ML answer questions that are too complex to answer through manual analysis

- Pieces of code that help people explore, analyze, and find meaning in complex data sets.
- Each algorithm is a finite set of unambiguous step-by-step instructions that a machine can follow to achieve a certain goal.
- In a machine learning model, the goal is to establish or discover patterns that people can use to make predictions or categorize information.
    - Machine learning algorithms use parameters that are based on training data—a subset of data that represents the larger set.
        - As the training data expands to represent the world more realistically, the algorithm calculates more accurate results.

aRt Verse
ART ∔ SCIENCE ∔ NATURE

Compiled by Dr T Oberholster

AI-900 version: October 2021

ML Algorithms

- **Supervised Learning:** Historical data points with known labels. You have an idea of the final outcome.
    - **Regression:** y is numerical
        - How much/many?
    - **Classification:** y is categorical
        - Is this a/b?
- **Unsupervised Learning:** Unlabelled historical data points. You do not know the outcome.
    - **Clustering:** groups measurements based on shared features
        - What is similar/same/different?

Math & Logic

Models & uses

ML Studio

Azure Machine Learning Designer

Modelling Steps

Create a workspace & compute resource → Explore data → Train model → Evaluate model → Create an inference pipeline → Deploy model

ML Model Lifecycle

Data Acquisition → Business understanding → Modeling → Testing → Deploy (& integrate / automate) → Monitor

Model Building

Web portal in Azure ML for project authoring

- Notebooks
    - Write and run code in managed Jupyter Notebook servers.
- Azure ML designer
    - Train and deploy machine learning models without writing any code.
    - Drag and drop datasets and components to create ML pipelines.
- Azure AutoML GUI
    - Create automated ML experiments with an easy-to-use interface.
- Data labellling:
    - Efficiently coordinate image labeling or text labeling projects.

Components & Assets

- Environment
    - An encapsulation of the environment for training or scoring ML models.
- Experiment
    - A grouping of many runs from a specified script for a workspace.
- Pipeline
    - Create and manage workflows that stitch together machine learning phases.
- Compute resources
    - A machine or set of machines you use to run your training script or host your service deployment
        - Compute instance: VM with multiple ML tools and environments
        - Compute cluster: Cluster of VMs with multi-node scaling capabilities

Components & Assets

- Datasets
    - A reference to the data source location and copy of metadata.
- Datastores
    - Datasets use datastores to securely connect to Azure storage services.
- Models
    - A piece of code that takes an input and produces output
- Endpoints
    - An instantiation of your model into a web service that can be hosted in the cloud.
        - Web service endpoint
        - Real-time endpoint
        - Pipeline endpoint

Step 1: Create a workspace

- Via the Azure portal
  - Provide subscription, resource group, workspace name, region, storage account, key vault, application insight and container registry info
- Sign into Azure Machine Learning Studio using Azure directory, subscription & workspace
  - Manage assets and resources via workspace

Step 2: Create a compute resource

- **Compute targets:** cloud-based resources on which to run models and do data exploration
  - **Compute Instances:** Development workstations that data scientists can use to work with data and models
  - **Computer Clusters:** Scalable clusters of virtual machines for on-demand processing of experiment code
  - **Inference Clusters:** Deployment targets for predictive services on trained models
  - **Attached Compute:** Link to existing resources (VMs and DBs)
- Compute Instances tab
  - Add new compute: provide compute name, VM type and VM size
  - Create compute cluster (while compute instance is pending): provide location, VM priority, VM type, VM size, compute name, min & max nodes, idle seconds, SSH access

Step 3: Explore the Data

- Create a pipeline via Azure ML Studio Designer page
  - Provide: Pipeline name (and date)
  - Select compute target in Settings pane
- Add & explore dataset
  - Drag dataset from **Sample dataset** section (>> to expand panel) onto the canvas (drag-and-drop visual interface)
  - Right-click on dataset, select **Dataset output** (preview data graph icon) on **Output** menu
  - Review schema and distribution histograms
  - Select column header for more details
  - Exclude datasets with high % missingness
  - Close results visualisation window
- Data transformations
  - Expand Data Transformations pane on the left
  - Drag **Select Columns in Dataset** into canvas and connect with Dataset
  - Click into **Select Columns in Dataset**, select **Edit column** & select columns by name using +
  - Add other model and connect them (Clean Missing Data with columns selected, Normalize Data module & transformation method MinMax)
    - Normalization to same scale prevents numeric columns with large values dominating the model outputs
- Run the pipeline
  - Applies data transformations
- View transformed data
  - Select last model & in Settings pane select **Outputs + logs** tab > Visualise
  - Close **results visualisation** window

Regression specific comments

No comments

Classification specific comments

Predicted/Target column has two values (0 or 1), one for each outcome

Clustering specific comments

No comments

Step 4: Train the model

- Create and run training pipeline
  - Train on a subset, rest used to test to compare actual vs predicted labels (model evaluation)
- Data transformation > Split Data & connect to Normalized Data output (left connector)
  - Splitting mode, Fraction of rows, random seed, stratified split
- Model Training > Train Model & connect right input to left output of Split Data
  - Label column

Regression specific comments

- Machine Learning Algorithm > *Regression* > *Linear Regression module* & connect to left input of Train Model
- Testing is done by scoring the validation dataset. Model Scoring & Evaluation > Score Model & connect left input with Train Model output, connect right input with right output from Split Data (carry data and model forward)
- Select Submit to run the pipeline

- Score Model > Outputs & logs > Data outputs > Scored dataset > Preview Data icon
  - Scored labels = predicted values
  - Close results visualisation

Classification specific comments

- Machine Learning Algorithm > *Classification* > *Two-Class Logistic Regression* & connect to left input of Train Model
- Testing is done by scoring the validation dataset. Model Scoring & Evaluation > Score Model & connect left input with Train Model output, connect right input with right output from Split Data (carry data and model forward)
- Select Submit to run the pipeline

- Score Model > Outputs & logs > Data outputs > Scored dataset > Preview Data icon
- Scored labels = predicted values
  - *Scored probabilities between 0 and 1, indicates a positive prediction.*
  - *Probabilities greater than 0.5 indicate 1, and probabilities less than 0.5 indicate 0.*
- Close results visualisation

Clustering specific comments

- Machine Learning Algorithm > *Clustering* > *K-Means Clustering* & connect to left input of Train Model
-
  - *K = number of clusters to create*
    - *Set Number of centroids*
  - *Measurements are treated as multidimensional vectors*
  - *The algorithm initialises K coordinates at randomly selected points (centroids) in n-dimensional space (n number of dimensions in feature vector)*
    - *Feature points are plotted on same dimensional space and each point is assigned to the closest centroid.*
    - *Centroids are moved to the middle of the assigned points (mean distance)*
    - *Reassign points*
    - *Repeat until cluster allocations stabilized or a number of iterations is completed*

- Score Model > Outputs & logs > Data outputs > Scored dataset > Preview Data icon
  - *Assignments column = clusters (0, 1, 2) to which each observation (row) is assigned.*
  - *New columns with distances of assigned points from cluster centers.*
- Close results visualisation

Step 5: Evaluate the model

- Model Scoring & Evaluation > Evaluate Model & connect left input with Score Model
- Select Submit

Regression specific comments

- Compare predicted and actual labels in validation dataset.
- Evaluate Model > Outputs & logs > Data outputs > Evaluation results > Preview Data icon
    - *Mean Absolute Error (MAE): Average difference between actual and predicted values. Lower ~ better model performance*
    - *Root Mean Squared Error (RMSE): Square root of mean squared difference between actual and predicted values. Large values = large variance in residuals*
    - *Relative Squared Error (RSE): Relative squared difference between 0 and 1 of predicted and actual values. Closer to zero ~ better model performance, use for different unit labels*
    - *Relative Absolute Error (RAE): Absolute difference between 0 and 1 of predicted and actual values. Closer to zero ~ better model performance, use for different unit labels*
    - *Coefficient of Determination (R2): R-squared, how much variance between actual and predicted explained by model. Closer to one ~ better model performance*
- Close results visualisation
- More than one regression model can be compared & connected to the evaluation module.

Classification specific comments

- Compare predicted and actual labels in validation dataset.
- Evaluate Model > Outputs & logs > Data outputs > Evaluation results > Preview Data icon
    - *Confusion matrix: Tabulation of predicted and actual value counts for each class. Binary classification models predict 2 values (0 or 1).*

Confusion Matrix		Predicted 1	Predicted 0
Actual	1	True positive	False positive
Actual	0	False negative	True negative

True positive and True negative values should contain the most counts

-
    - *Accuracy: Ratio of correct predictions (true positive + true negative) to the total number of predictions. How much of the model predictions were correct.*
    - *Precision: Fraction of correctly identified positive cases (True positive / True + False positives). How much of the positive predictions were correct.*
    - *Recall: Fraction of positive cases which are positive (True positives / True positives + False negatives). True positive rate. How much of the actual positives did the model identify.*
    - *F1 Score: Combines Precision and Recall*
    - *Threshold: Slider changes probability scores, moving it to the 0 (Recall = 1) and to 1 (Recall = 0)*

Clustering specific comments

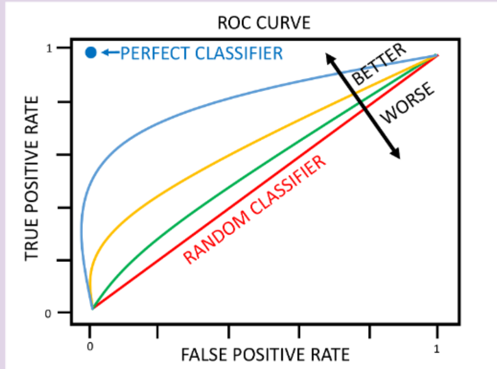
- Evaluate Model > Outputs & logs > Data outputs > Evaluation results > Preview Data icon to view cluster separation metrics
    - *Average Distance to Other Center: Average closeness of each point to all other cluster centers.*
    - *Average Distance to Cluster Center: Average closeness of each point to same cluster center.*
    - *Number of points: assigned to each cluster*
    - *Maximal Distance to Cluster Center: Max distance between each point and cluster centroid. High numbers = high cluster dispersion.*
- *Compare Average Distance to Cluster Center to Maximal Distance to Cluster Center to determine cluster spread*
- Close results visualisation

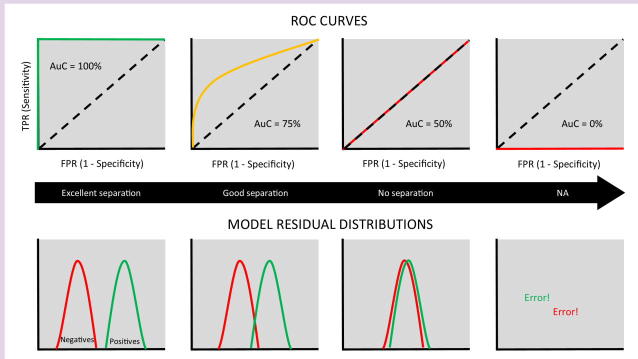
Regression specific comments

No comments

Classification specific comments

- *ROC curve:* *Received Operator Characteristic, measures True positive rate (recall) against False positive rate. Large area under the curve = better model performance (AUC). AUC of 0.5 = random change of being correct.*





- Close results visualisation
- More than one classification model can be compared & connected to the evaluation module.

Clustering specific comments

No comments

Step 6: Create an inference pipeline

- Perform data transformations on new data & use training model to infer/predict new labels
- Real-time inference pipeline > duplicates training pipeline
  - Rename pipeline
  - Provide Web Service/Enter Data Manually (csv) input for new data & connect to Select Columns in Dataset
  - Remove original raw dataset
  - Remove Evaluation Model module & replace with Execute Python Script (connect left input to Score Model)
    - Replace default values in script with model column names
  - Connect left Execute Python Model output to Web Service Output module (*For Clustering: Assign Data to Cluster)*
- Submit to compute-cluster
  - Execute Python Script > Output & logs > Results dataset (*For Clustering: Assign Data to Clusters > Output & logs > Results dataset)*
  - Close results visualisation

Step 7: Deploy model as a predictive service

- Publish real-time inference pipeline as service for client application
- Deploy
  - Provide name, description & compute type (Azure Contained instance)
  - Test service. Endpoints > select pipeline > Consume
    - REST end point
    - Primary Key
- Create second browser instance of Azure ML Studio > Notebooks (Author) > My Files > New File
  - Provide file location, file name, file type & overwrite settings
  - Check compute instance is running
  - Paste pipeline run code (with one observation's data) into notebook with endpoint & primary key details from previous browser > Consume
- Run notebook & verify return value.