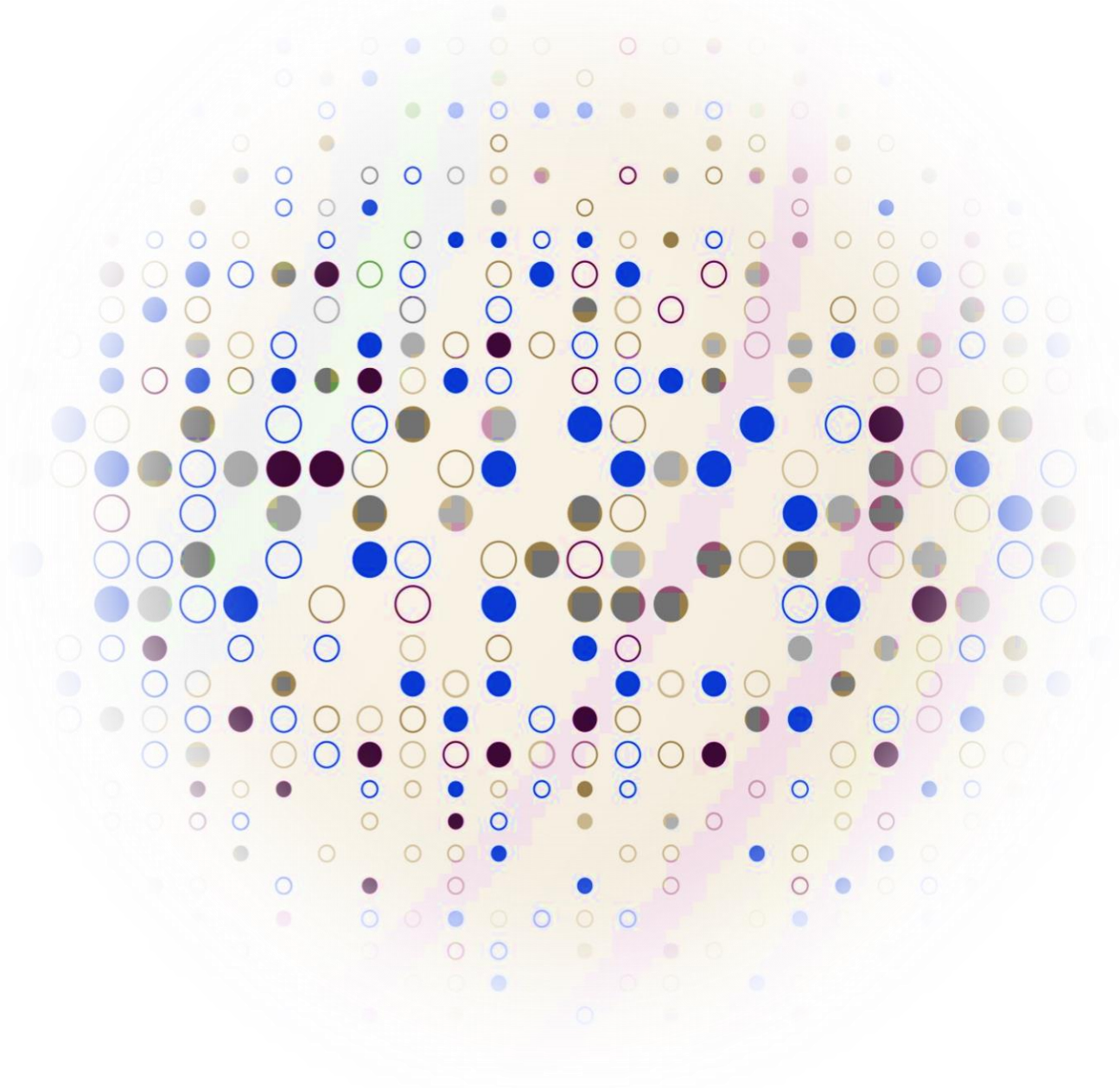
An Introduction to Machine Learning for Language Assessment: Transitioning from Regression

Vahid Aryadoust

National Institute of Education

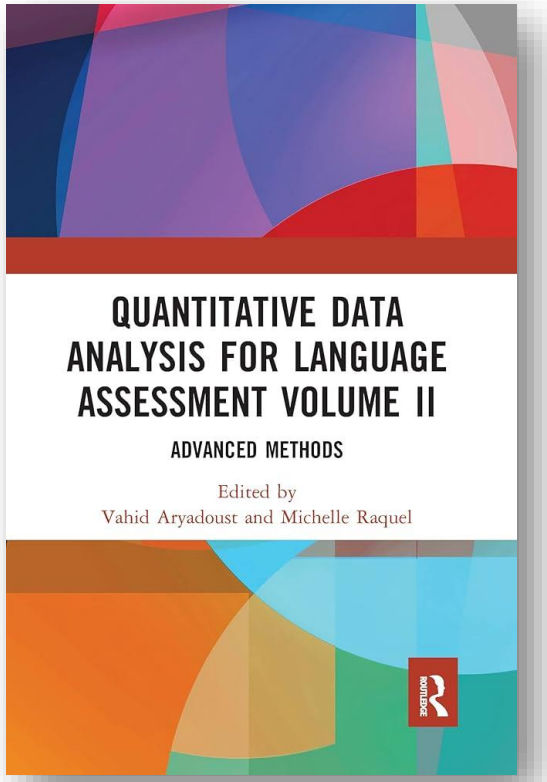
Nanyang Technological University

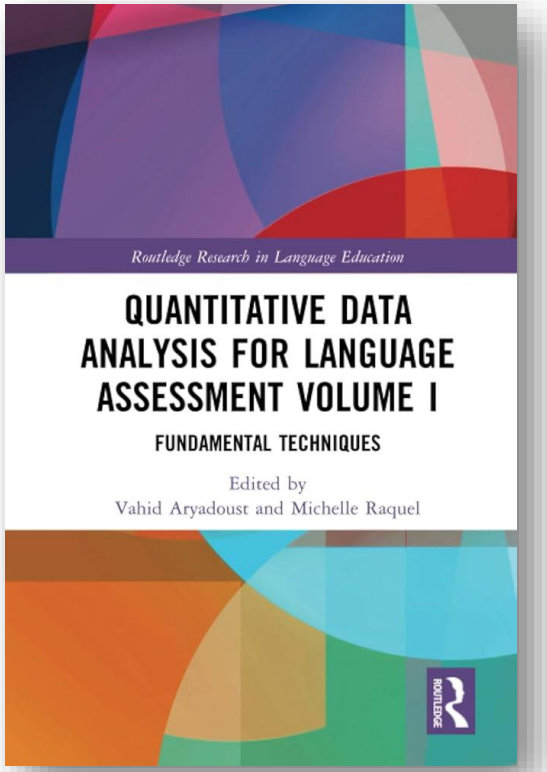
Singapore

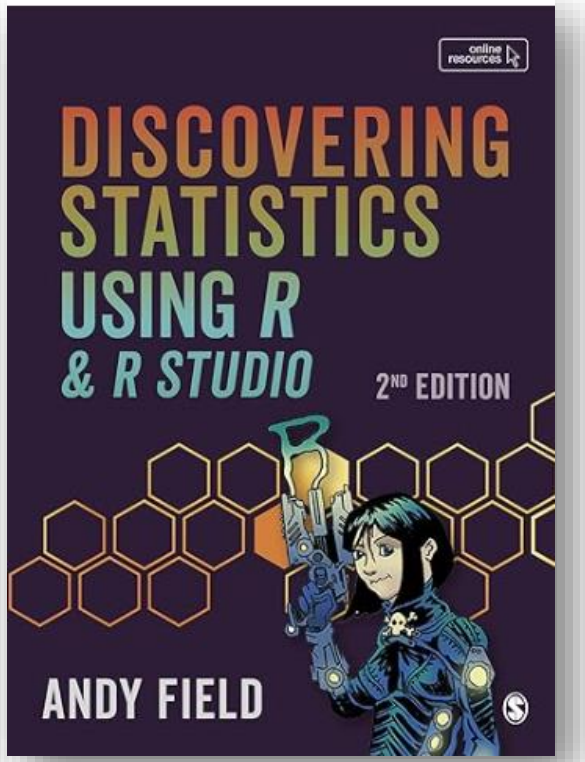


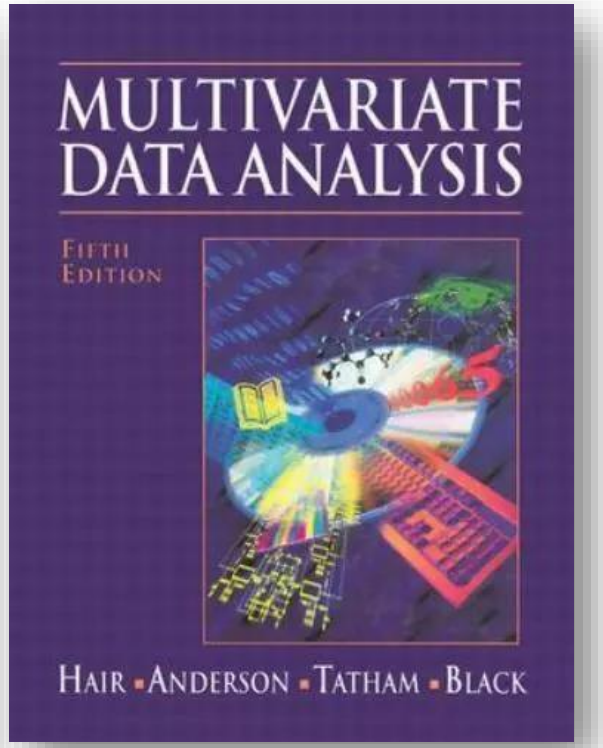
What is this workshop about?

- This **introductory** workshop aims to disseminate essential knowledge and tools for **predictive modeling** and **machine learning** in **language assessment**. Led by Dr. Vahid Aryadoust, this two-day online workshop comprises two parts. The **first** part focuses on the **general linear model**, specifically the linear regression model, providing a foundation in **statistical inference**. The **second** part delves into **machine learning**, offering some hands-on experience with **GUI** software and techniques for interpreting machine learning results. Each session will be **approximately two hours long**. Interested participants should register by scanning the barcode to receive necessary information. This workshop is supported by a grant from the **UK Association for Language Testing and Assessment (UKALTA)**.


QUANTITATIVE DATA ANALYSIS FOR LANGUAGE ASSESSMENT VOLUME II
ADVANCED METHODS
Edited by
Vahid Aryadoust and Michelle Raquel


Routledge Research in Language Education
QUANTITATIVE DATA ANALYSIS FOR LANGUAGE ASSESSMENT VOLUME I
FUNDAMENTAL TECHNIQUES
Edited by
Vahid Aryadoust and Michelle Raquel


online resources
DISCOVERING STATISTICS USING R & R STUDIO
2ND EDITION
ANDY FIELD

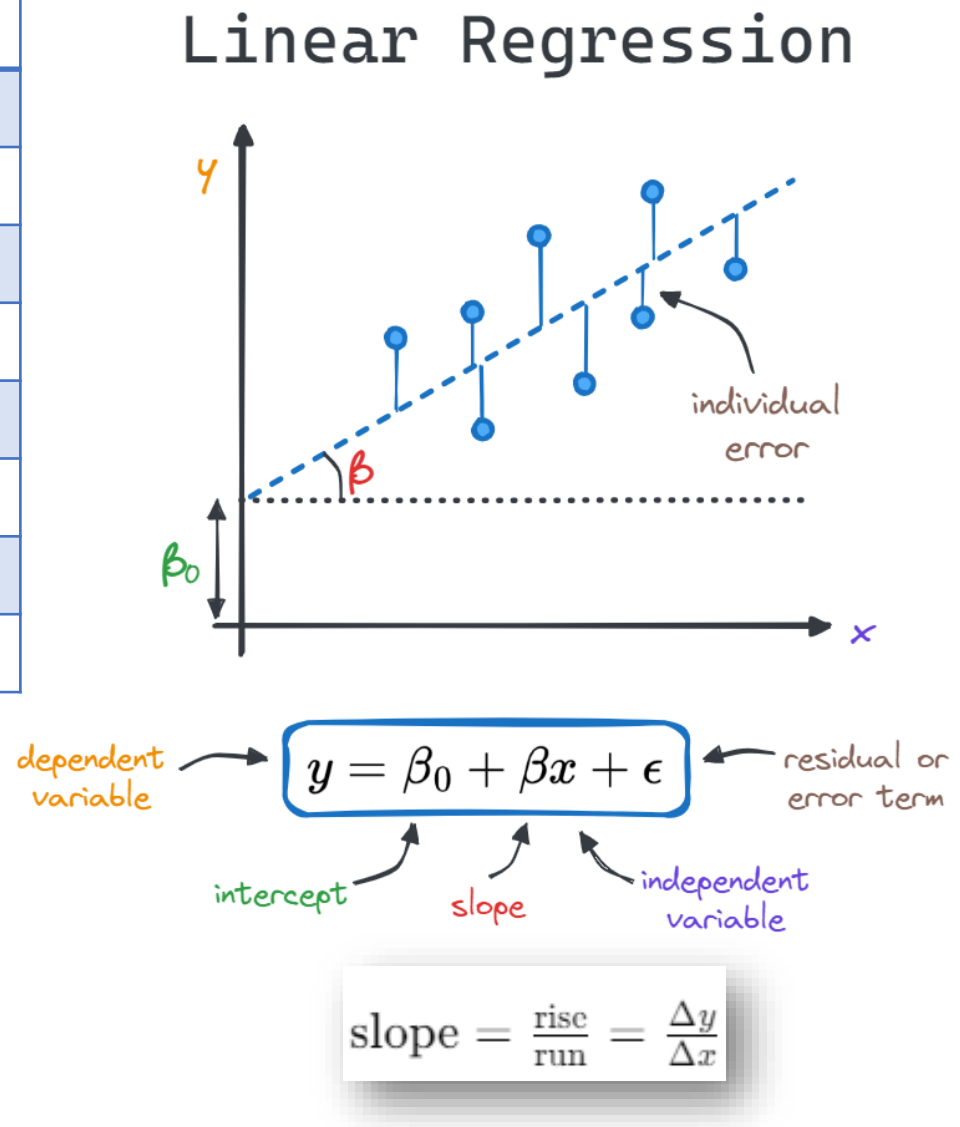

MULTIVARIATE DATA ANALYSIS
FIFTH EDITION
HAIR · ANDERSON · TATHAM · BLACK

Resources

General linear model (GLM)

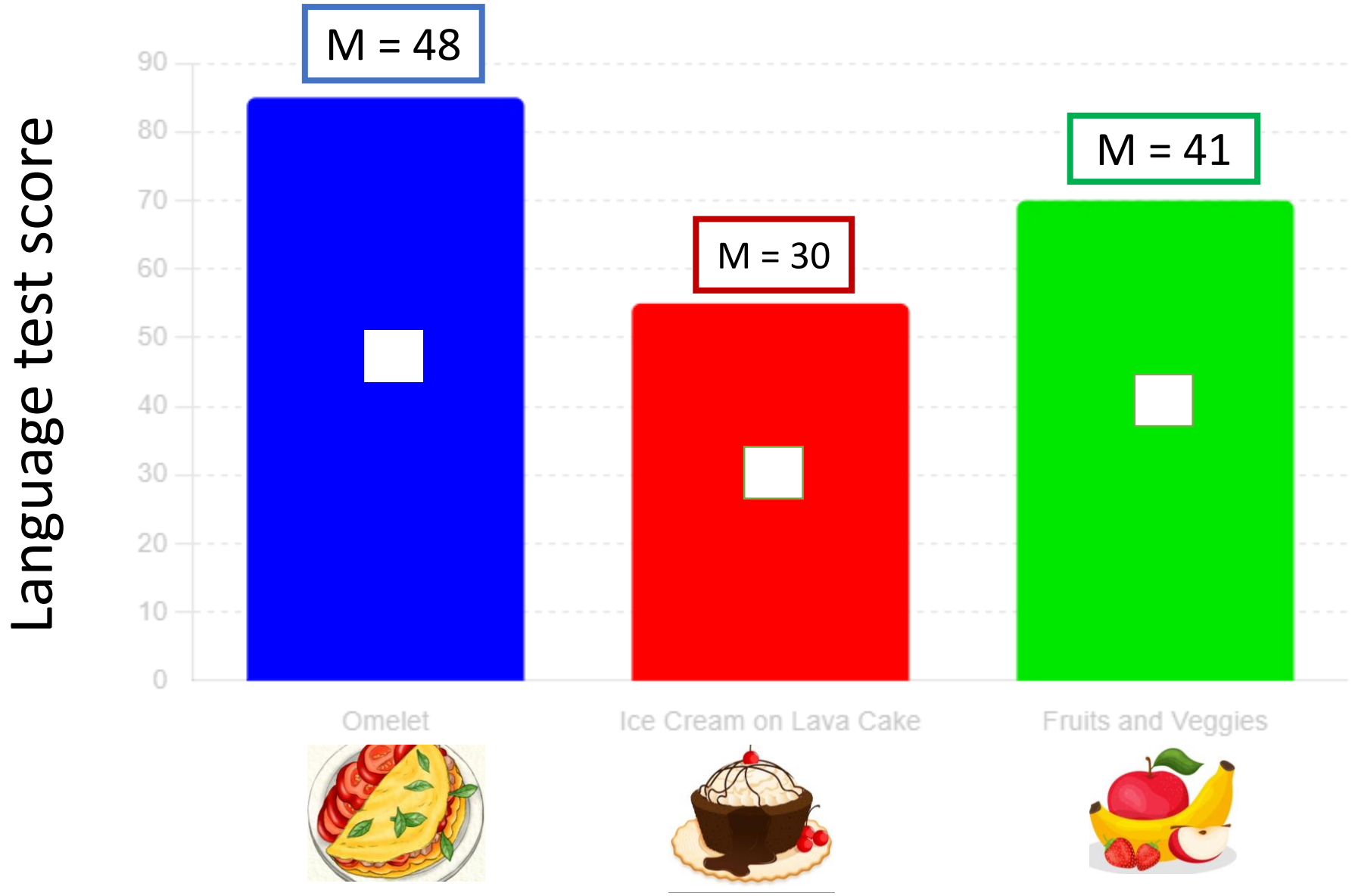
- A flexible statistical model that generalizes **multiple linear regression** to accommodate various types of independent variables.
- GLM incorporates **ANOVA, ANCOVA, MANOVA, MANCOVA, ordinary linear regression**, **t-test,** and **F-test**.

X	y	ŷ
3	6	
4	7	
5	4.5	
6	8.3	
7	5	
8	7	
9	9.3	
10	8	



Source: @data_science_learn • Linear Regression clearly explained Tech Threads • Threads

ANOVA


Language test score
90
80
70
60
50
40
30
20
10
0
M = 48
M = 30
M = 41
Omelet
Ice Cream on Lava Cake
Fruits and Veggies

GLM's mathematical notation

- $Y=\beta_0+\beta_1X_1+\beta_2X_2+...+\beta_nX_n+\varepsilon$, where:
- Y is the **dependent** variable.
- β_o is the **intercept**.
- $\beta_1,\beta_2,...,\beta_n$ are the coefficients for the **independent** variables $X_1, X_2,...,X_n$,
- X can be **continuous** or **categorical.**
- ε is the **error** term or **variability** in the dependent variable that is not explained by the linear combination of the predictors.

Standardized ±

$$Z_X = \frac{X - \bar{X}}{\sigma_X}$$

Unstandardized ±

What is multiple linear regression?

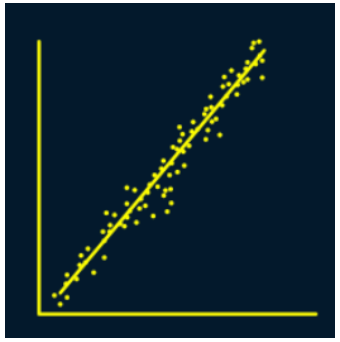
- A type of **GLM** used to model the relationship between one dependent (response) variable and two or more independent (predictor) variables.
- DV: **continuous**
- IVs: **continuous** or **categorical**
- $Y=\beta_0 +\beta_1 X_1 +\beta_2 X_2 +...+\beta_n X_n +\boldsymbol{\varepsilon}$

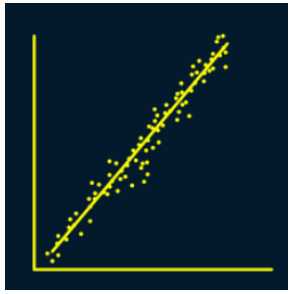
residuals

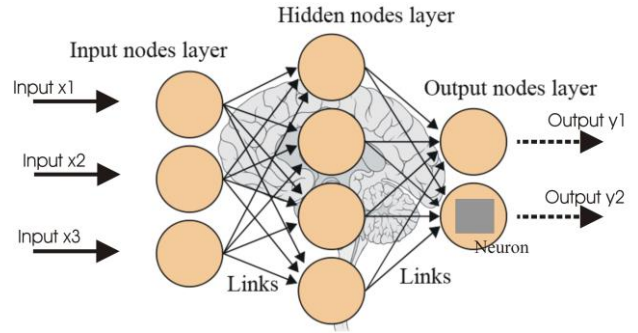
2 methods of doing multiple linear regression

- The **statistical model-based** method vs the **algorithm-based (machine learning)** method
- **Model-Based**:
  - The model is specified before the data analysis begins.
  - Assumptions (e.g., linearity, normality, homoscedasticity)
- **Algorithm-Based**:
  - Involve dividing data into training and testing datasets.
  - Assumptions (e.g., linearity, normality, homoscedasticity)

Our roadmap







Statistical linear regression

Linear regression like ML

Neural Network or Machine Learning

Multiple linear regression: The statistical model-based method

Assumptions of regression

- **Linearity**: The relationship between the dependent and independent variables is linear.
- **Independence**: Observations are independent of each other. (Durbin-Watson Test [0-4, with 2 indicating no autocorrelation] for longitudinal data)
- **Homoscedasticity (in ANOVA: homogeneity of variances)**: The variance of the error terms is constant across all levels of the independent variables.
- **Normality**: The residuals (errors) of the model are **normally** distributed.
- **No Multicollinearity**: Independent variables are not highly correlated with each other.

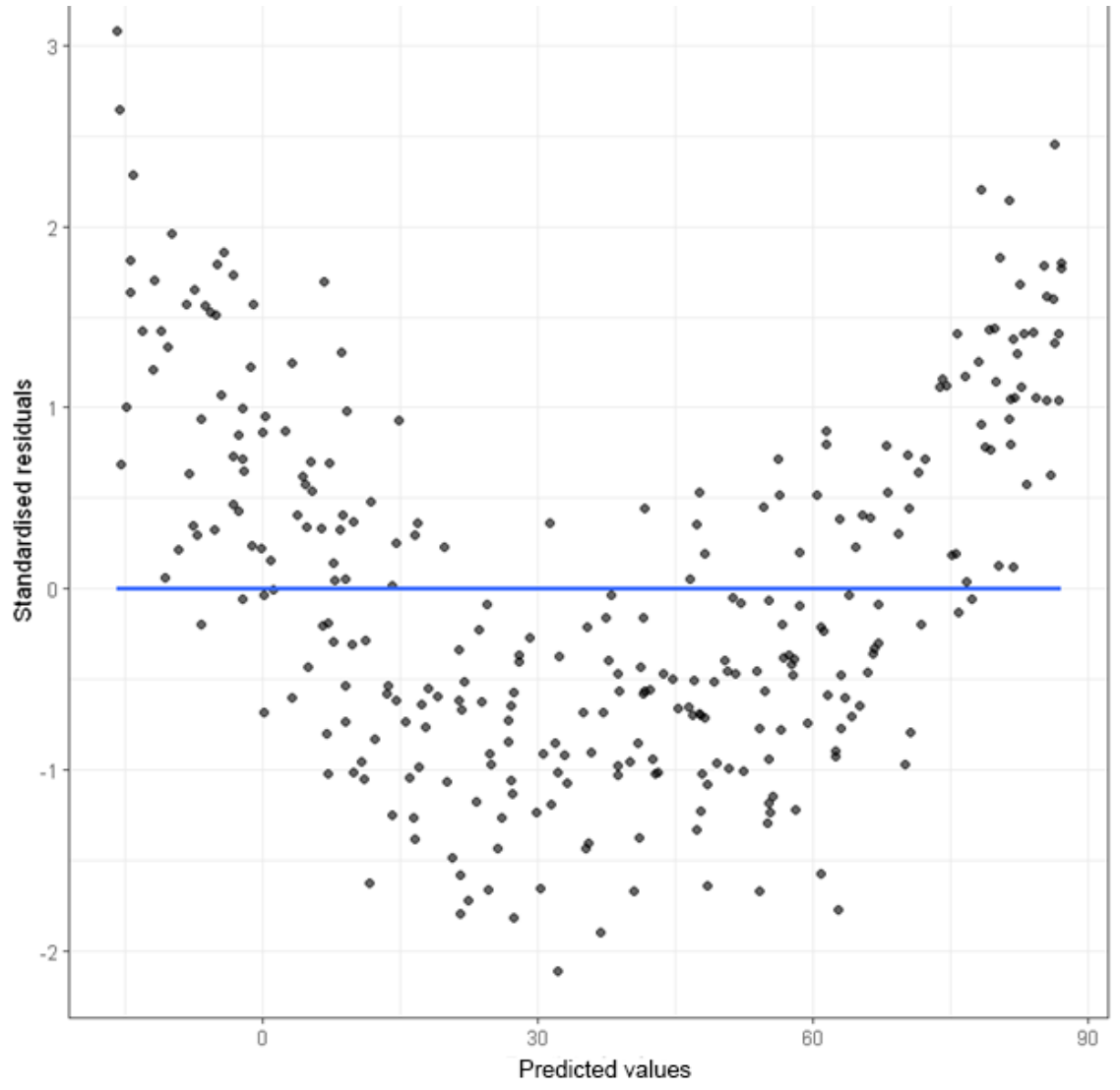
Assumptions of regression

Details	In-Text Citation
Linearity: Scatterplots	(Kutner, Nachtsheim, Neter, & Li, 2005)
Independence: Durbin-Watson Test	(Kutner, Nachtsheim, Neter, & Li, 2005)
Homoscedasticity: Residuals Plot; the White test; Breusch-Pagan Test	(Kutner, Nachtsheim, Neter, & Li, 2005)
Normality of Errors: Q-Q Plot; skewness & kurtosis	(Kutner, Nachtsheim, Neter, & Li, 2005)
No Multicollinearity: Variance Inflation Factor (VIF)	(Kutner, Nachtsheim, Neter, & Li, 2005)

1

Non-linearity

When the linearity assumption is violated, the points in the residual plot will not be randomly scattered. Instead, the points will often show some “curvature”.

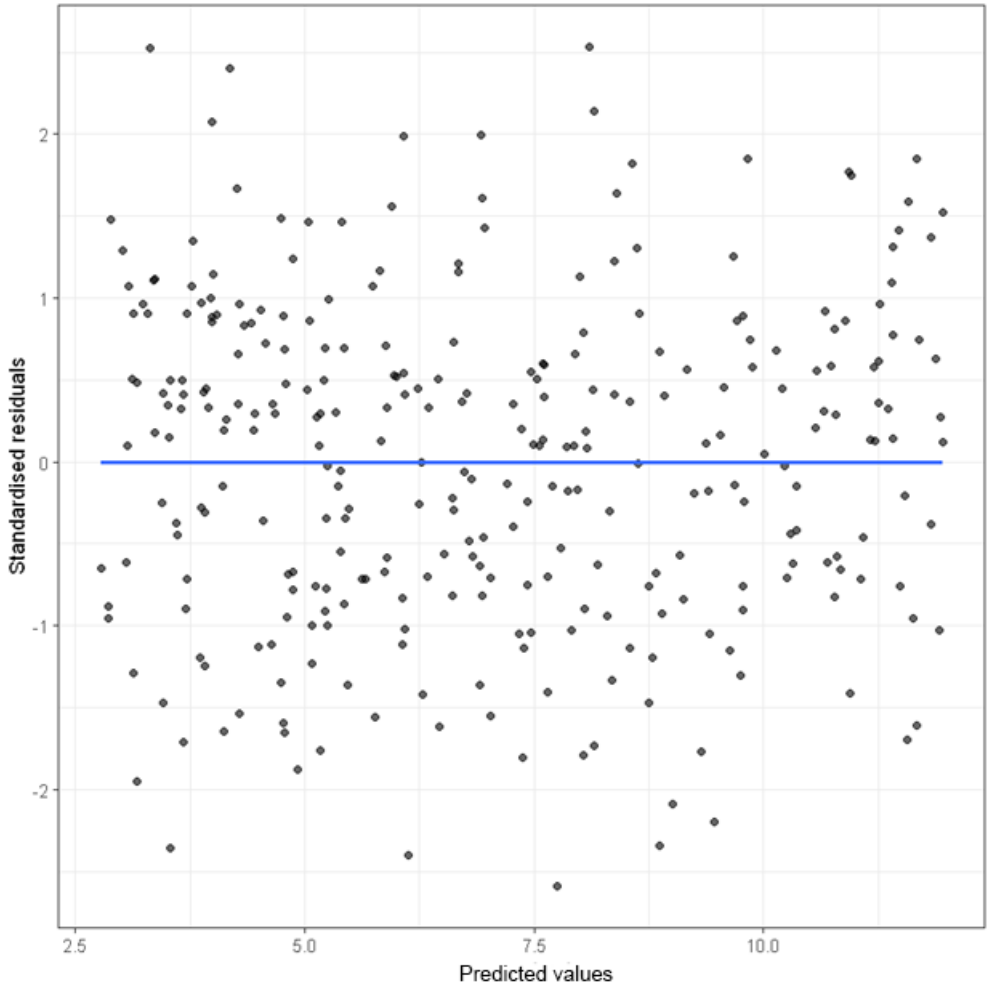


Source: Residual Plots and Assumption Checking - StatsNotebook - Simple. Powerful. Reproducible.

Homoscedasticity

Assumption met

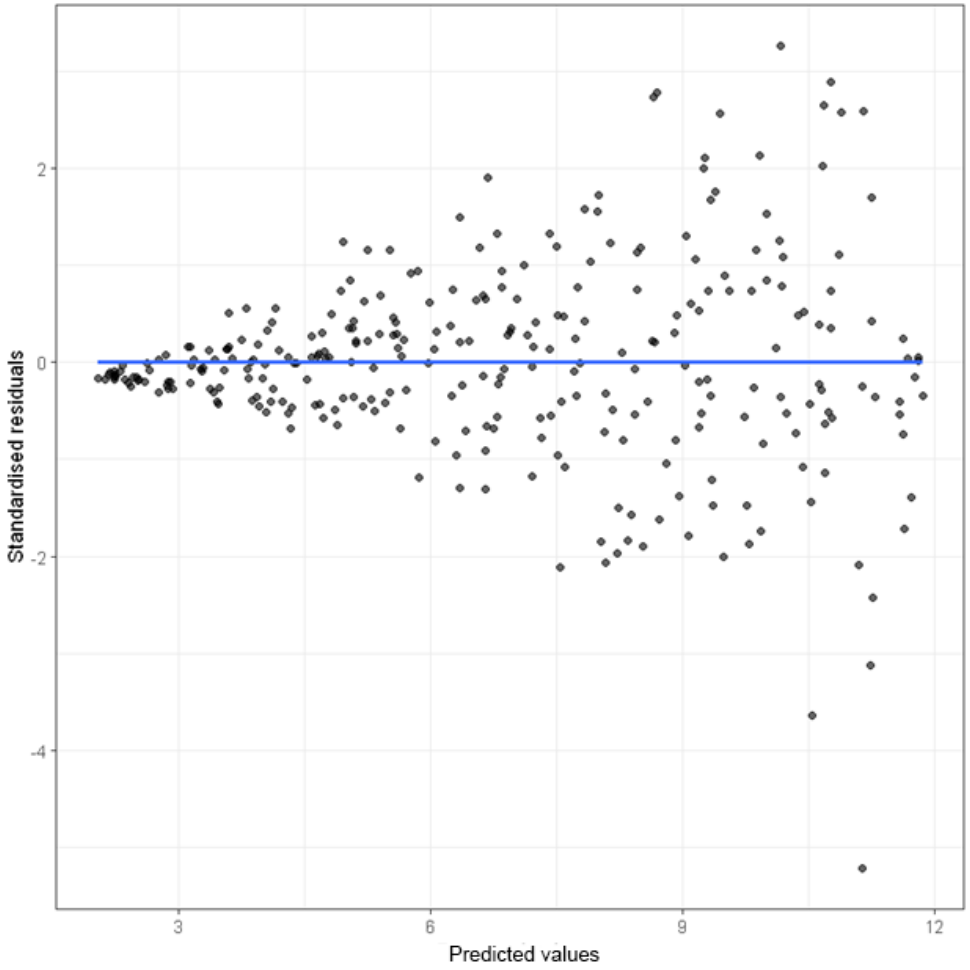
When both the assumption of linearity and **homoscedasticity** are met, the points in the residual plot (plotting standardised residuals against predicted values) will be **randomly** scattered.



2

Heteroscedasticity

When the homoscedasticity assumption is violated, the "spread" of the points across predicted values are not the same. The following are two plots that indicate a violation of this assumption.



Source: Residual Plots and Assumption Checking - StatsNotebook - Simple. Powerful. Reproducible.

Sphericity of variances
(repeated measures ANOVA)

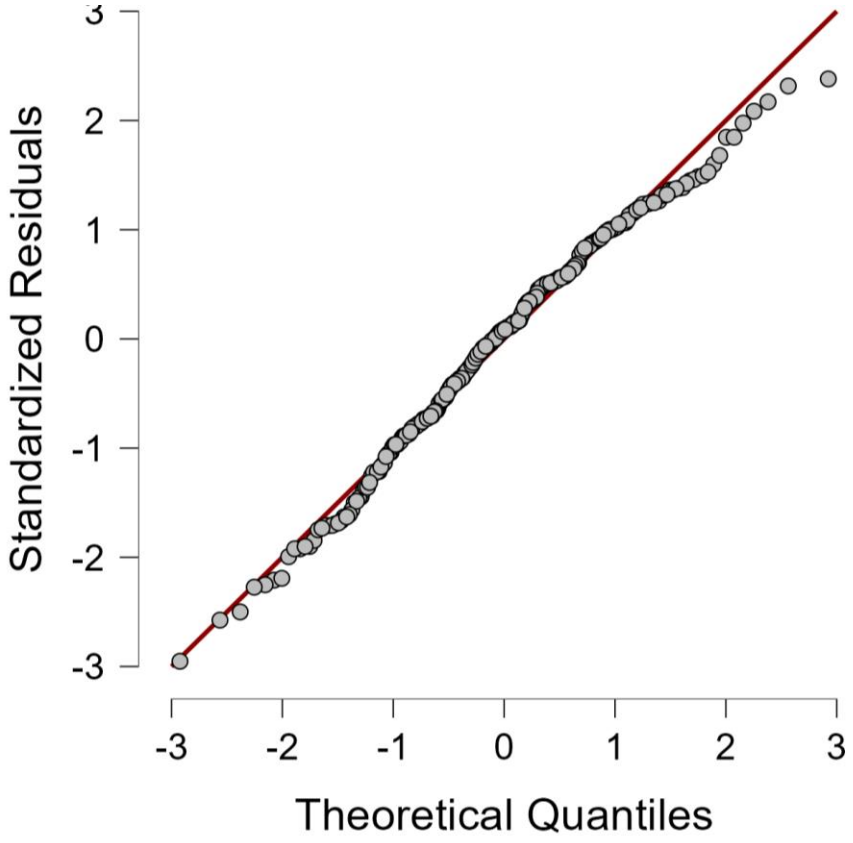
2

Patient	Tx A	Tx B	Tx C	Tx A − Tx B	Tx A − Tx C	Tx B − Tx C
1	30	27	20	3	10	7
2	35	30	28	5	7	2
3	25	30	20	−5	5	10
4	15	15	12	0	3	3
5	9	12	7	−3	2	5
			Variance:	17	10.3	10.3

Mauchly's test of sphericity

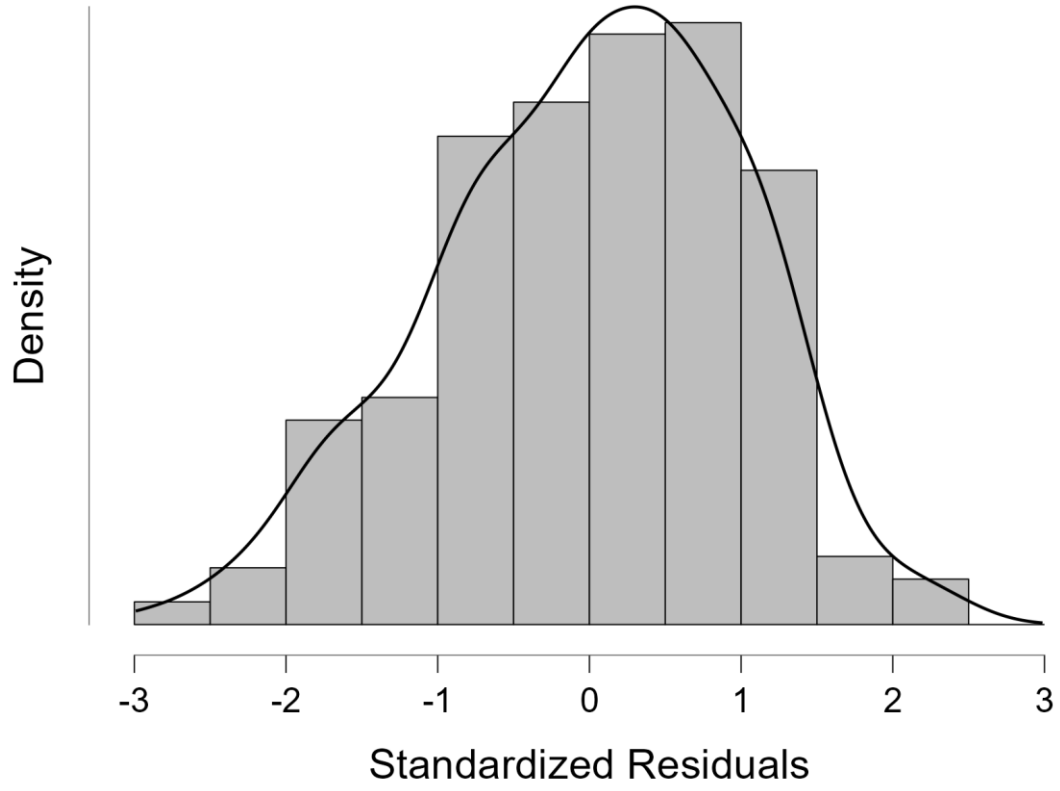
Normality

Q-Q Plot Standardized Residuals



The points should lie approximately along the reference line (red line).

Standardized Residuals Histogram



It should resemble a bell-shaped curve (normal distribution).

Multicollinearity test in JASP

4

Collinearity Diagnostics ▼

Model	Dimension	Eigenvalue	Condition Index	Variance Proportions			
				(Intercept)	Grammar	Vocab	Reading
H_1	1	3.806	1.000	0.005	0.004	0.007	0.006
	2	0.083	6.752	0.321	0.034	0.729	0.006
	3	0.073	7.212	0.127	0.009	0.210	0.914
	4	0.037	10.079	0.547	0.954	0.054	0.074

Note. The intercept model is omitted, as no meaningful information can be shown.

- **Dimension**: This refers to the principal components of the predictors' variance-covariance matrix. Each dimension explains a certain amount of variance in the predictors.
- **Eigenvalue**: Eigenvalues correspond to the variance explained by each dimension. A small eigenvalue indicates that a dimension explains very little variance, which could be a sign of multicollinearity.
- **Condition Index**: The condition index is the square root of the ratio of the largest eigenvalue to each individual eigenvalue. Higher condition index values (**generally above 30**) indicate potential multicollinearity problems.
- **Variance Proportions**:
  - how much of the variance of each predictor's regression coefficient is associated with each eigenvalue.
  - Look for rows where two or more variables have high proportions (**usually >0.5**)

Multicollinearity: VIF & Tolerance

4

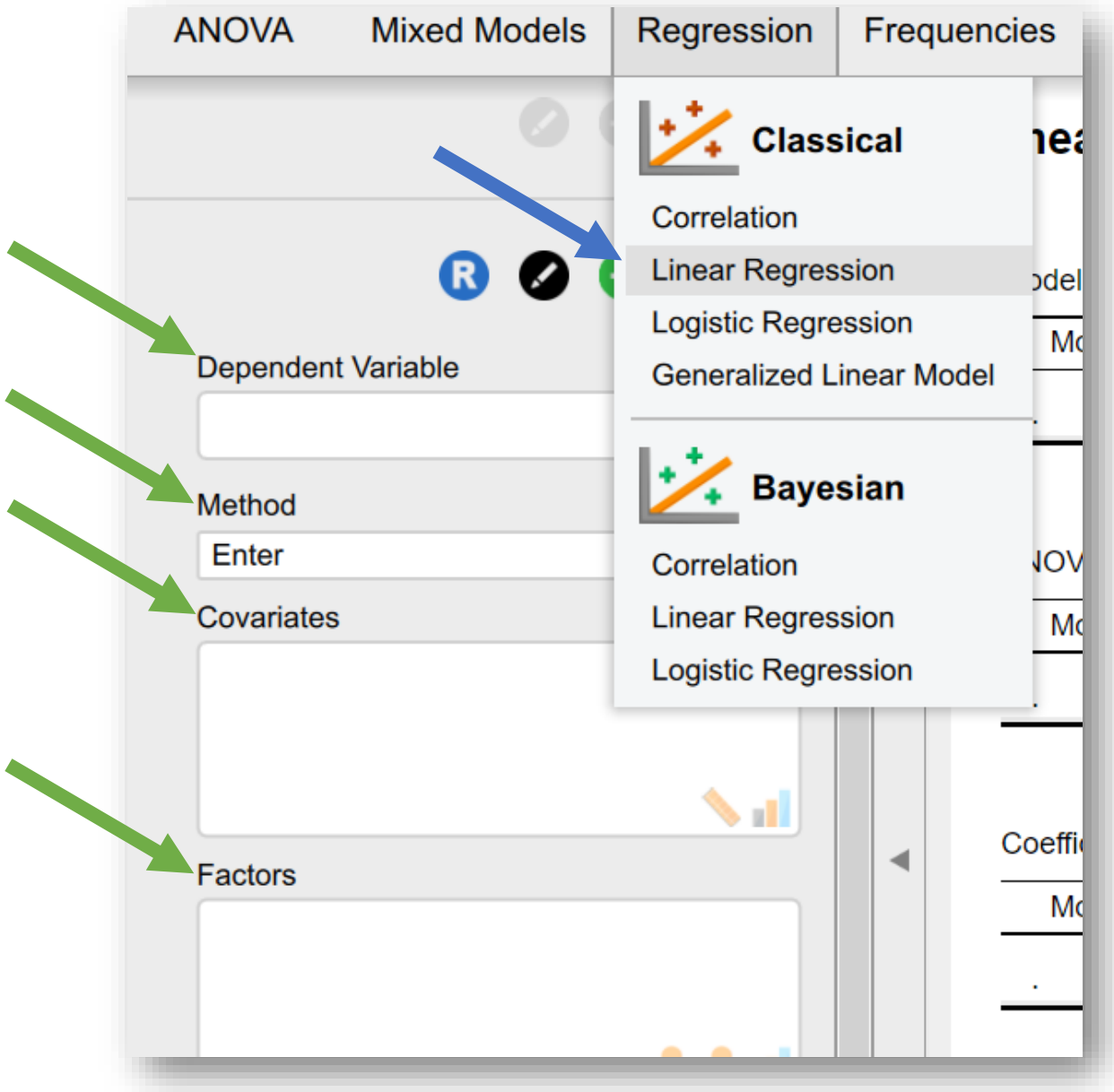
- **Variance Inflation Factor (VIF)** measures how much the variance of a regression coefficient is inflated due to **multicollinearity** with other predictors.
- **VIF = 1**: No correlation between the predictor and other predictors.
- **1 < VIF < 5**: Moderate correlation that is typically not problematic.
- **VIF > 5**: Indicates a potentially problematic level of **multicollinearity**.
- **VIF > 10**: Suggests significant multicollinearity, warranting corrective measures.

- **Tolerance** is the inverse of VIF and represents the proportion of variance in a predictor that is not explained by the other predictors.
- **Tolerance = 1**: No multicollinearity.
- **Tolerance < 0.2**: Indicates potential multicollinearity issues.
- **Tolerance < 0.1**: Suggests severe multicollinearity problems.

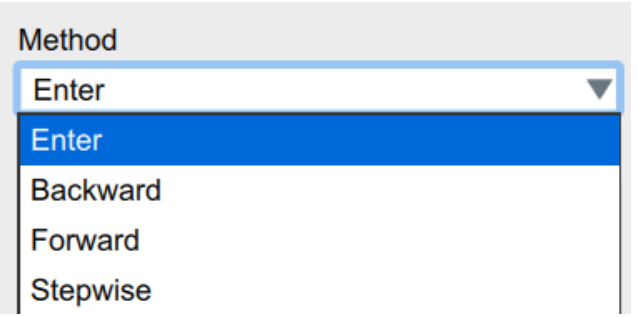
Sample Interpretation

4

- If a predictor variable has a **VIF of 6** and a **Tolerance of 0.167**:
- **VIF Interpretation**: A VIF of 6 indicates that the variance of the regression coefficient for this predictor is 6 times higher than it would be if there were no correlation with other predictors. This suggests a moderate to high level of multicollinearity that may need to be addressed.
- **Tolerance Interpretation**: A Tolerance of 0.167 (which is 1/6) indicates that approximately 16.7% of the variance in the predictor is not explained by the other predictors. This low value also signals a concern with multicollinearity.

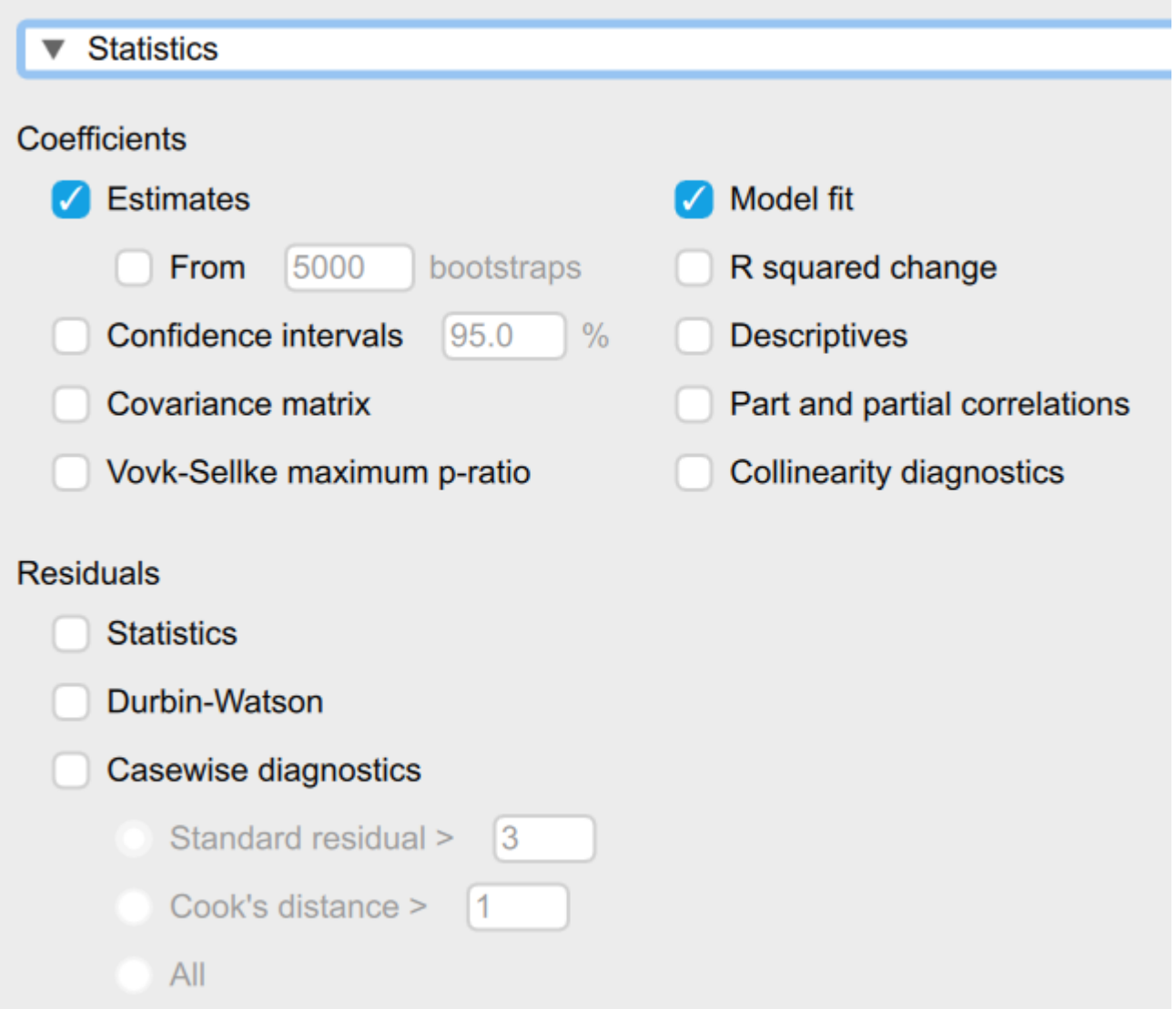

ANOVA
Mixed Models
Regression
Frequencies
Classical
Correlation
Linear Regression
Logistic Regression
Generalized Linear Model
Bayesian
Correlation
Linear Regression
Logistic Regression
Dependent Variable
Method
Enter
Covariates
Factors

Methods



- **Enter**:
  - includes **all** the specified independent variables in the regression model simultaneously, regardless of their statistical significance.
  - **useful** when you have **strong theoretical reasons** to include certain variables in the model.
- **Backward**: (useful for model comparison [in some software])
  - starts with **all** the independent variables included in the model. It then **iteratively** removes the least significant variable (based on a chosen p-value threshold) until all remaining variables are statistically significant.
  - Use it when you have many potential predictors and want to identify the most important ones.
- **Forward**: (useful for model comparison)
  - starts with **no variables** in the model. It then adds the **most significant** variable at each step. This continues until no additional variables meet the inclusion criteria.
  - When you want to build a model incrementally, starting with the most important predictors.
- **Stepwise**: (useful for model comparison)
  - a combination of **Forward** Selection and **Backward** Elimination.
  - adds variables to the model **one by one** (like Forward Selection) but also **tests** at each step whether any variables should be removed (like Backward Elimination).
  - ensures that only the most significant variables remain in the model.

- The Durbin-Watson test
  - a statistical test used to detect the presence of **autocorrelation** (also called serial correlation) in the residuals of a regression analysis.
  - particularly useful in **time series data**
  - one of the key assumptions of **ordinary least squares** (OLS) regression



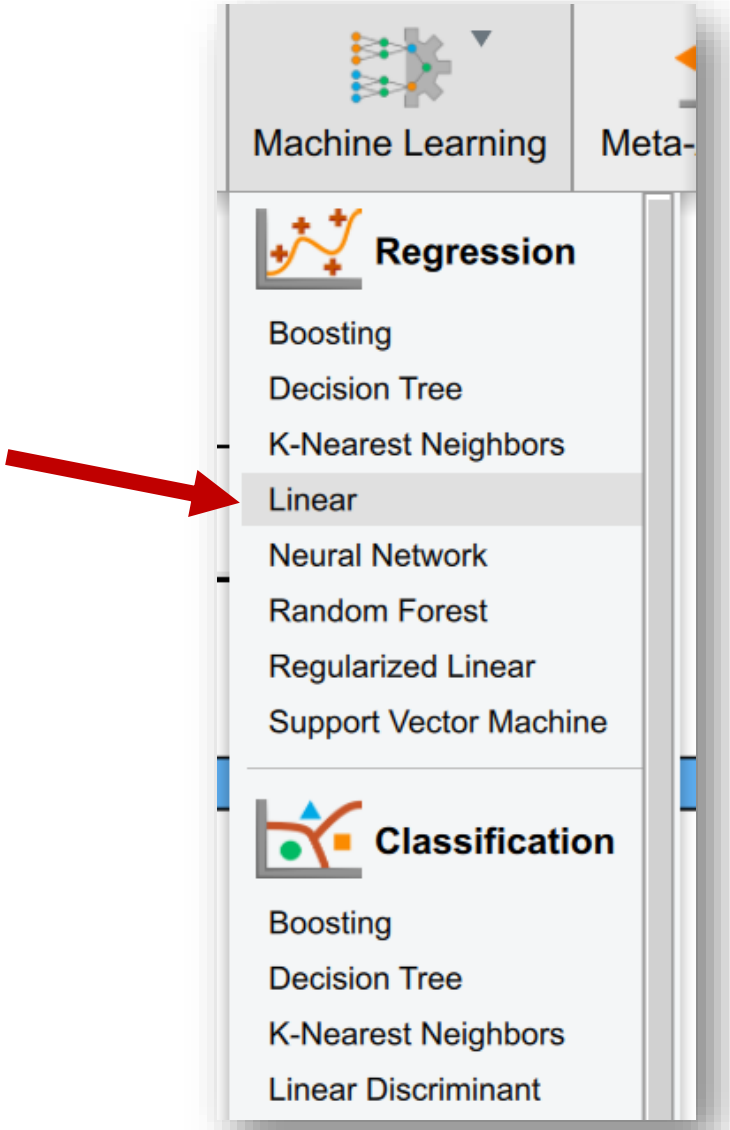
Output

- **R^2** and **adjusted R^2 (**adjusted for the number of predictors**)**
- **RMSE** (Root Mean Square Error):
  - A general measure of the **differences** between values **predicted** by a model or an estimator and the values **observed**.
  - To measure the **average** magnitude of the **prediction error**.
  - The **square root** of the **average** of the **squared differences** between **predicted** and **observed** values.
  - **Lower values** of RMSE indicate better predictive accuracy of the model.
- **Unstandardized** and **standardized** coefficients & *p* values

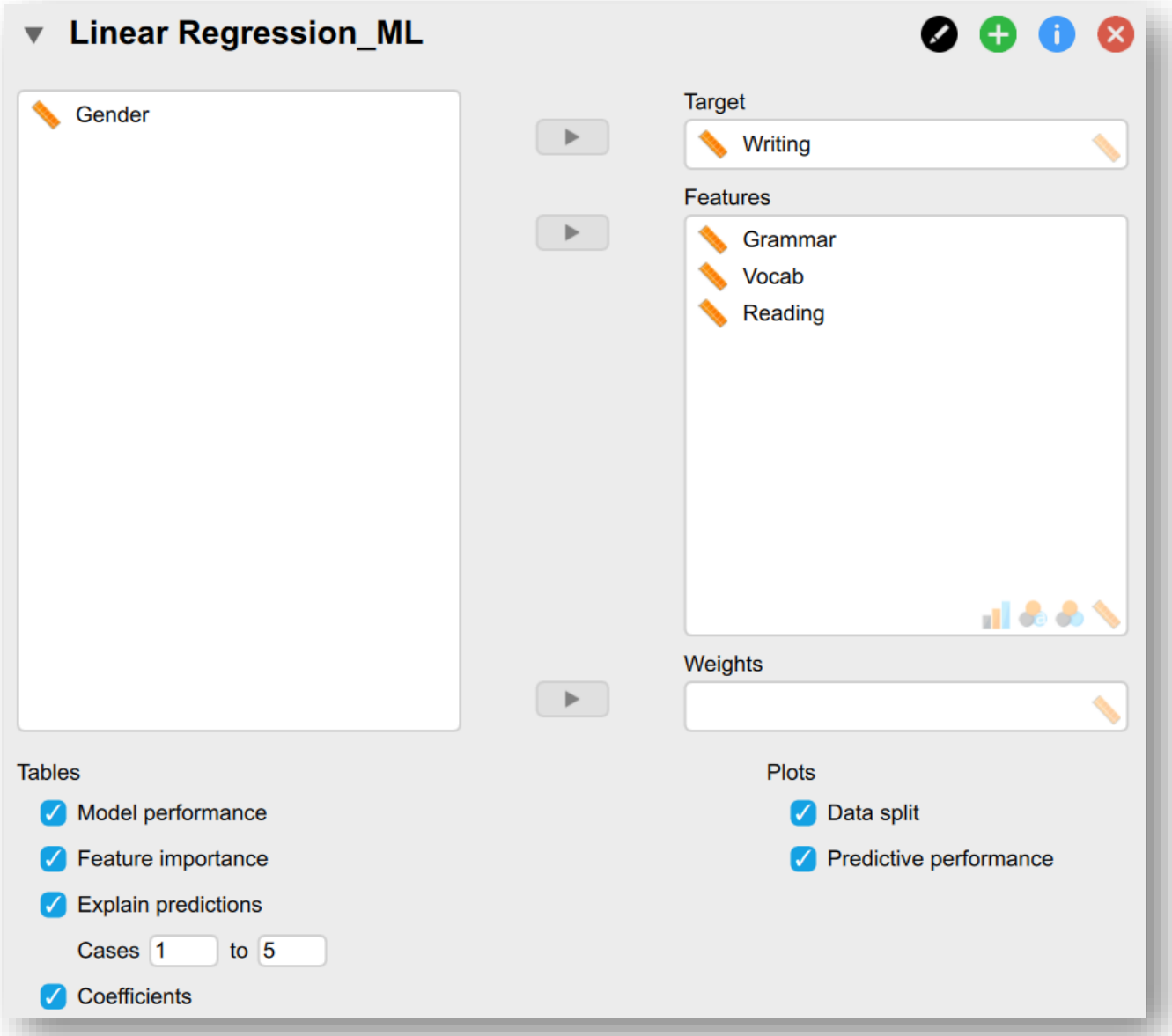
$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

Multiple linear regression: The algorithm-based method

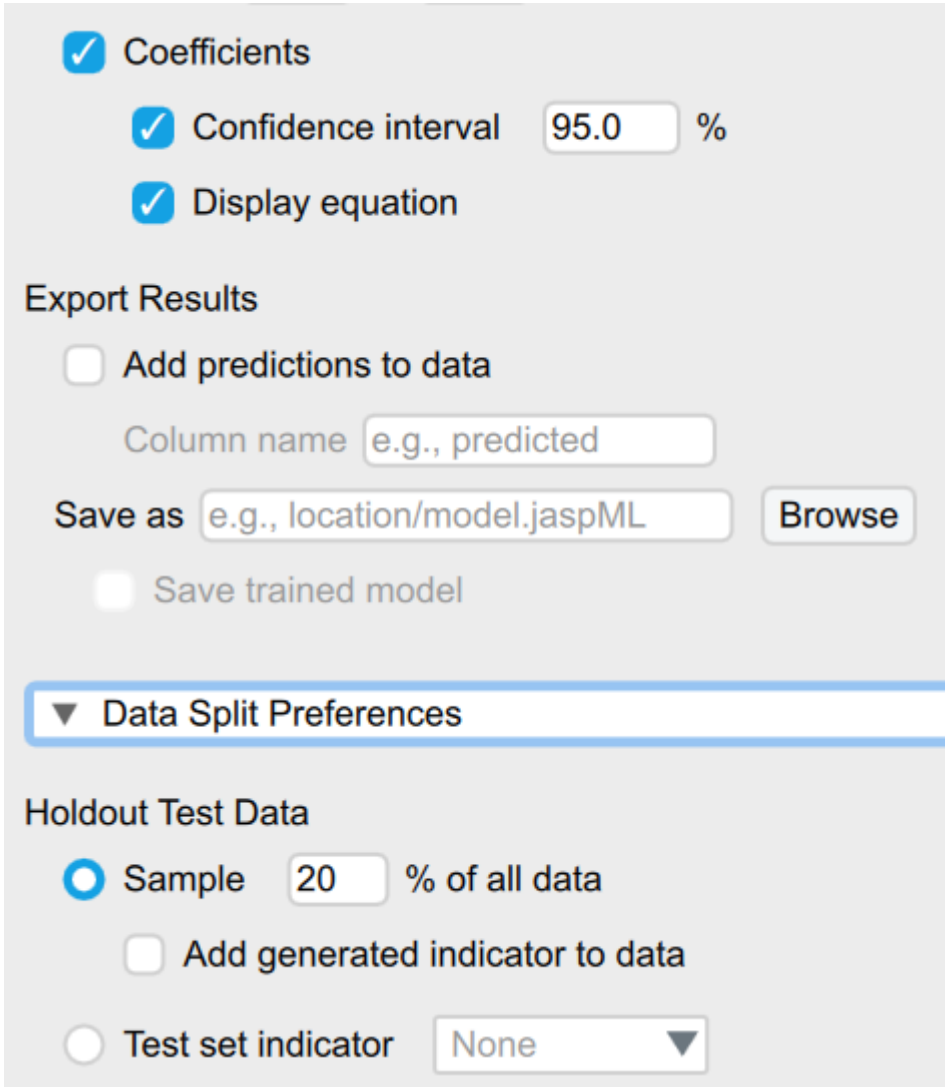
1


Machine Learning
Meta-
Regression
Boosting
Decision Tree
K-Nearest Neighbors
Linear
Neural Network
Random Forest
Regularized Linear
Support Vector Machine
Classification
Boosting
Decision Tree
K-Nearest Neighbors
Linear Discriminant

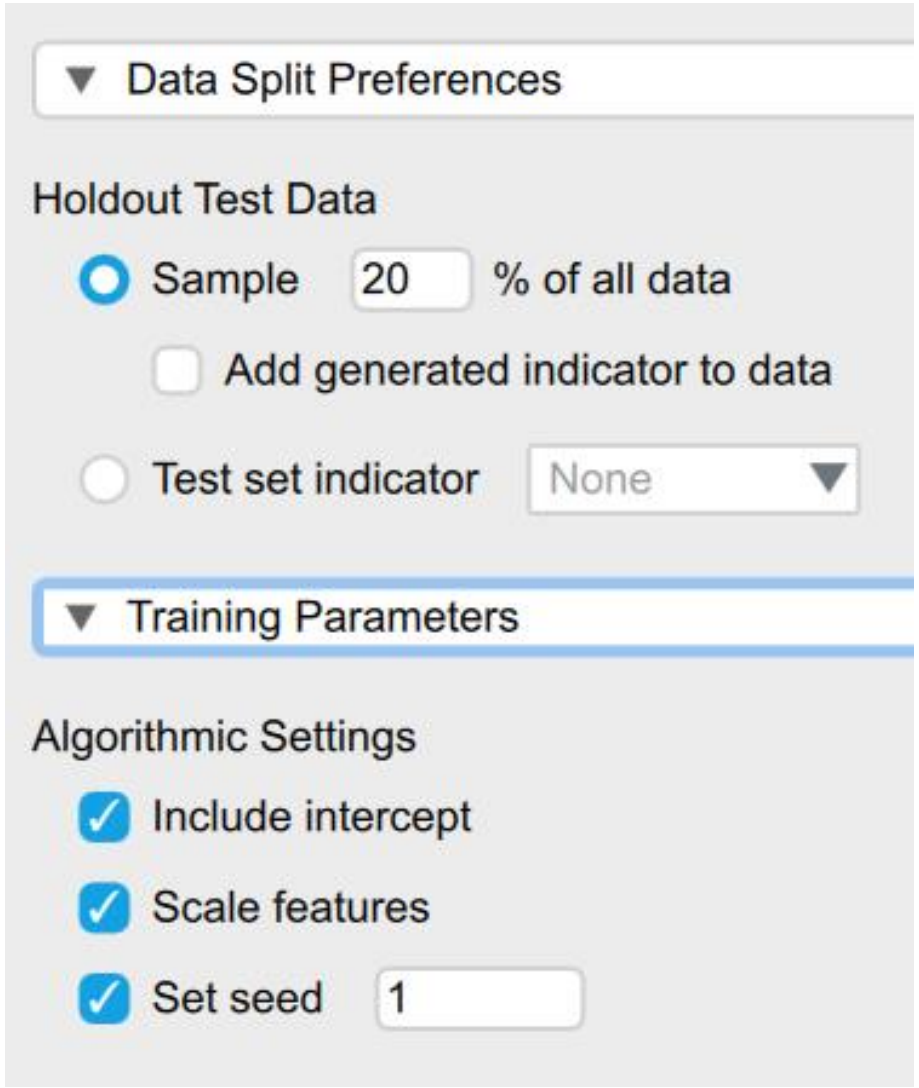
2


Linear Regression_ML
Gender
Target
Writing
Features
Grammar
Vocab
Reading
Weights
Tables
Model performance
Feature importance
Explain predictions
Cases 1 to 5
Coefficients
Plots
Data split
Predictive performance

3

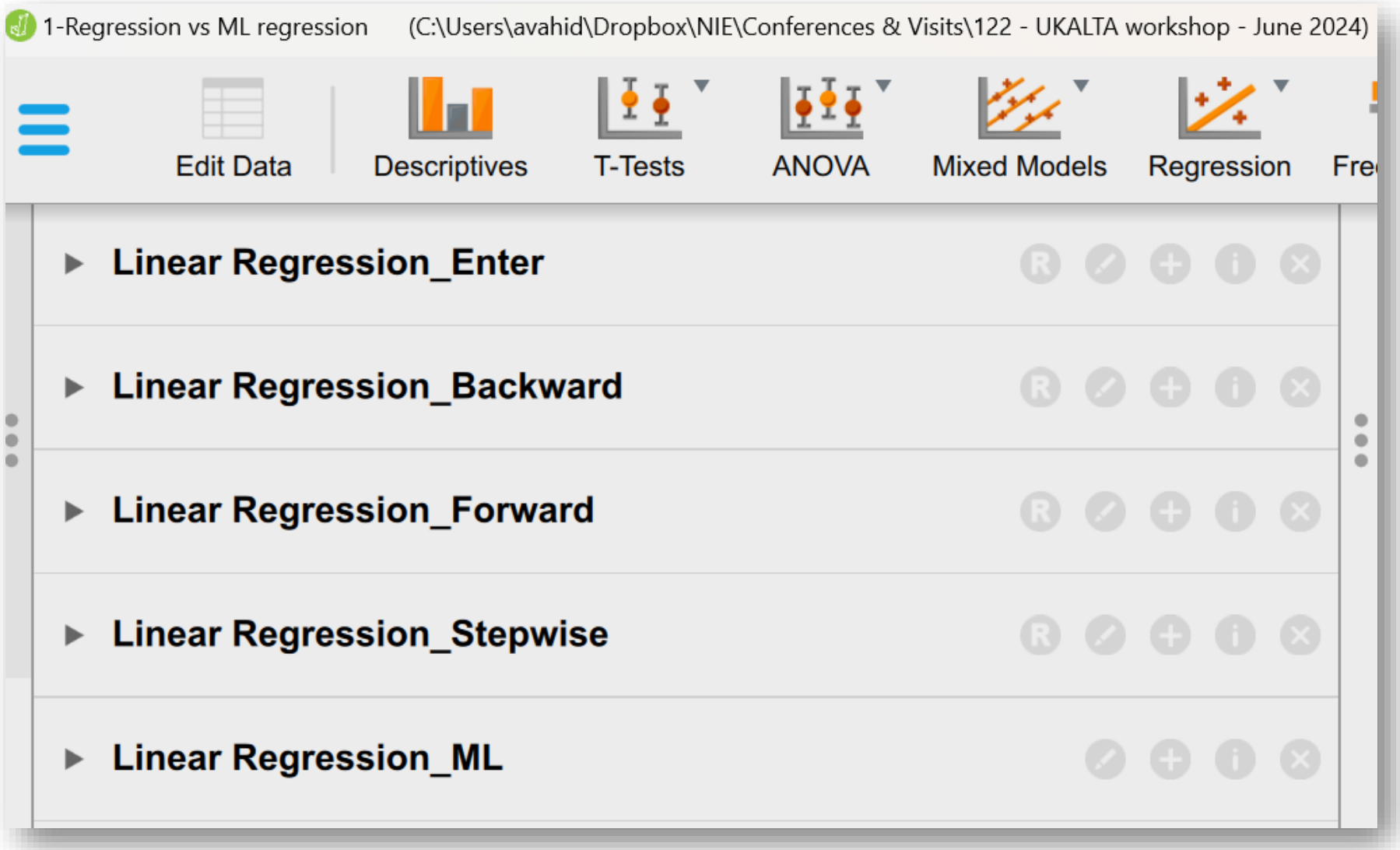

Coefficients
Confidence interval 95.0 %
Display equation
Export Results
Add predictions to data
Column name e.g., predicted
Save as e.g., location/model.jaspML Browse
Save trained model
Data Split Preferences
Holdout Test Data
Sample 20 % of all data
Add generated indicator to data
Test set indicator None

4


Data Split Preferences
Holdout Test Data
Sample 20 % of all data
Add generated indicator to data
Test set indicator None
Training Parameters
Algorithmic Settings
Include intercept
Scale features
Set seed 1

Hands-on Activity:

1-Regression vs ML regression.jasp



calculated using the entire dataset

Linear Regression

n(Train)	n(Test)	Test MSE	R^2	Adjusted R^2
232	57	22.846	0.205	0.194

Data Split

Model Performance Metrics ▼

	Value
MSE	22.846
RMSE	4.78
MAE / MAD	3.93
MAPE	Inf%
R^2	0.289

calculated using the testing dataset (?)

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Model Performance Metrics ▼

	Value
MSE	22.846
RMSE	4.78
MAE / MAD	3.93
MAPE	Inf%
R^2	0.289

Mean Squared Error (MSE): the average of the squared differences between the actual and predicted values.

Root Mean Squared Error (RMSE): the square root of the MSE and provides a measure of the standard deviation of the residuals (prediction errors). The lower, the better.

Mean Absolute Error (MAE) / Mean Absolute Deviation (MAD): the average of the absolute differences between actual and predicted values. The lower, the better.

Mean Absolute Percentage Error (MAPE) measures the average absolute percentage error between the predicted and actual values. It is expressed as a percentage, making it scale-independent and useful for comparing models on different datasets.

0% = perfect fit.
below 10% = excellent,
10-20% = good
20-50% = acceptable,
above 50% = poor.

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100$$

R2: Values closer to 1 are better. >0.7 is generally great; 0.4 or smaller might be acceptable in some fields.

Feature importance metrics

- Feature importance metrics helps in understanding which **features contribute most** to the predictive power of the model.
- The values are calculated based **on 50 permutations**, which means the model's performance was assessed **50 times** for each feature.
- The importance of features is estimated by systematically 'dropping out' or **removing** each feature from the model and observing the increase in the **model's loss (error)**. More important features, when removed, will cause a larger increase in the model's loss.
- A **higher** mean dropout loss indicates that the feature is **more important** because its absence leads to a **significant degradation** in model performance.

Grammar has the highest mean dropout loss (4.924), suggesting it is **the most important feature** among the three because its removal has the greatest negative impact on model performance.

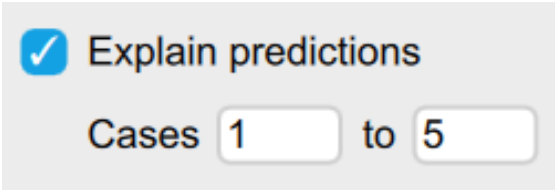
Feature Importance Metrics ▼

	Mean dropout loss
Grammar	4.924
Vocab	4.520
Reading	4.421

Note. Mean dropout loss is based on 50 permutations.

- Case: This is simply the identifier for each test case. i.e., an individual datum or observation.
- Predicted: This column shows the final predicted value for each case, combining all features.
- Base: This represents the baseline prediction without considering any specific features. It's constant across all cases (12.634), suggesting this is likely **the mean of the target variable** in the training set.
- These values represent how each feature contributes to the predicted value for each case in the test set.
- Example: **Case 1:** The predicted value of 16.107 is obtained by adding the base value of 12.634 and the contributions from Grammar (2.051), Vocab (1.018), and Reading (0.405).

The table provided shows a sample of cases for illustrative purposes.

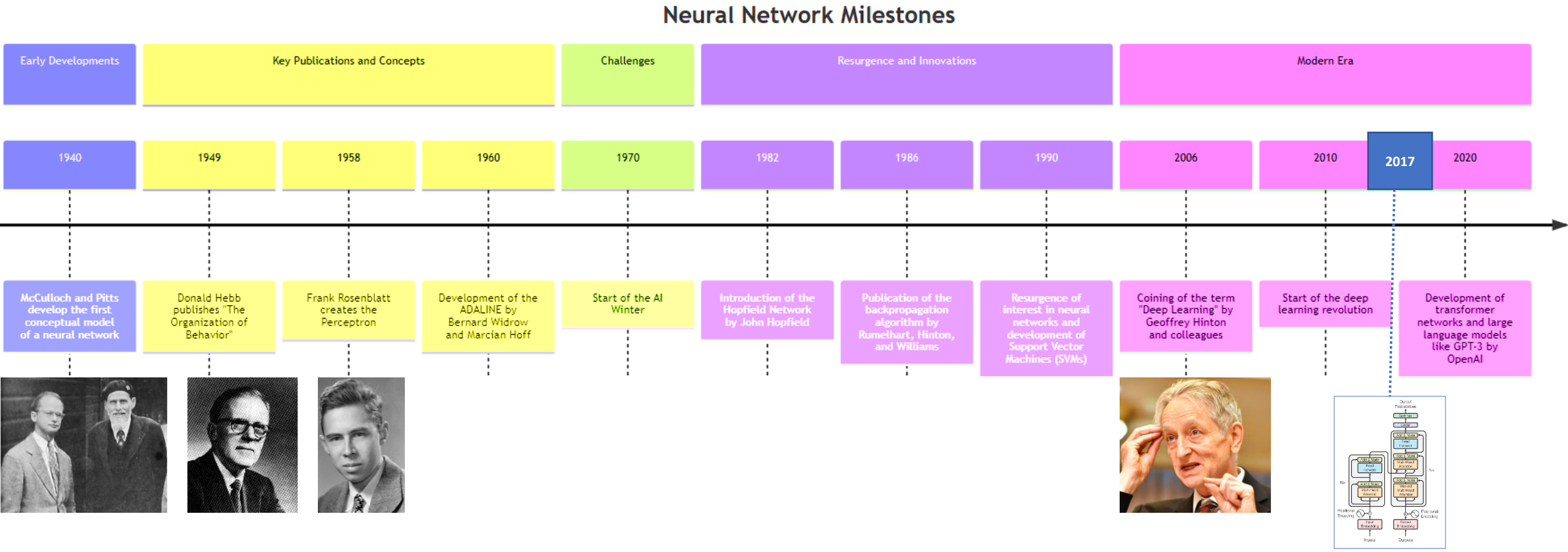


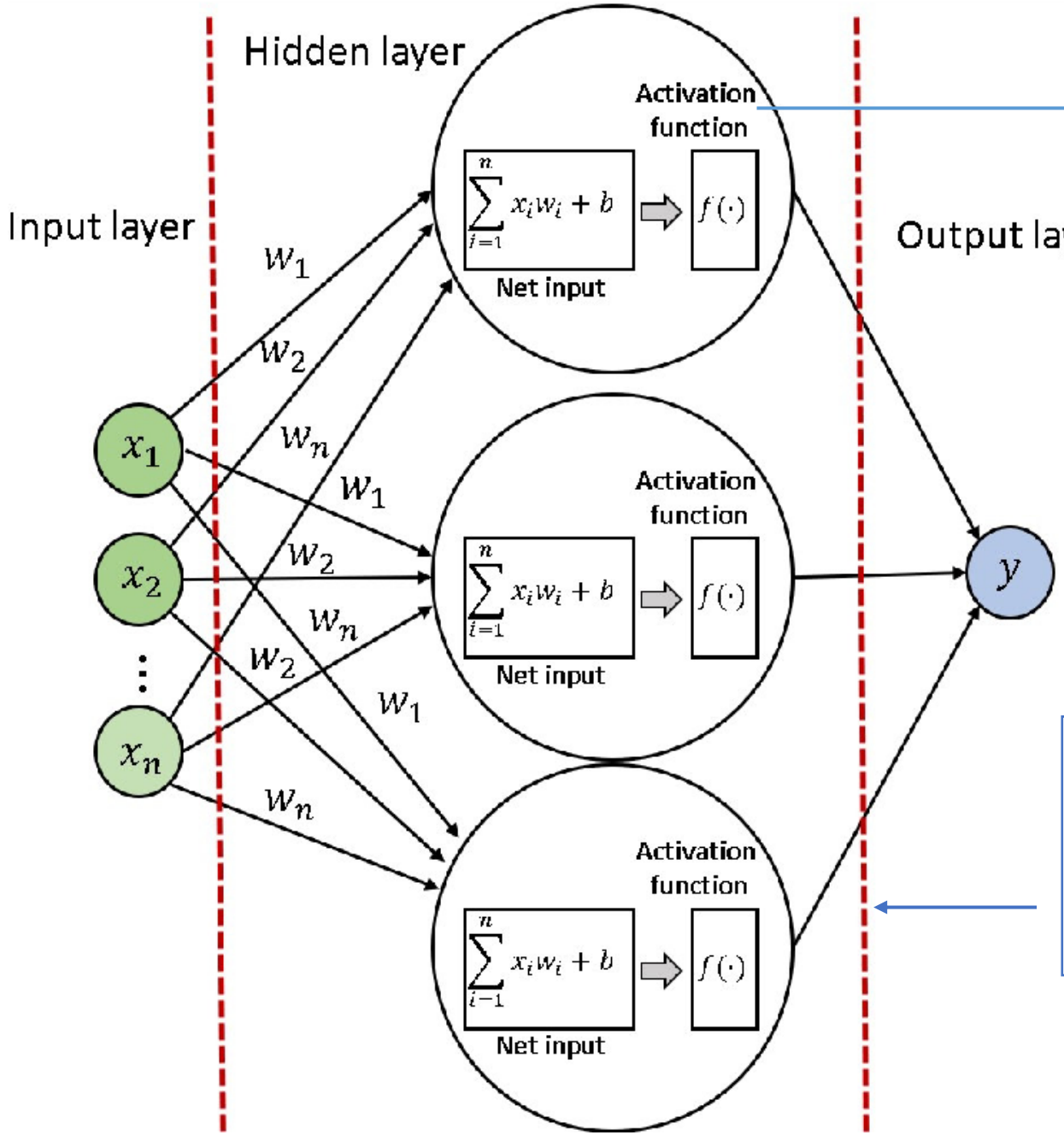
Additive Explanations for Predictions of Test Set Cases

Case	Predicted	Base	Grammar	Vocab	Reading
1	16.107	12.634	2.051	1.018	0.405
2	15.636	12.634	2.625	0.330	0.047
3	13.622	12.634	1.477	−0.357	−0.132
4	15.355	12.634	1.477	1.018	0.226
5	14.732	12.634	2.051	−0.357	0.405

Note. Displayed values represent feature contributions to the predicted value without features (column 'Base') for the test set.

Can machines think? Artificial neural networks





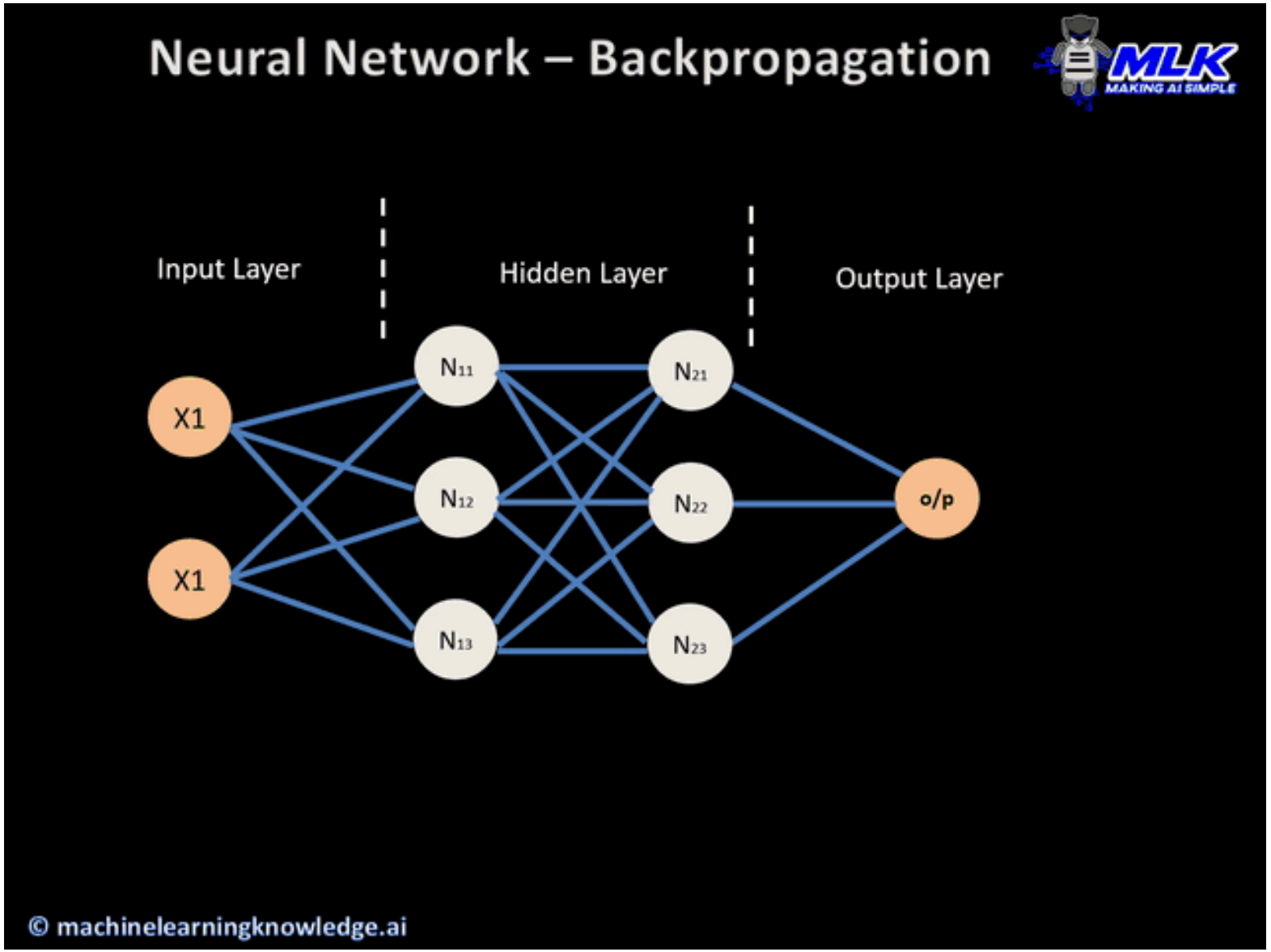
1. Rectified Linear Unit (ReLU)
2. Leaky ReLU
3. Parametric ReLU (PReLU)
4. Exponential Linear Unit (ELU)
5. Scaled Exponential Linear Unit (SELU)
6. Hyperbolic Tangent (tanh)
7. Softmax
8. Softplus
9. Sigmoid or Logistic
10. Swish

Loss function

calculates "loss" or an aggregate measure derived from the "errors"

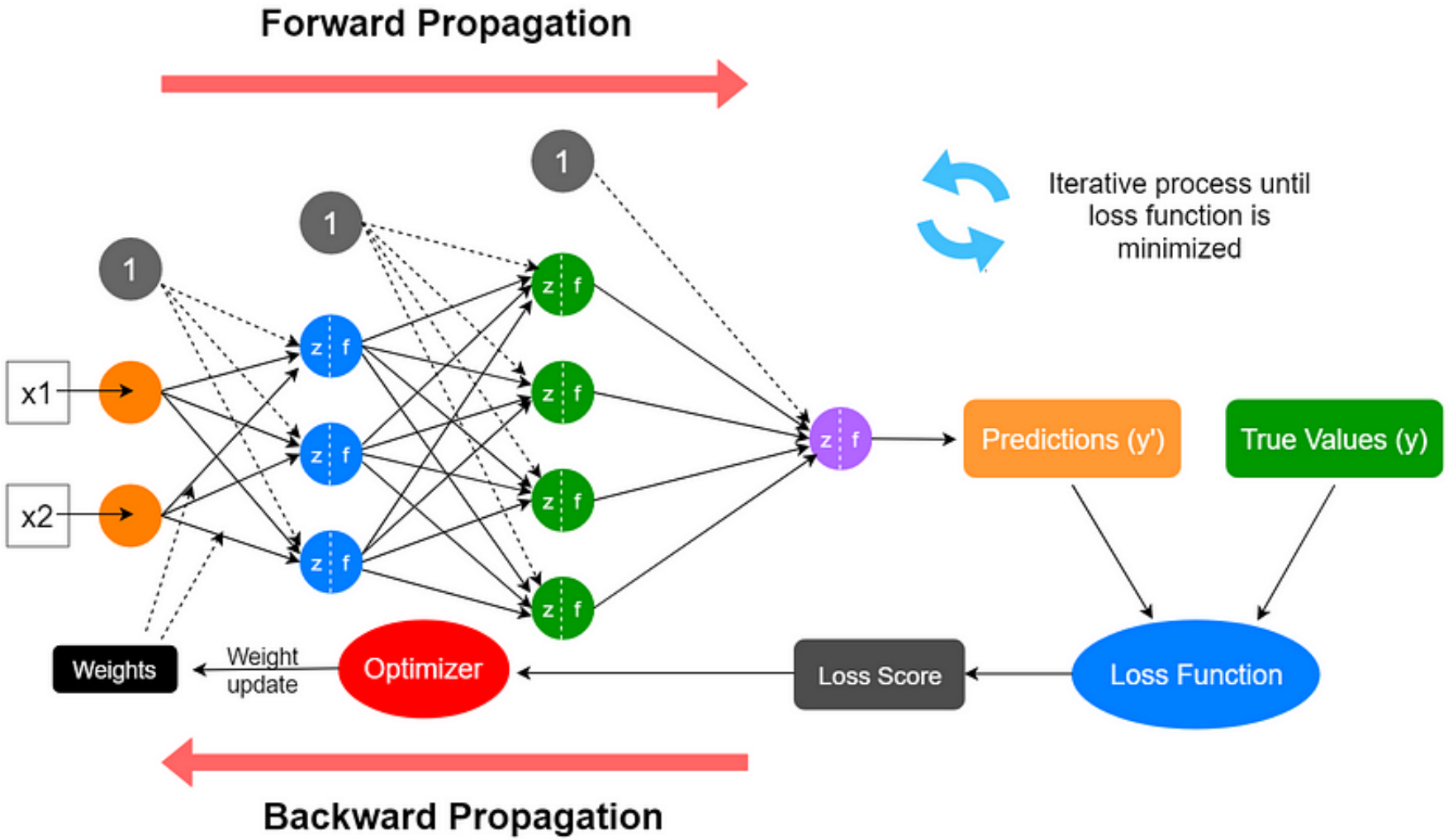
Backpropagation: using the error to adjust & optimize the weights of the network

Salim, S. (2023). Neural Networks in 10mins. Simply Explained! | by Sadaf Saleem | Medium

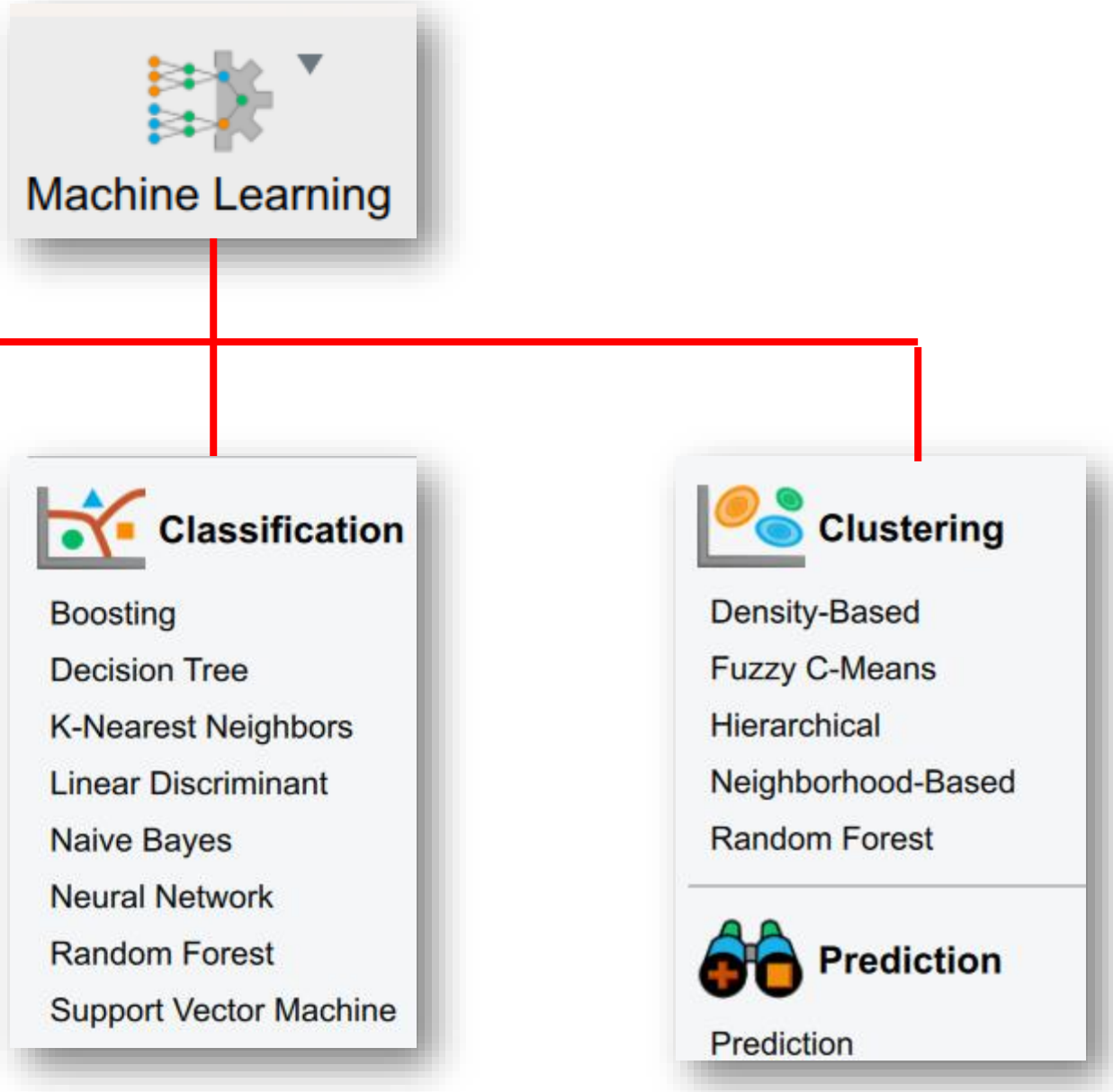


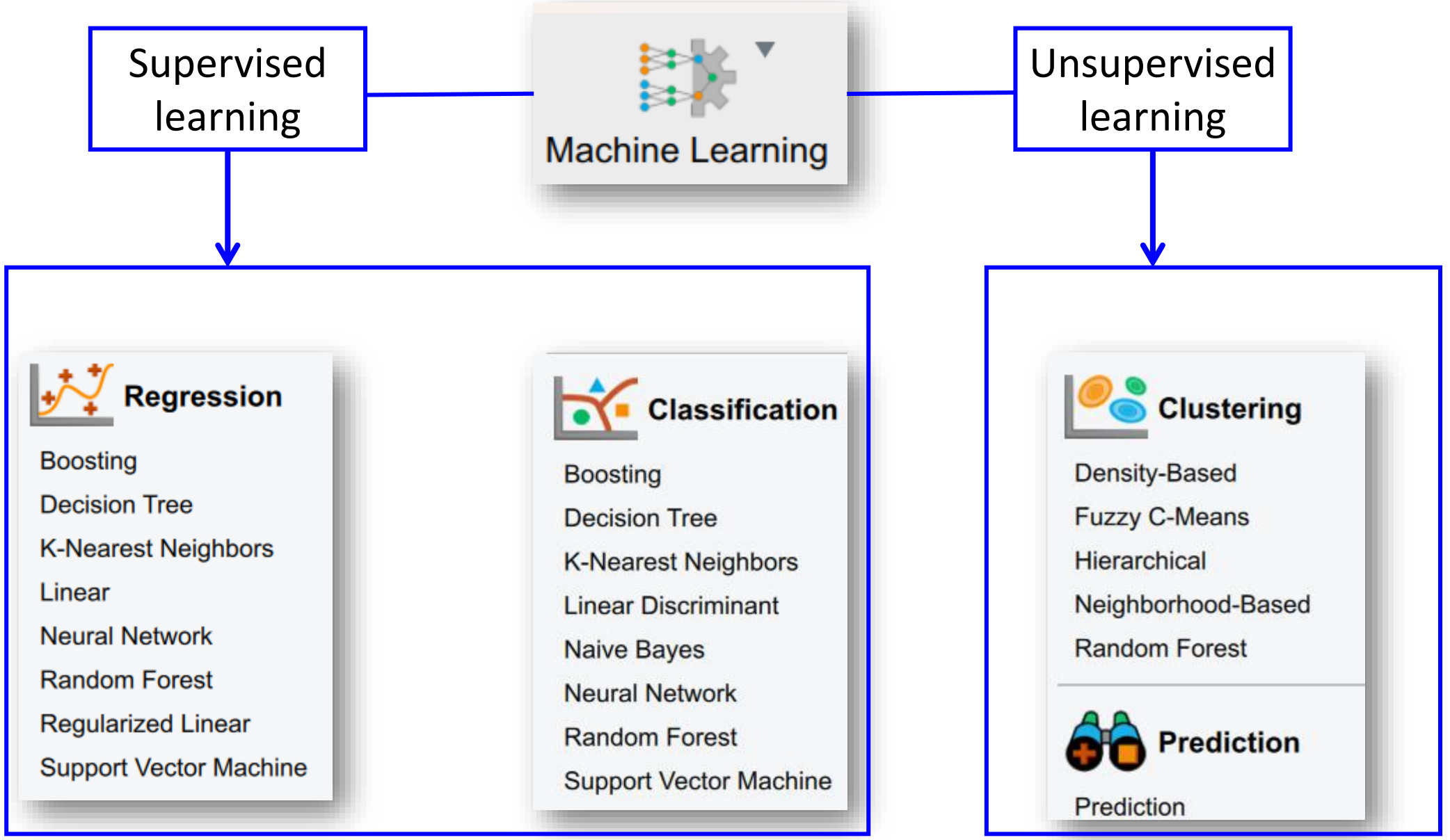
(machinelearningknowledge.ai)

A two-layered neural network



Source: Overview of a Neural Network's Learning Process | by Rukshan Pramoditha | Data Science 365 | Medium


Machine Learning
Regression
Boosting
Decision Tree
K-Nearest Neighbors
Linear
Neural Network
Random Forest
Regularized Linear
Support Vector Machine
Classification
Boosting
Decision Tree
K-Nearest Neighbors
Linear Discriminant
Naive Bayes
Neural Network
Random Forest
Support Vector Machine
Clustering
Density-Based
Fuzzy C-Means
Hierarchical
Neighborhood-Based
Random Forest
Prediction
Prediction


Machine Learning
Supervised learning
Unsupervised learning
Regression
Boosting
Decision Tree
K-Nearest Neighbors
Linear
Neural Network
Random Forest
Regularized Linear
Support Vector Machine
Classification
Boosting
Decision Tree
K-Nearest Neighbors
Linear Discriminant
Naive Bayes
Neural Network
Random Forest
Support Vector Machine
Clustering
Density-Based
Fuzzy C-Means
Hierarchical
Neighborhood-Based
Random Forest
Prediction
Prediction

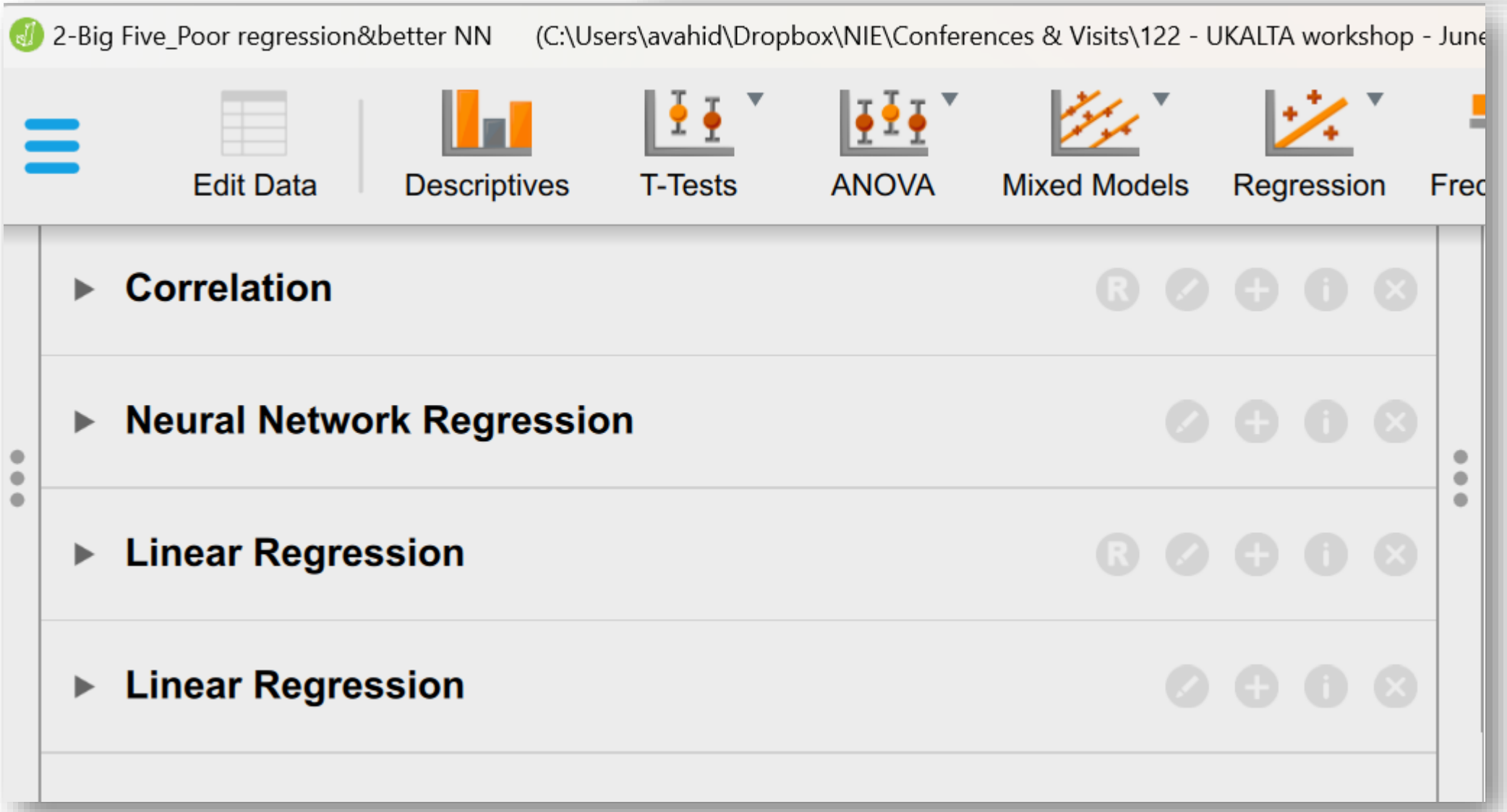
Machine learning

- **Supervised learning** is a type of machine learning where the model is trained on **labeled** data. This means that each training example is paired with an output label.
  - **Classification**: Identifying the category of an input (e.g., spam detection in emails).
  - **Regression**: Predicting a continuous value (e.g., predicting house prices).
- **Unsupervised learning** is a type of machine learning where the model is trained on **unlabeled** data. The goal is to identify patterns or structures within the data.

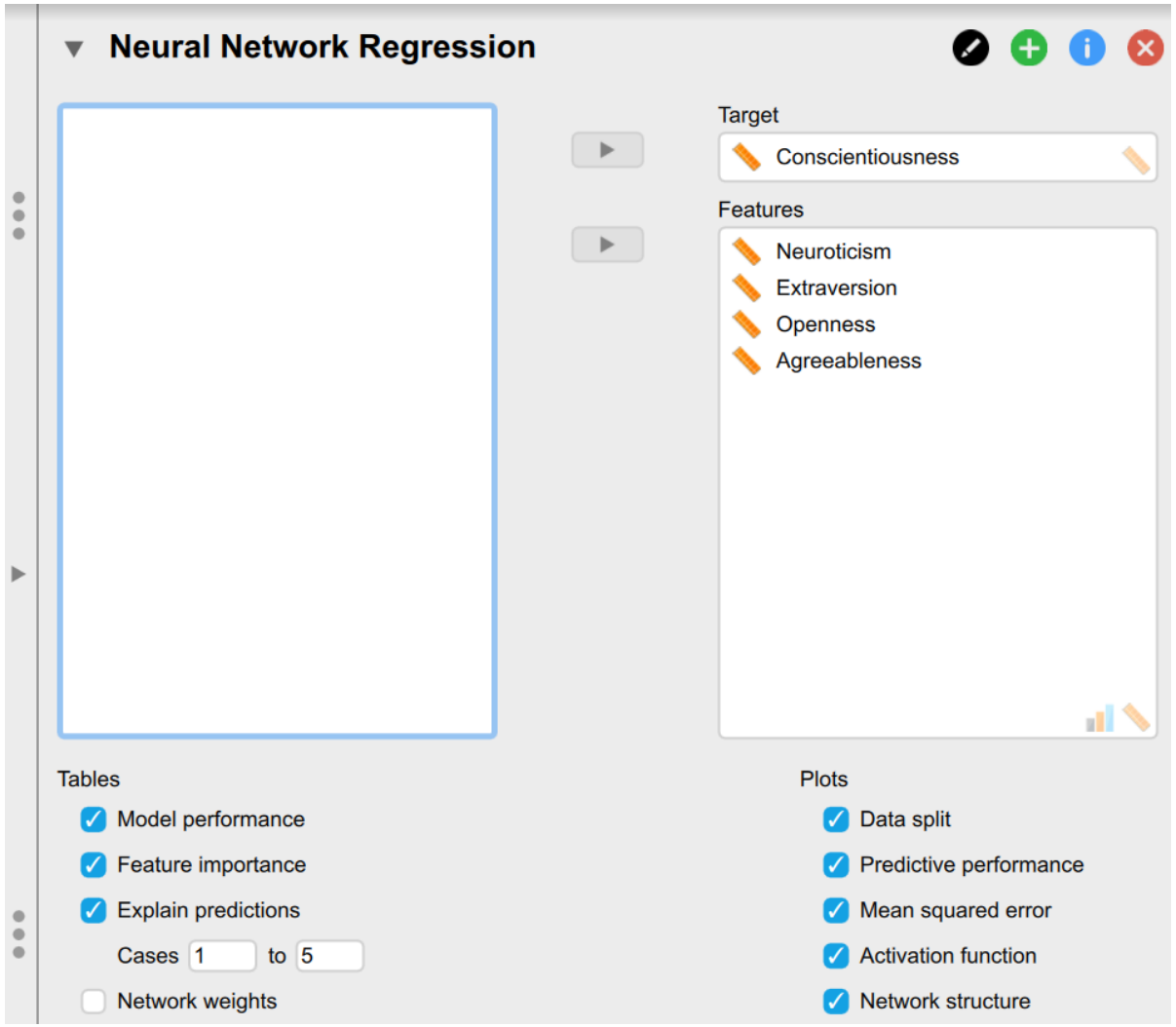
Other types of ML

- **Semi-supervised learning** is a type of machine learning that involves a small amount of labeled data and a large amount of unlabeled data; Combines both supervised and unsupervised learning techniques.
- **Reinforcement learning** is a type of machine learning where an agent learns to make decisions by performing actions and receiving feedback from the environment. E.g., The agent receives rewards or penalties based on its actions
- **Self-supervised learning** is a type of machine learning where the data provides the supervision. It involves generating labels from the input data itself, allowing the model to learn without explicit external labels. E.g., Predicting the next word in a sentence.

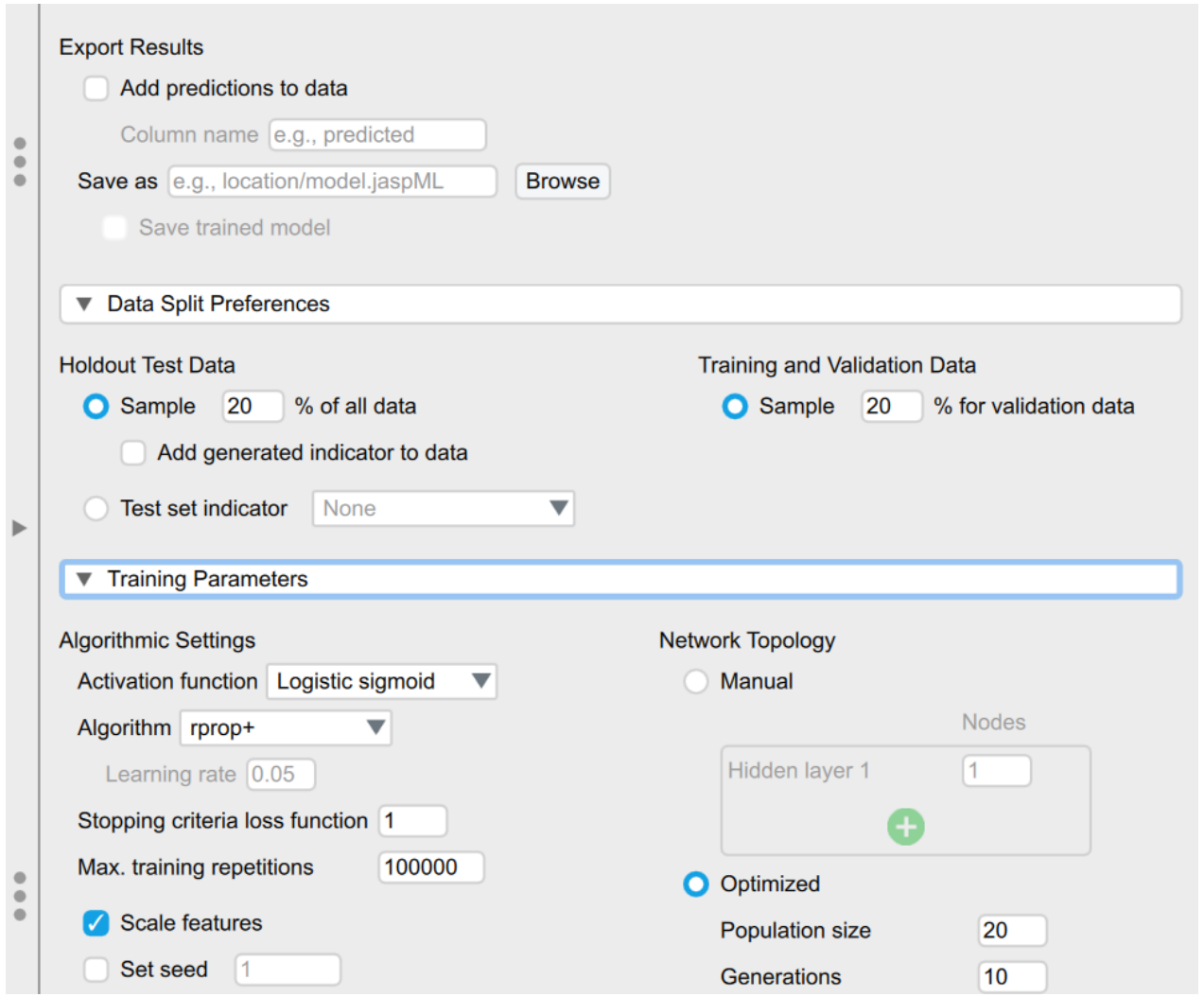
Hands-on practice



1

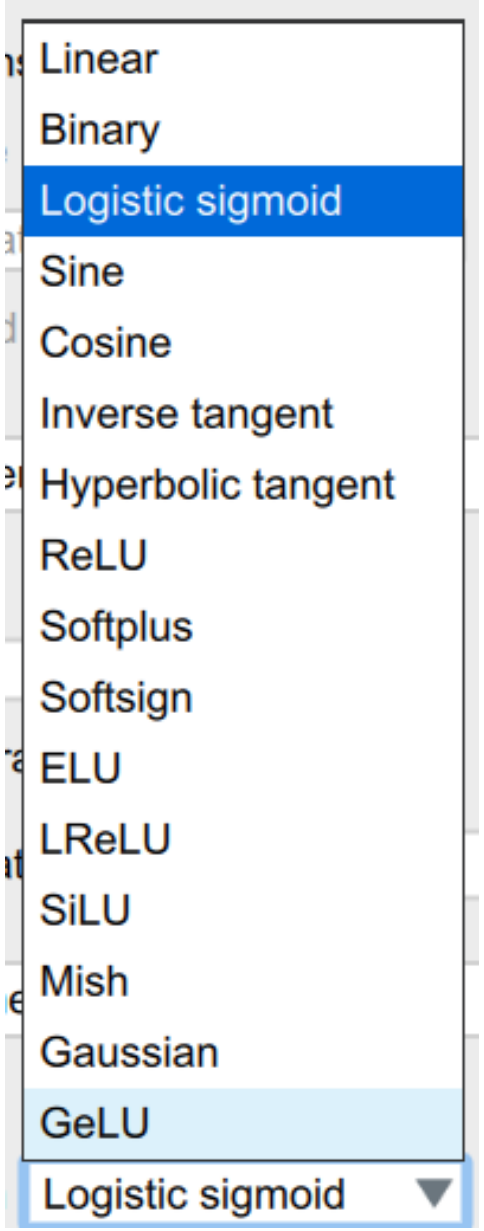

Neural Network Regression
Target
Conscientiousness
Features
Neuroticism
Extraversion
Openness
Agreeableness
Tables
Model performance
Feature importance
Explain predictions
Cases 1 to 5
Network weights
Plots
Data split
Predictive performance
Mean squared error
Activation function
Network structure

2


Export Results
Add predictions to data
Column name e.g., predicted
Save as e.g., location/model.jaspML
Browse
Save trained model
Data Split Preferences
Holdout Test Data
Sample 20 % of all data
Add generated indicator to data
Test set indicator None
Training and Validation Data
Sample 20 % for validation data
Training Parameters
Algorithmic Settings
Activation function Logistic sigmoid
Algorithm rprop+
Learning rate 0.05
Stopping criteria loss function 1
Max. training repetitions 100000
Scale features
Set seed 1
Network Topology
Manual
Nodes
Hidden layer 1 1
Optimized
Population size 20
Generations 10

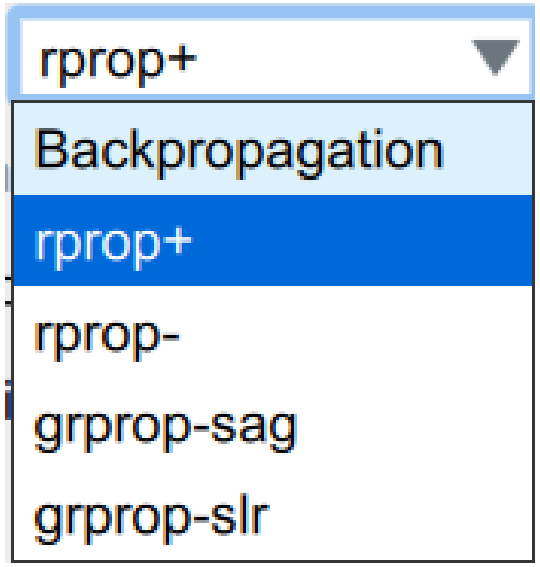
Activation functions

- **Logistic Sigmoid**: Commonly used in the output layer for binary classification problems but is also used in prediction problems.
- **Sine** and **Cosine**: periodic or oscillatory data
- **Inverse Tangent** (tanh): Popular choice for hidden layers in neural networks. Outputs are zero-centered, which can help with faster convergence.
- **ReLU** (Rectified Linear Unit): Most commonly used in hidden layers of deep neural networks. Efficient computation, helps mitigate the vanishing gradient problem, and promotes sparse activation.
- **LReLU (Leaky ReLU)** Used to mitigate the dying ReLU problem. Allows a small, non-zero gradient when the unit is not active.



Algorithms

- **Backpropagation**: The standard algorithm for training neural networks.
- **rprop+** (Resilient Propagation) : An improved version of the basic Rprop algorithm. Faster convergence than standard backpropagation and less sensitive to the learning rate.
- **rprop**: A variant of Rprop without weight-backtracking.
- **grprop-sag** (Gradient Resilient Propagation with Stochastic Average Gradient) : Combines the principles of Rprop with the stochastic average gradient method.
- **grprop-slr** (Gradient Resilient Propagation with Stochastic Line Search): An advanced variant of Rprop that incorporates stochastic line search.

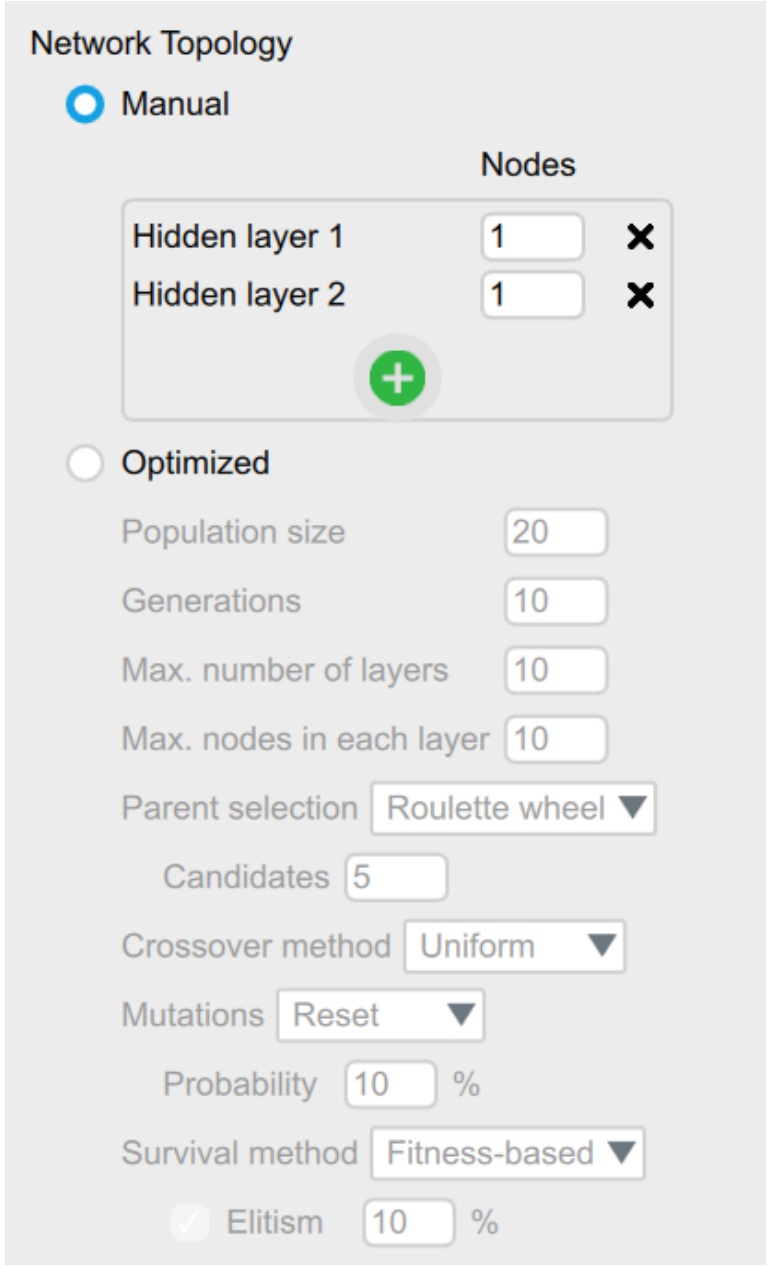


Network topology refers to the **structure** and **organization** of the neural network, including the **number** of **layers**, the number of **nodes** in each layer, and the **connections** between nodes.

If you want to set the Network Topology:

Start Simple: Begin with **one or two hidden** layers and increase them if needed.

Nodes per Layer: There is **no fixed rule**, but a common heuristic is to start with a number of nodes in the hidden layers that is between the number of input nodes and the number of output nodes. For instance, if you have 4 input nodes, you might start with 4 or fewer nodes in the first hidden layer.



Algorithmic Settings

Activation function: ReLU

Algorithm: rprop+

Learning rate: 0.05

Stopping criteria loss function: 1

Max. training repetitions: 100000

Network Topology

Manual

Nodes

Hidden layer 1: 4

Optimized

Model Performance Metrics

	Value
MSE	0.164
RMSE	0.405
MAE / MAD	0.312
MAPE	9.99%
R^2	0.074

Algorithmic Settings

Activation function: ReLU

Algorithm: rprop+

Learning rate: 0.05

Stopping criteria loss function: 1

Max. training repetitions: 100000

Network Topology

Manual

Nodes

Hidden layer 1: 3

Optimized

Model Performance Metrics

	Value
MSE	0.159
RMSE	0.399
MAE / MAD	0.305
MAPE	9.81%
R^2	0.048

Activation function: ReLU

Algorithm: rprop+

Learning rate: 0.05

Stopping criteria loss function: 1

Max. training repetitions: 100000

Manual

Nodes

Hidden layer 1: 1

Hidden layer 2: 1

Model Performance Metrics

	Value
MSE	0.143
RMSE	0.378
MAE / MAD	0.292
MAPE	9.42%
R^2	0.126

Activation function: Logistic sigmoid
Algorithm: rprop+
Learning rate: 0.05
Stopping criteria loss function: 1
Max. training repetitions: 100000

Manual

Nodes	
Hidden layer 1	8
Hidden layer 2	4

Model Performance Metrics

	Value
MSE	0.141
RMSE	0.375
MAE / MAD	0.297
MAPE	9.58%
R^2	0.137

Activation function: Logistic sigmoid
Algorithm: rprop+
Learning rate: 0.05
Stopping criteria loss function: 1
Max. training repetitions: 100000

Manual

Nodes	
Hidden layer 1	4
Hidden layer 2	2

Model Performance Metrics

	Value
MSE	0.139
RMSE	0.373
MAE / MAD	0.293
MAPE	9.48%
R^2	0.143

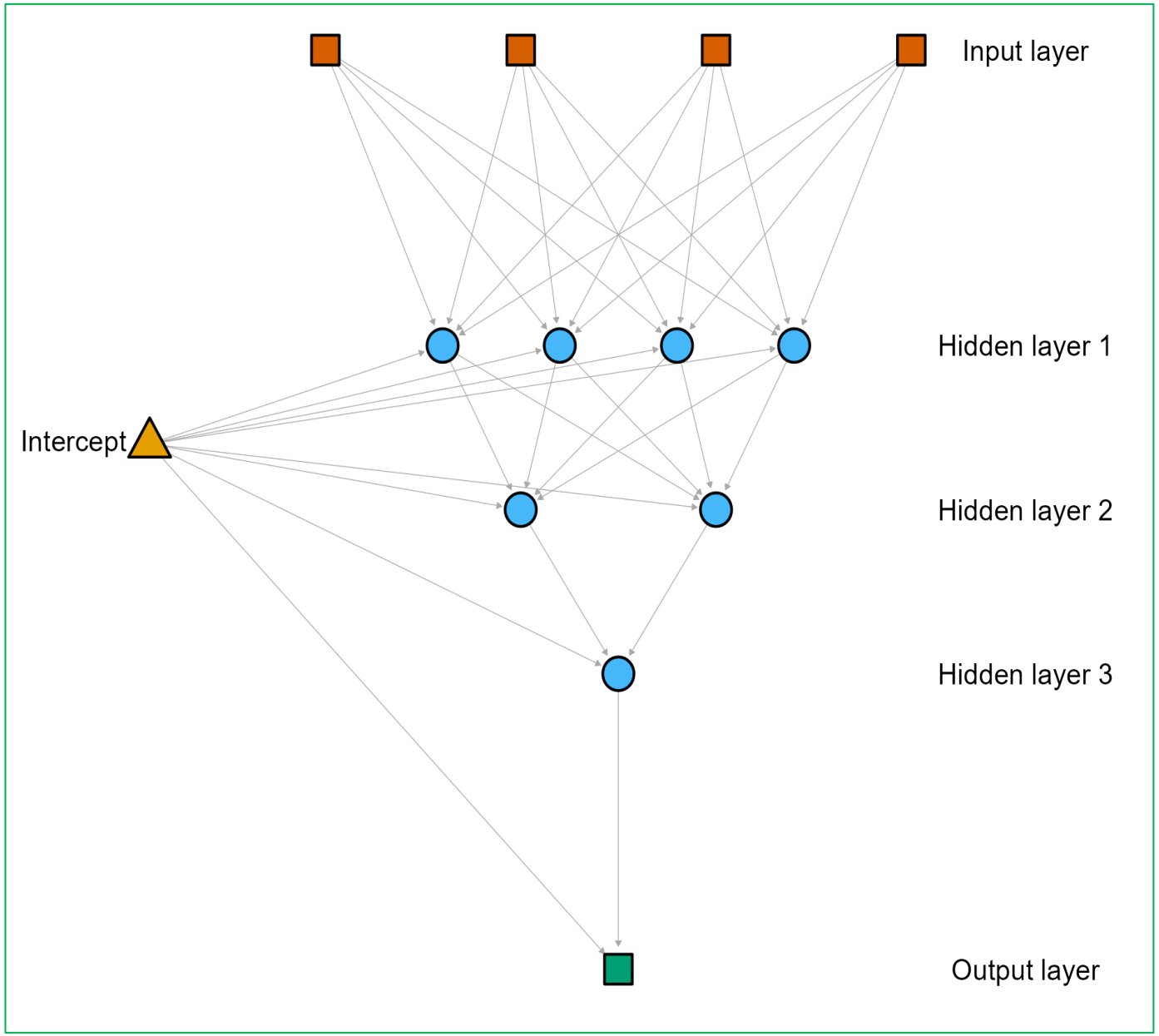
Activation function: Logistic sigmoid
Algorithm: rprop+
Learning rate: 0.05
Stopping criteria loss function: 1
Max. training repetitions: 100000

Manual

Nodes	
Hidden layer 1	4
Hidden layer 2	2
Hidden layer 3	1

Model Performance Metrics

	Value
MSE	0.137
RMSE	0.37
MAE / MAD	0.286
MAPE	9.29%
R^2	0.157



Model Performance Metrics

	Value
MSE	0.137
RMSE	0.37
MAE / MAD	0.286
MAPE	9.29%
R^2	0.157

	Nodes
Hidden layer 1	4
Hidden layer 2	2
Hidden layer 3	1

If you want to use the optimized topology, then find the best configurations:

Algorithmic Settings

Activation function: Logistic sigmoid

Algorithm: rprop+

Learning rate: 0.05

Model Performance Metrics

	Value
MSE	0.138
RMSE	0.371
MAE / MAD	0.291
MAPE	9.41%
R^2	0.149

Algorithmic Settings

Activation function: Hyperbolic tangent

Algorithm: rprop+

Learning rate: 0.05

Model Performance Metrics

	Value
MSE	0.197
RMSE	0.444
MAE / MAD	0.335
MAPE	10.88%
R^2	0.009

Optimized

Population size: 20

Generations: 10

Max. number of layers: 10

Max. nodes in each layer: 10

Parent selection: Roulette wheel

Candidates: 5

Crossover method: Uniform

Mutations: Reset

Probability: 10 %

Survival method: Fitness-based

Elitism: 10 %

3-College Success_Better regression&poor NN*

Activation function: Logistic sigmoid
Algorithm: rprop+
Learning rate: 0.05
Stopping criteria loss function: 1
Manual
Nodes
Hidden layer 1: 1

Model Performance Metrics	Value
MSE	0.442
RMSE	0.665
MAE / MAD	0.536
MAPE	19.45%
R^2	0.3

Activation function: Logistic sigmoid
Algorithm: rprop+
Learning rate: 0.05
Stopping criteria loss function: 1
Manual
Nodes
Hidden layer 1: 1
Hidden layer 2: 1

Model Performance Metrics	Value
MSE	0.432
RMSE	0.657
MAE / MAD	0.517
MAPE	18.64%
R^2	0.279

Activation function: Logistic sigmoid
Algorithm: rprop+
Learning rate: 0.05
Stopping criteria loss function: 1
Max. training repetitions: 100000
Manual
Nodes
Hidden layer 1: 1
Hidden layer 2: 1
Hidden layer 3: 1

Model Performance Metrics	Value
MSE	0.436
RMSE	0.66
MAE / MAD	0.523
MAPE	18.96%
R^2	0.245

Activation function: Logistic sigmoid

Algorithm: rprop+

Learning rate: 0.05

Stopping criteria loss function: 1

Max. training repetitions: 100000

Manual

Nodes	
Hidden layer 1	5
Hidden layer 2	3
Hidden layer 3	1

Model Performance Metrics	Value
MSE	0.373
RMSE	0.611
MAE / MAD	0.48
MAPE	16.84%
R^2	0.385

Activation function: Logistic sigmoid

Algorithm: rprop+

Learning rate: 0.05

Stopping criteria loss function: 1

Max. training repetitions: 100000

Manual

Nodes	
Hidden layer 1	5
Hidden layer 2	5
Hidden layer 3	3
Hidden layer 4	1

Model Performance Metrics ▼	Value
MSE	0.484
RMSE	0.696
MAE / MAD	0.583
MAPE	21.92%
R^2	0.332

Thank you!

Vahid Aryadoust

National Institute of Education

Nanyang Technological University

Singapore

Q-Q Plot

You want to see if it follows a normal distribution. You would:

1. Sort the data set.
2. Calculate the **quantiles** of the data set (e.g., the 0.1, 0.2, ..., 0.9 quantiles).
3. Calculate the corresponding quantiles of the normal distribution (e.g., using the inverse of the normal CDF).
4. Plot these data quantiles against the theoretical quantiles.