LEIC-T 2024/2025
Aprendizagem - Machine Learning
Homework I

Group 36
Francisco Uva – 106340
Pedro Pais - 107482

I) Correlation

a)

- **Pearson Correlation**

X_1 = [-4, -2, 0, 2, 4]

X_2 = 0.25 * X_1

X_2 = [-1, -0.5, 0, 0.5, 1]

Mean(X_1) = 0

Mean(X_2) = 0

$\text{Cov}(X_1, X_2) = \frac{\sum(X_1-\overline{X_1})(X_2-\overline{X_2})}{n} = \frac{10}{4} = 2.5$

Standard Deviation $S_1 = \sqrt{\frac{40}{4}} = \sqrt{10}$

Standard Deviation $S_2 = \sqrt{\frac{2.5}{4}} = \sqrt{0.625}$

Corr(X_1, X_2) = $\frac{\text{Cov}(X_1,X_2)}{S_1 * S_2} = \frac{10}{\sqrt{40*2.5}} = 1$

- **Spearman's Rank Correlation**

X_1 = [−4, −2, 0, 2, 4] converts to ranks [1, 2, 3, 4, 5]

X_2 = [−1, −0.5, 0, 0.5, 1] converts to ranks [1, 2, 3, 4, 5]

Since the ranks are identical and there is no difference in rank positions, the Spearman's rank correlation is calculated using the Pearson formula on these ranks, resulting in a correlation coefficient of 1.

- **Conclusion**

Both the Pearson and Spearman's rank correlation coefficients between X_1 and X_2 are 1, indicating a perfect linear relationship. This equivalence arises because the transformation from X_1 to X_2 is linear, thus preserving the order of data points and ranks, which in turn yields identical correlation values.

LEIC-T 2024/2025
Aprendizagem - Machine Learning
Homework I

Group 36
Francisco Uva – 106340
Pedro Pais - 107482

b)

- **Pearson Correlation**

  X_1 = [-4, -2, 0, 2, 4]

  X_2 = [0, 0, 1, 1, 1]

  Mean(X_1) = 0

  Mean(X_2) = 0.6

  $\text{Cov}(X_1, X_2) = \frac{\sum(X_1-\overline{X_1})(X_2-\overline{X_2})}{n} = \frac{6}{4} = 1.5$

  Standard Deviation $S_1 = \sqrt{\frac{40}{4}} = \sqrt{10}$

  Standard Deviation $S_2 = \sqrt{\frac{1.2}{4}} = \sqrt{0.3}$

  Corr(X_1, X_2) = $\frac{\text{Cov}(X_1,X_2)}{S_1 * S_2} = \frac{6}{\sqrt{40*1.2}} = 0.866$

- **Spearman's Rank Correlation**

  X_1 = [−4, −2, 0, 2, 4] converts to ranks [1, 2, 3, 4, 5]

  X_2 = [0, 0, 1, 1, 1] the ranks are:

  - The values 0,0 receive the rank 1.5 (average of rank positions 1 and 2).
  - The values 1,1,1 receive the rank 4 (average of rank positions 3, 4, and 5).

  X_2 = [0, 0, 1, 1, 1] converts to ranks [1.5, 1.5, 4, 4, 4]

  $\rho = 1 - \frac{6(2.5)}{5(25-1)}$ = 1 – 0.125 = 0.875

- **Conclusion**

  The Pearson correlation is approximately 0.866, and the Spearman's rank correlation is 0.875. These values are similar but not exactly the same due to the nonlinear transformation applied by the unit step function, which affects the ranks and linear relationship slightly.

LEIC-T 2024/2025
Aprendizagem - Machine Learning
Homework I

Group 36
Francisco Uva – 106340
Pedro Pais - 107482

c)

- **Pearson Correlation**

X₁ = [-4, -2, 0, 2, 4]

- For $X_1 = -4$, $X_2 = \frac{1}{1+e^4} = 0.018$
- For $X_1 = -2$, $X_2 = \frac{1}{1+e^2} = 0.119$
- For $X_1 = 0$, $X_2 = \frac{1}{1+e^0} = 0.5$
- For $X_1 = 2$, $X_2 = \frac{1}{1+e^{-2}} = 0.881$
- For $X_1 = 4$, $X_2 = \frac{1}{1+e^{-4}} = 0.982$

$X_2 = [0.018, 0.119, 0.5, 0.881, 0.982]$

$\text{Mean}(X_1) = 0$

$\text{Mean}(X_2) = 0.5$

$\text{Cov}(X_1, X_2) = \frac{\sum(X_1-\overline{X_1})(X_2-\overline{X_2})}{n} = \frac{5.38}{4} = 1.345$

Standard Deviation $S_1 = \sqrt{\frac{40}{4}} = \sqrt{10}$

Standard Deviation $S_2 = \sqrt{\frac{0.75497}{4}} = \sqrt{0.189}$

$\text{Corr}(X_1, X_2) = \frac{\text{Cov}(X_1,X_2)}{S_1 * S_2} = \frac{5.38}{\sqrt{10*0.754}} = 0.979$

- **Spearman's Rank Correlation**

X₁ = [−4, −2, 0, 2, 4] converts to ranks [1, 2, 3, 4, 5]

X₂ = [0.018, 0.119, 0.5, 0.881, 0.982] converts to ranks [1,2,3,4,5]

Since the ranks are the same, the Spearman's rank correlation, which uses the Pearson formula on the ranks, results in a correlation of 1.

- **Conclusion**

The Pearson correlation of approximately 0.979 and a Spearman's rank correlation of 1 indicate a strong positive relationship between X_1 and X_2 with the transformation via the sigmoid function. The Spearman's correlation reaching 1 highlights the preservation of order and ranking of the data points, despite the non-linear transformation of the sigmoid function.

II) Decision Trees

For simplicity purposes, consider Go for a Walk as G, TV as T and Reading as R from now on.

a)

p(G) = 2/5 p(T) = 1/5 p(R) = 2/5

Log2[x] = Log[x] / Log[2]

I(table) = -2 * 2/5 * Log2[2/5] – 1/5 * Log2[1/5] = 1.52193 bit

E(P) = $\sum_{i=1}^{n} \frac{|C_i|}{|C|} I(C_i)$ gain(P) = I(C) – E(P)

Weekend C_{Yes} = (G,R) C_{No} = (T,R,G)

I(C_{Yes}) = -2 * ½ * Log2[1/2] = 1 bit

I(C_{No}) = -3 * 1/3 * Log2[1/3] = 1.58496 bit

E(Weekend) = 2/5 * 1 + 3/5 * 1.58496 = 1.350976 bit

Gain(Weekend) = 1.52193 – 1.350976 = 0.170954 bit

LEIC-T 2024/2025
Aprendizagem - Machine Learning
Homework I

Group 36
Francisco Uva – 106340
Pedro Pais - 107482

Weather C_{Sunny} = (G) C_{Rain} = (T,R) C_{Cloudy} = (G,R)

I(C_{Sunny}) = 0 bit

I(C_{Rain}) = I(C_{Cloudy}) = -2 * ½ * Log2[1/2] = 1 bit

E(Weather) = 1/5 * 0 + 2/5 *1 + 2/5*1 = 0.8 bit

Gain(Weather) = 1.52193 – 0.8 = 0.72193 bit

Tired C_{No} = (G,R,G) C_{Yes} = (T,R)

I(C_{No}) = -2/3 * Log2[2/3] – 1/3 * Log2[1/3] = 0.918296 bit

I(C_{Yes}) = -2 * ½ * Log2[1/2] = 1

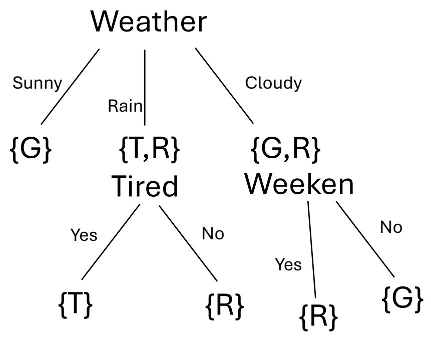
E(Tired) = 3/5 * 0.918296 + 2/5 * 1 = 0.950978 bit

Gain(Tired) = 1.52193 – 0.950978 = 0.570952 bit

Following the ID3 algorithm, Weather is chosen as the root.

b)

Weekend	Tired	What to Do?
No	Yes	TV
No	No	Reading
No	No	Go for a walk
Yes	Yes	Reading



c)

TRUE

		G	T	R
PREDICTE	G	1	0	0
	T	0	1	0
	R	0	0	3

III) Software Experiments

a)

Using a *train_size* of 0.1 (10% of the data for training and 90% for testing), the decision tree achieved an accuracy of 0.83 with a depth of 2. With a *train_size* of 0.9 (90% of the data for training and 10% for testing), the accuracy reached 1.0 with a tree depth of 4.

These differences can be explained by the amount of data available for the model to learn from. With a larger training set (90% of the data), the decision tree model has significantly more examples to learn the underlying patterns in the data, allowing it to build a more complex model (showed by the greater depth) that can achieve higher accuracy. In contrast, a smaller training set (10% of the data) limits the model's ability to learn, resulting in a simpler tree (shallower depth) and lower accuracy.

In conclusion, training with more data generally allows a decision tree to build a more detailed and accurate model, as it can capture more of the variability and complexity inherent in the data. This experiment clearly shows how the size of the training set can significantly change the performance and complexity of a decision tree model.

b)

The accuracy is lower without using *stratify=y* because this command ensures that the training and test sets have approximately the same percentage of samples of each class label as the original dataset. Without stratifying, the split may result in a non-representative distribution of classes, particularly in datasets where some classes are underrepresented.

In conclusion, omitting the *stratify=y* parameter can lead to training and test datasets that do not accurately reflect the class distribution of the original data. This can adversely affect the model's learning, making it less effective at predicting outcomes for underrepresented classes. The resulting model, therefore, might not generalize well to new data, leading to decreased accuracy as observed in this experiment. This highlights the importance of using stratification in train-test splits to ensure balanced class representation, especially in datasets with uneven class distributions.