DIGITAL FINANCE

This project has received funding from the Horizon Europe research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 101119635

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

State Secretariat for Education, Research and Innovation SERI

Funded by the European Union

MARIE CURIE ACTIONS

DIGITAL

White Box AI - Intrinsic Explainability

Faizan Ahmed

DIGITAL

Funded by
the European Union

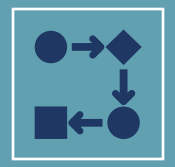
Reading

- **Mandatory Reading Material**
  - Molnar, Christoph. *Interpretable machine learning*.2020.[Section 6-11] https://christophm.github.io/interpretable-ml-book/
  - Explainable AI with Python Chapter "Intrinsic Explainable Models" https://link.springer.com/chapter/10.1007/978-3-030-68640-6_3 [Access through University Library]
- **Recommended Reading Material**
  - Lipton, Zachary C. "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." *Queue* 16.3 (2018): 31-57. https://arxiv.org/abs/1606.03490
  - **If you wanted to know a lot more:**
  - Fisher, Aaron, Cynthia Rudin, and Francesca Dominici. "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously." http://arxiv.org/abs/1801.01489 (2018).
  - Wei, Pengfei, Zhenzhou Lu, and Jingwen Song. "Variable importance analysis: a comprehensive review." Reliability Engineering & System Safety 142 (2015): 399-432
  - Kim, Been, Rajiv Khanna, and Oluwasanmi O. Koyejo. "Examples are not enough, learn to criticize! Criticism for interpretability." Advances in Neural Information Processing Systems (2016). [Very mathematical]
    - The talk is interesting: https://www.youtube.com/watch?v=bQfYRcXc9F0&ab_channel=MicrosoftResearch

DIGITAL

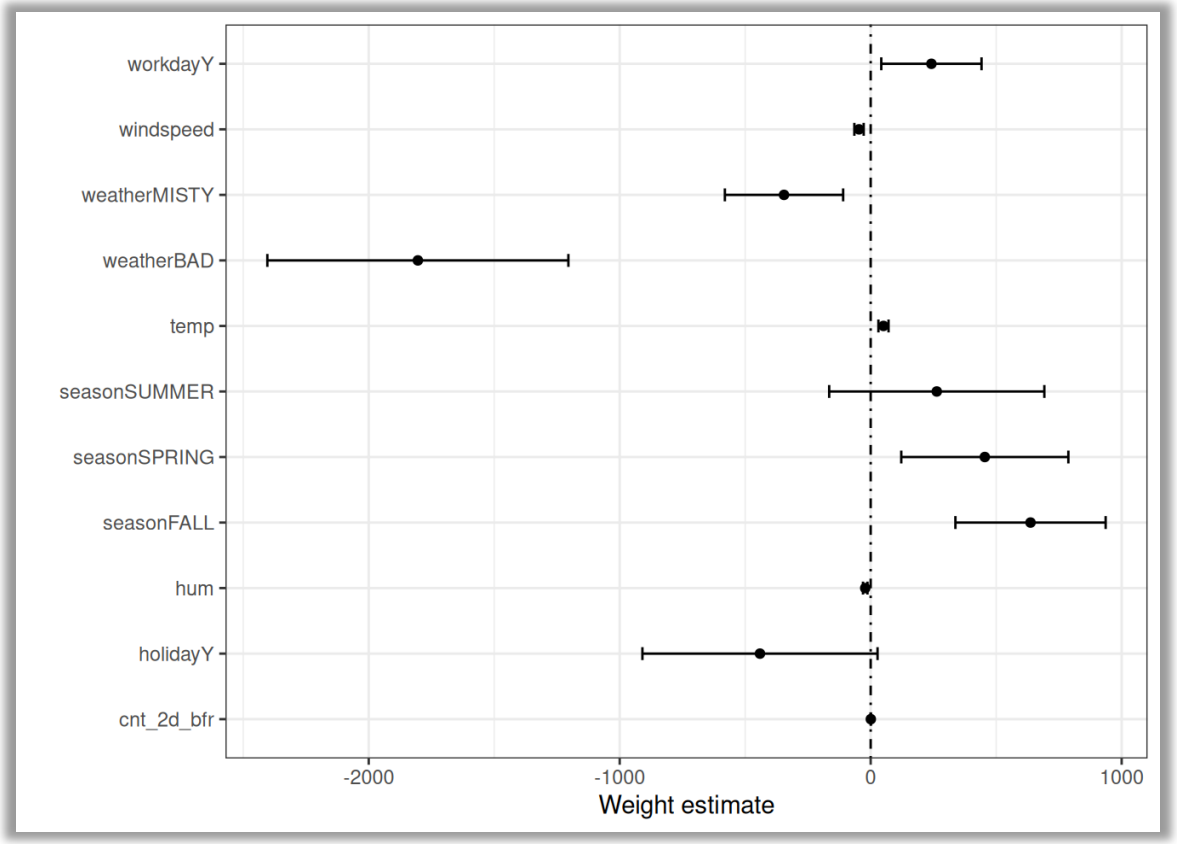
Intrinsically Interpretable Models

Models that are interpretable by design.



No post-processing steps are needed to achieve interpretable.

DIGITAL



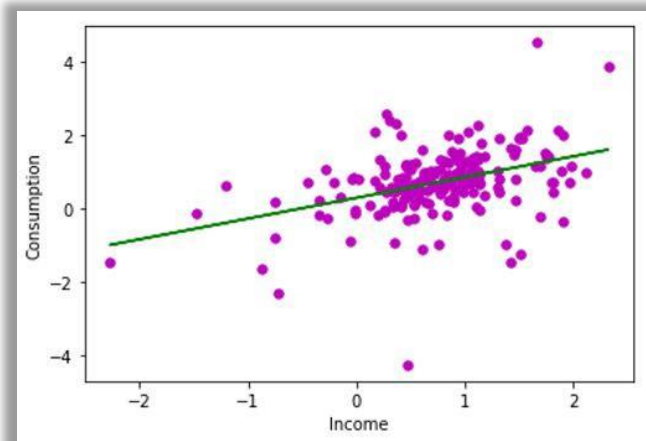
Linear Regression: Interpretations

- **Numerical feature**: An increase of feature x_j by one unit increases the prediction for y by βj units when all other feature values remain fixed.
- **Categorical feature**: Changing feature x_j from the reference category to the other category increases the prediction for y by β_j when all other features remain fixed.
- **Feature Importance:** The importance of a feature in a linear regression model can be measured by the absolute value of its t-statistic.

$$t_{\hat{\beta}_j} = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)}$$

DIGITAL

Linear Regression Example



	Consumption	Income	Production	Savings	Unemployment
1970-01-01	0.615986	0.972261	-2.452700	4.810312	0.9
1970-04-01	0.460376	1.169085	-0.551525	7.287992	0.5
1970-07-01	0.876791	1.553271	-0.358708	7.289013	0.5
1970-10-01	-0.274245	-0.255272	-2.185455	0.985230	0.7
1971-01-01	1.897371	1.987154	1.909734	3.657771	-0.1
...	...	...	...	...	...
2015-07-01	0.664970	0.801663	0.380606	3.180930	-0.3
2015-10-01	0.561680	0.740063	-0.845546	3.482786	0.0
2016-01-01	0.404682	0.519025	-0.417930	2.236534	0.0
2016-04-01	1.047707	0.723721	-0.203319	-2.721501	-0.1
2016-07-01	0.729598	0.644701	0.474918	-0.572858	0.0

$$Consumption = a_1 \text{Income} + a_2 \text{Production} + a_3 \text{Savings} + a_4 Unemployment + \text{b}$$

Intercept	0.2673
Income	0.7145
Production	0.0459
Savings	-0.0453
Unemployment	-0.2048

DIGITAL

Property	Assessment
Completeness	Full completeness achieved without the need of trading- off with interpretability being an intrinsic explainable model
Expressive power	Correlation coefficients provide a direct interpretation of the linear regression weights
Translucency	High, we can look directly at the internals to provide explanations
Portability	Low, explanations rely specifically on linear regression machinery
Algorithmic complexity	Low, no need of complex methods to generate explana- tion
Comprehensibility	Good level of human-understandable explanations to build as much confidence as possible

Linear Regression Example

Further reading: **Sparse linear models**

DIGITAL

4.1. Linear models are not strictly more interpretable than deep neural networks

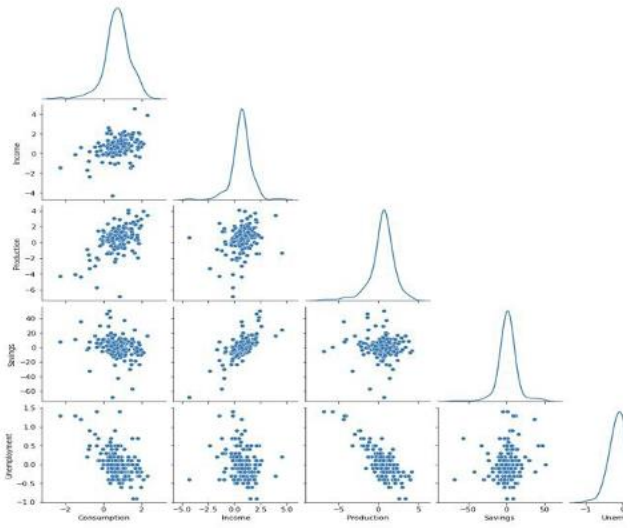
Despite this claim's enduring popularity, its truth content varies depending on what notion of interpretability we employ. With respect to *algorithmic transparency*, this claim seems uncontroversial, but given high dimensional or heavily engineered features, linear models lose *simulatability* or *decomposability*, respectively.

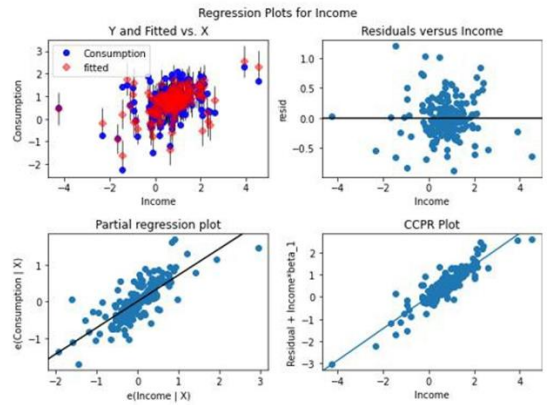
When choosing between linear and deep models, we must often make a trade-off between *algorithic transparency* and *decomposability*. This is because deep neural networks tend to operate on raw or lightly processed features. So if nothing else, the features are intuitively meaningful, and post-hoc reasoning is sensible. However, in order to get comparable performance, linear models often must operate on heavily hand-engineered features. Lipton et al. (2016) demonstrates such a case where linear models can only approach the performance of RNNs at the cost of decomposability.

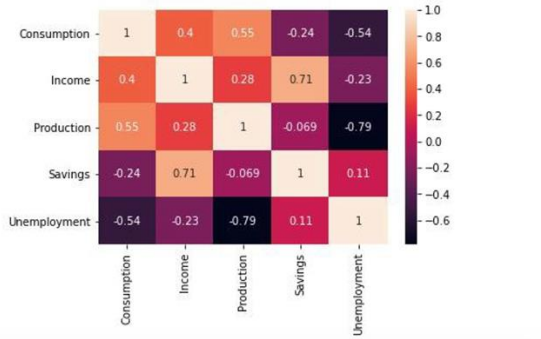
For some kinds of post-hoc interpretation, deep neural networks exhibit a clear advantage. They learn rich representations that can be visualized, verbalized, or used for clustering. Considering the desiderata for interpretability, linear models appear to have a better track record for studying the natural world but we do not know of a theoretical reason why this must be so. Conceivably, post-hoc interpretations could prove useful in similar scenarios.

Source: https://arxiv.org/pdf/1606.03490

Are Linear Model Really Interpretable?







DIGITAL

Logistic Regression

- Linear Regression Models do not work well for Classification
- Probability → reflecting the confidence of the output and classification.

$$P(Y=1) = \frac{1}{\exp\left(-(\beta_0 + \beta_1 x_1 \ + \cdots + \beta_p x_p\ )\right)}$$

$$\left(\frac{P(Y=1)}{P(Y=0)}\right) = odds = exp\left(\beta_0 + \beta_1 x_1 \ + \cdots + \beta_p x_p\ \right)$$

How the prediction changes when one of the features x_k is changed by 1 unit.

$$\frac{odds(x_k+1)}{odds(x_k)} = \frac{exp\left(\beta_0 + \beta_1 x_1 \ + \cdots + \beta_k\,(x_k+1) + \cdots + \beta_p x_p\ \right)}{exp\left(\beta_0 + \beta_1 x_1 \ + \cdots + \beta_k\,(x_k) + \cdots + \beta_p x_p\ \right)} = \exp(\beta_k)$$

DIGITAL

Logistic Regression

- **Numerical feature:** If you increase the value of feature x_k by one unit, the estimated odds change by a factor of $exp(x_j)$.

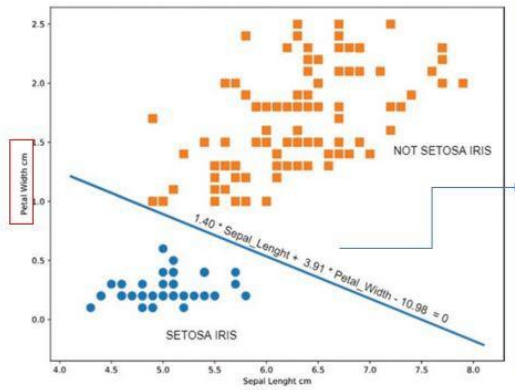
- Binary categorical feature: One of the two values of the feature is the reference category (in some languages, the one encoded in 0). Changing the feature x_k from the reference category to the other category changes the estimated odds by a factor of $\exp(\beta_k)$.

- Intercept β_0: When all numerical features are zero and the categorical features are at the reference category, the estimated odds are $\exp(\beta_0)$). The interpretation of the intercept weight is usually not relevant.

DIGITAL

https://christophm.github.io/interpretable-ml-book/logistic.html

Logistic Regression: In Practice



$$\log\frac{P(Y=\text{Setosa})}{P(Y=\text{non-Setosa})} = \log\text{odds}(Y=\text{Setosa}) = m_0 + m_1x_1 + m_2x_2$$
$$= m_0 + m_1(\text{Sepal Length}) + m_2(\text{Petal Width})$$

$$\frac{\text{odds}(x_1+1)}{\text{odds}(x_1)} = \exp(m_1)x_1 = \text{sepal length}$$

$$\frac{\text{odds}(x_2+1)}{\text{odds}(x_2)} = \exp(m_2)x_2 = \text{petal width}$$

	Setosa	*Versicolor*	*Virginica*
Sepal length	1.44936119	1.61875962	0.21035554
Sepal width	4.09477593	0.20667587	0.20414071
Petal length	0.11623966	1.55210045	11.00944064
Petal width	0.38491152	0.33653155	8.63283183

- **Sepal Length**
  A one-unit increase in sepal length decreases the odds of being Setosa by a factor of

$$\exp(-1.40) \approx 0.25$$

(i.e., odds are reduced to 25% of what they were before, holding petal width constant)

- **Petal Width**
  A one-unit increase in petal width decreases the odds of being Setosa by a factor of

$$\exp(-3.91) \approx 0.02$$

(a strong negative effect , odds drop to just 2% of previous odds, holding sepal length constant)

DIGITAL

Logistic Regression

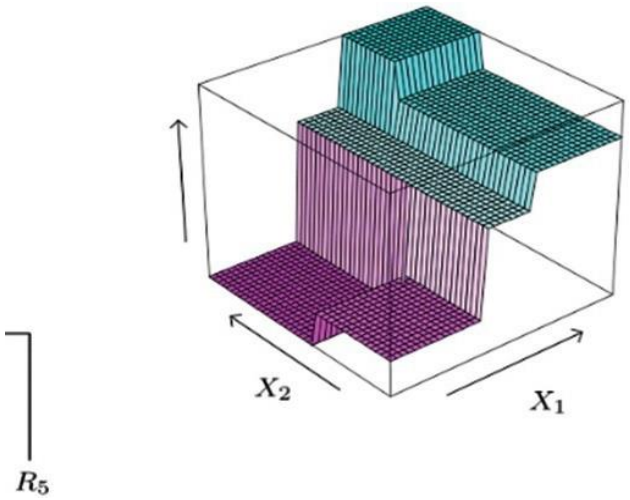
Property	Assessment
Completeness	Full completeness achieved without the need of trading-off with interpretability being an intrinsic explainable model
Expressive power	Less than linear regression case. Interpretation of coefficients is not so straightforward
Translucency	As any intrinsic explainable model, we can look at the internals. Weights are used to provide explanations but not so directly as in linear regression case
Portability	Method is not portable, specific for logistic regression
Algorithmic complexity	Low but not trivial as in linear regression case
Comprehensibility	Explanations are human understandable also for not technical people

DIGITAL

Decision Trees

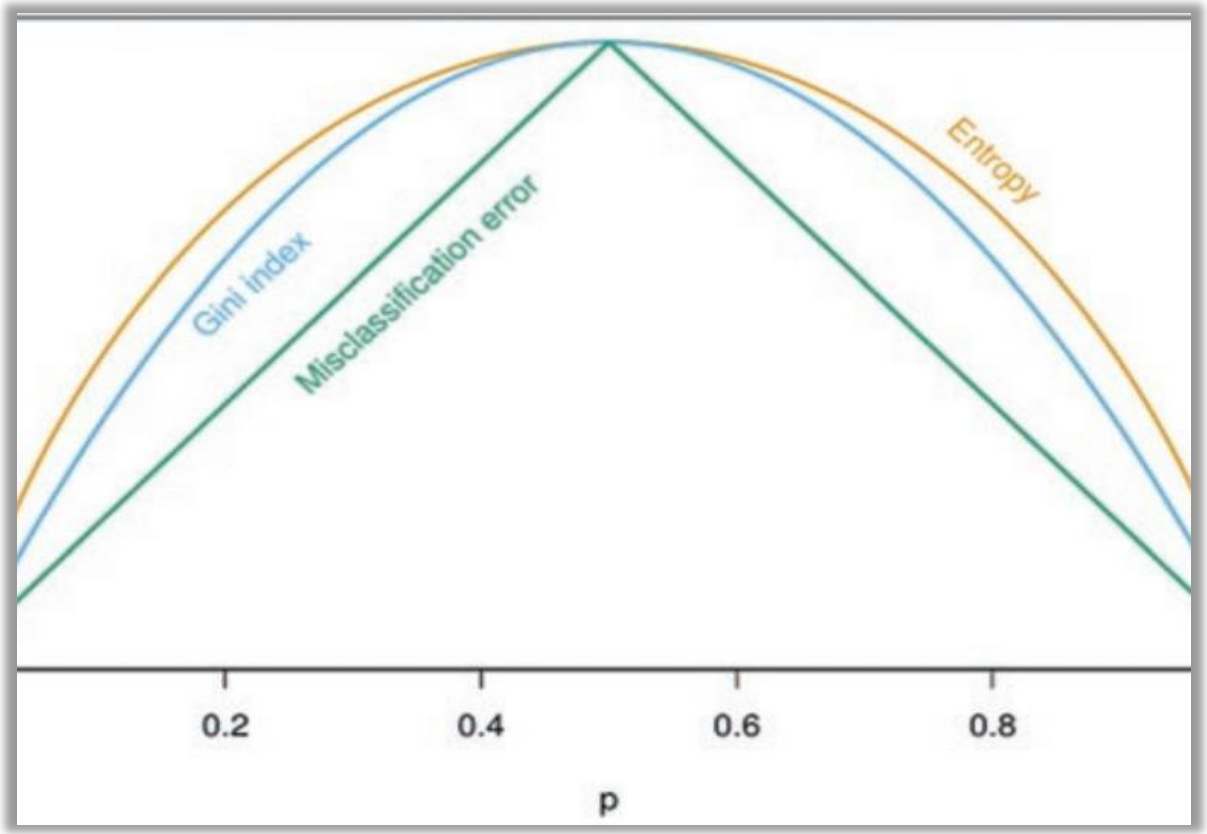
- Regression models fails if:
  - ❑ there is a non-linear relations
  - ❑ And features interact with each other

- **Decision Trees**
  - Categorizes data through repeated splits based on feature values.
  - Forms groups that predict outcomes in final nodes.
  - Data is split multiple times creating subsets.
  - Final subsets are called terminal or leaf nodes.
  - Intermediate subsets are known as internal or split nodes.
  - Outcome prediction in leaf nodes uses average outcome of training data.
  - CART (Classification and Regression Trees) is the most popular.



DIGITAL

Decision Trees - Different Methods to Split



Impurity quantification:

- Gini equation: $1 - \sum_{i=1}^{C}(p_i)^2$
- Shannon Entropy: $\sum_{i=1}^{C} -p_i \log_2(p_i)$
- Classification Error: $1 - \max(p_i)$
- C is the total number of classes, p_i is the probability of a random observation being class i in the remaining observation

DIGITAL

Decision Trees

Decision Tree Interpretation

- Start at the **root node** and follow the branches (edges).
- Each edge represents a **condition** (e.g., "Feature ≤ Threshold").
- **All conditions are connected with AND.**
- At the **leaf node**, the outcome is the **predicted value** (e.g., mean of instances in that node).

Feature Importance in Decision Trees

- For each feature:
- Check how much it **reduced variance** or **Gini index** at its splits.
- Sum all reductions where the feature was used.
- Scale total importance values to **100%.**
- Result: Each feature's importance = its **share of overall model impact**.

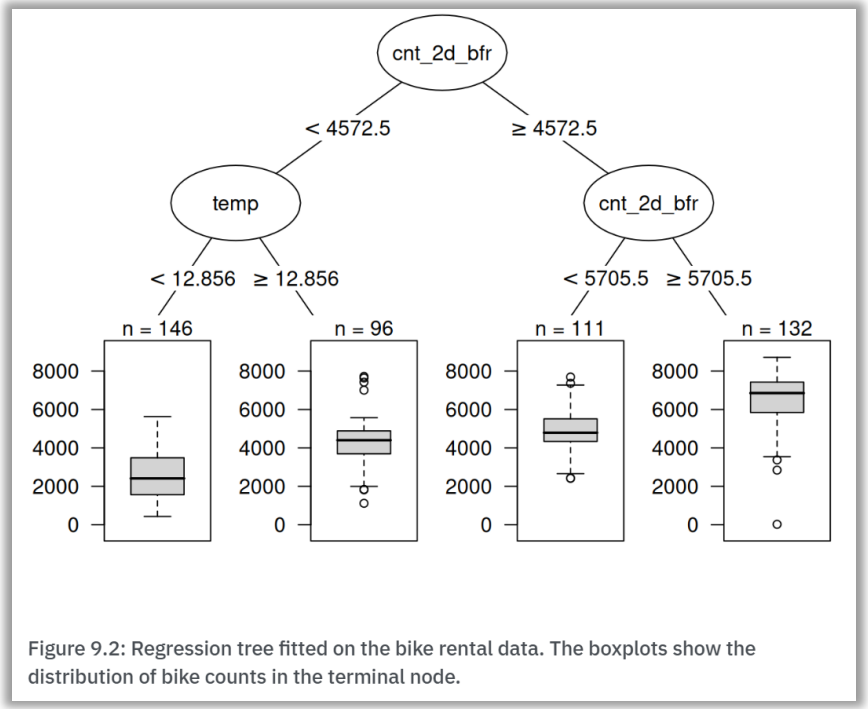
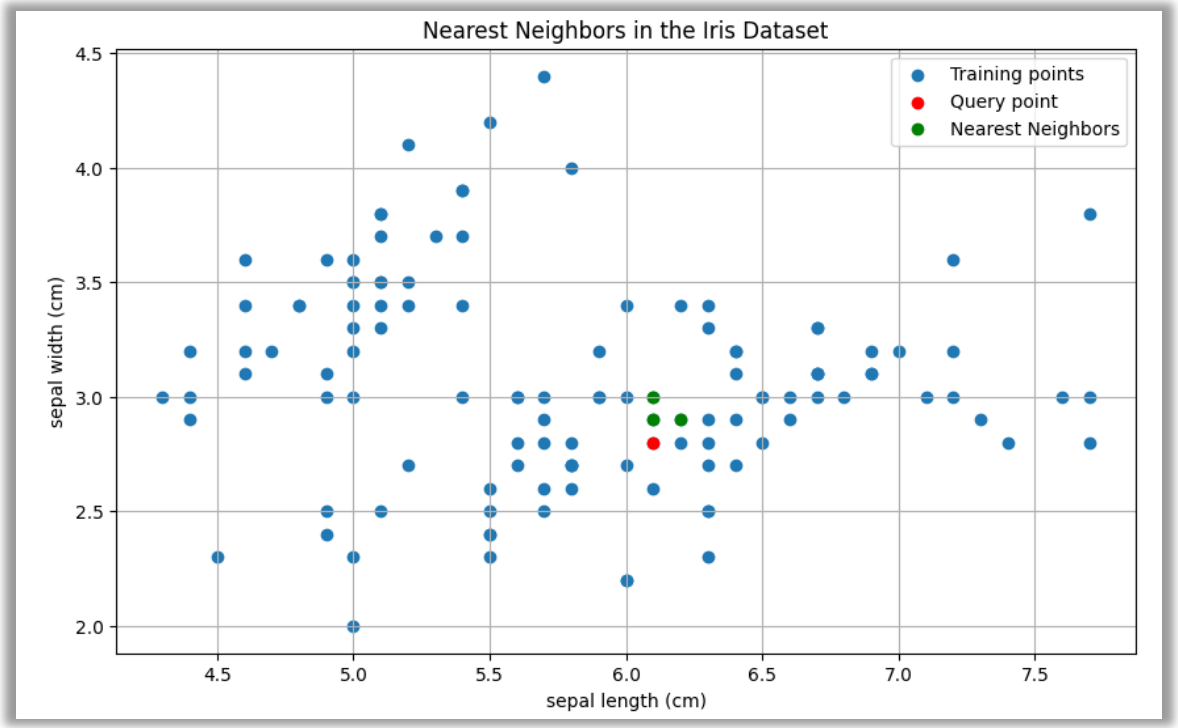


Figure 9.2: Regression tree fitted on the bike rental data. The boxplots show the distribution of bike counts in the terminal node.

DIGITAL

Properties of explanations

Property	Assessment
Completeness	Full completeness achieved without the need of trading off with interpretability being an intrinsic explainable model
Expressive power	High expressive power; in fact DTs mimic to some extent human reasoning
Translucency	Intrinsic explainable easy to guess results
Portability	In fact many models are derived from decision trees such as Random Forest and boosted trees so DT results can be incorporated in such models
Algorithmic complexity	Decision trees are NP-complete, but we resort to heuristic for fast evaluation
Comprehensibility	Easy explanations to humans



K-Nearest Neighbors (KNN)

- KNN use the nearest neighbors of a data point for prediction.
  - **Regression:** Takes the average outcome of the neighbors.
  - **Classification:** Assigns the most common class of the nearest neighbors.
- Selecting the right number of neighbors (k).
- Choosing the distance metric to define the neighborhood.

DIGITAL

Properties of explanations

Property	Assessment
Completeness	Full completeness achieved without the need of trading-off with interpretability being an intrinsic explainable model
Expressive power	High expressive power in terms of counterfactual and con-trastive explanations
Translucency	Intrinsic explainable easy to guess results
Portability	KNN has a unique class in its own not portable
Algorithmic complexity	Simple training, complex inference step
Comprehensibility	Easy explanations to humans

DIGITAL

Current trends and challenges

Model (Year)	Interpretability Approach	Application Domain
IGANN (2024)	Additive neural network (shape functions) + boosting (ELM-based training) for high accuracy **and** transparency.	Tabular data (e.g. productivity, credit, recidivism)
tiSFM (2023)	CNN architecture with **motif-based** filters & layers – parameters directly correspond to sequence motifs.	Genomics (DNA/RNA functional sequence modeling)
MoE-X (2025)	**Mixture-of-Experts** layer redesigned to be wide & sparse for disentangled neurons; uses ReLU experts + sparsity-aware routing.	NLP (language modeling, tested on chess moves and text)
InterpretCC (2024)	**Conditional computation** with feature-level gating or group-level expert routing – only human-relevant features/groups activated per sample.	Tabular, time-series, text (human-centric domains like education, health)
IDEAL (2025)	**Prototype-based** classification using frozen foundation model features; classifies by similarity to learned prototypes.	Vision (image classification, transfer & continual learning)
WYM (Why Match?) (2023)	**Decision units** (paired or unpaired feature tokens) as atomic inputs; interpretable matcher computes each unit's impact on a match/non-match decision.	Data integration (entity matching across databases)
B-cos Networks (2022)	Standard CNN layers replaced by **B-cos transforms** that enforce weight–input alignment; the network reduces to a single linear mapping aligned with meaningful features.	Vision (image recognition – e.g. integrated into ResNet, DenseNet on ImageNet)

DIGITAL

DIGITAL

Funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Horizon Europe: Marie Skłodowska-Curie Actions. Neither the European Union nor the granting authority can be held responsible for them.

DIGITAL

This project has received funding from the European Union's Horizon Europe research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101119635