
21. Training job metrics are available for monitoring the performance of the model.

**BigQuery Machine Learning supported models**

1. **Logistic Regression:**
   * **Binary Logistic Regression: Suitable for binary classification problems where the label is true/false or 1/0. For instance, determining if a flight will be late.**
   * **Multi-class Logistic Regression: Applicable when the label consists of multiple categories, such as classifying emails into primary, social, promotions, updates, or forums tabs.**
2. **Linear Regression:**
   * **Used for regression problems where the label is a numerical value. For example, forecasting product sales on a certain day.**
3. **TensorFlow-based Deep Neural Networks (DNN):**
   * **DNNRegressor: Used for regression problems.**
   * **DNNClassifier: Suitable for both binary and multi-class classification problems.**
4. **Boosted Decision Trees:**
   * **Boosted Tree Regressor: For regression tasks.**
   * **Boosted Tree Classifier: Used for binary and multi-class classification problems. Boosted decision trees tend to outperform regular decision trees, especially on extensive datasets.**
5. **Matrix Factorization:**
   * **Used for creating recommendation systems. For example, recommending the next product for a customer based on past purchases, historical behavior, and product ratings.**
6. **k-means Clustering:**
   * **Applied when labels are unavailable, typically for customer segmentation.**
7. **BigQuery ML for Time Series:**
   * **Popular for estimating future demands, such as retail sales or manufacturing production forecasts. It automatically detects and corrects for anomalies, seasonality, and holiday effects.**
8. **AutoML Tables:**
   * **Suitable for regression, classification, and time series forecasting problems. It automatically searches through various models to find the best one for the given problem.**
9. **TensorFlow Model Importing:** 
   * **If you have previously trained TensorFlow models and want to import them to BigQuery for performing predictions.**

**These options provide flexibility for addressing various machine learning tasks within the BigQuery environment.**

**Certainly, here are all the points extracted from the passage:**

1. **Hyperparameter tuning in machine learning identifies optimal hyperparameters for a learning algorithm.**
2. **Hyperparameters are set before the learning process begins, unlike parameters learned from the data.**
3. **It allows spending less time manually iterating hyperparameters and more time exploring insights from data.**
4. **This hyperparameter tuning feature is made possible in BigQuery ML by using Vertex Vizier.**
5. **In this lesson, we examined BigQuery ML hyperparameter tuning options.**
6. **BigQuery ML supports hyperparameter tuning when training ML models using create model statements.**
7. **Hyperparameter tuning is commonly used to improve model performance by searching for the optimal hyperparameters.**
8. **Hyperparameter tuning supports the following model types: Linear regression, logistic regression, k-means, matrix factorization, boosted tree classifier, boosted tree regressor, DNN classifier, and DNN regressor.**
9. **In an example of hyperparameter tuning, a simple deep neural network (DNN) has three standard hyperparameter options: DATA\_SPLIT\_METHOD set to RANDOM, EARLY\_STOP set to TRUE, HIDDEN\_UNITS set to 30, 50.**
10. **In the second example, additional hyperparameters are added or modified to improve model performance, such as NUM\_TRIALS, MAX\_PARALLEL\_TRIALS, DROPOUT HPARAM range, OPTIMIZER, and LEARN\_RATE HPARAM\_CANDIDATES.**
11. **The ROC AUC curve is an evaluation metric for binary classification problems.**
12. **It plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold values.**
13. **The AUC measures the ability of a classifier to distinguish between classes, with higher values indicating better performance.**
14. **In the example provided, a DNN model without hyperparameter tuning has an ROC AUC of 0.53.**
15. **After hyperparameter tuning, the ROC AUC improves to 0.79, indicating enhanced model performance.**

**These points cover the key concepts and examples related to hyperparameter tuning in machine learning, specifically within the context of BigQuery ML.**

1. **Recommendation systems are machine learning systems that help users discover new products and services.**
2. **Recommendation systems are all about personalization and have a number of benefits in terms of user engagement, up selling and cross selling.**
3. **The pressure, especially for a retail business in today's age to deliver recommendations at the right time and in the right context has never been greater.**
4. **Companies must be able to train and implement recommendation systems quickly and effectively to improve conversions, increase click-through rates, and build customer satisfaction, loyalty, and brand affinity.**
5. **One common use case is to prepare training data in BigQuery.**
6. **Train a recommendation system with BigQuery ML and use the predicted recommendations in production.**
7. **Feedback can take many forms, such as using stars, which is direct or explicit feedback, or inferring interest indirectly, which is called implicit feedback.**
8. **Implicit feedback tends to be more common.**
9. **Data includes visitor ID, item ID, and session duration.**
10. **Selecting the data straightforward here, select from the view.**
11. **With the training data, create the model specifying model name, model type (matrix factoring for recommendation systems), user column, item column, rating column, and set feedback type to implicit.**
12. **After training is complete, evaluate the model using select from ML evaluate.**
13. **Average rank or mean percentile rank is a commonly used metric for implicit matrix factoring.**
14. **Lower mean rank indicates closer predicted recommendations match the behavior and the test data.**
15. **Make predictions with BigQuery ML on a single user by running select from ML recommend.**
16. **Specify the model created and the user ID(s) for which you want to make predictions.**
17. **Sort by descending predicted confidence and limit end to get top recommendations.**
18. **Export recommendations for ad redirection campaigns with Google Analytics.**
19. **Create a new column for likelihood of purchase based on expected recommendations.**
20. **Scale the time spent viewer between zero and one for each user as a proxy for likelihood to purchase.**
21. **Reimport predictions into Google Analytics to create new campaigns for those products.**
22. **Connect the intended recommendations with your customer relationship management system (CRM).**
23. **Create targeted email campaigns to deliver relevant products directly to the inbox.**

**Top of Form**