

Data Set 1:

(a) URL of the webpage to access the data set: <https://archive.ics.uci.edu/ml/datasets/Iris>

(b) Brief description of the data set:

The Iris data set is a classic and widely used data set in machine learning. It contains measurements of four attributes (sepal length, sepal width, petal length, and petal width) for 150 iris flowers from three different species: setosa, versicolor, and virginica. The data set is often used for classification tasks and pattern recognition algorithms.

Number of objects: 150 (iris flowers)

Number of attributes: 4 (sepal length, sepal width, petal length, petal width)

Attribute types: Numeric (floating-point values in cm)

Class Labels: 3 (Iris Setosa, Iris Versicolour, Iris Virginica)

(c) Knowledge that may be mined from this data set:

The Iris data set can be used to explore patterns and relationships between the measured attributes and the different species of iris flowers. This is perhaps the best-known database to be found in the pattern recognition literature. It is widely used for tasks such as classification and clustering, particularly for tutorial/teaching purposes. It can be mined to understand the characteristics that distinguish one species from another and to build predictive models for classification tasks.

(d) How the knowledge would be useful in some applications:

Analyzing the Iris data set can yield valuable insights that find application in multiple domains, including species identification, horticulture, and botanical research. These insights can be leveraged to create automated algorithms or systems for the accurate classification of iris flowers using their measurements. Furthermore, the knowledge derived from this data set plays a crucial role in enhancing our understanding of plant taxonomy and facilitating the development of effective conservation strategies.

Data Set 2:

(a) URL of the webpage to access the data set: <https://www.kaggle.com/c/titanic/data>

(b) Brief description of the data set:

The Titanic data set is based on the famous Titanic passenger list and contains information about individual passengers aboard the Titanic, including their survival status, demographic attributes, cabin class, ticket fare, and more. The data set is commonly used for predictive modeling and data analysis tasks.

Number of objects: 1,309 (passengers)

Number of attributes: 12

Attribute types: A mix of categorical (e.g., sex, cabin class) and numeric (e.g., age, fare)

(c) Knowledge that may be mined from this data set:

The Titanic data set presents a chance to examine the factors that impacted the survival of passengers aboard the Titanic. It enables the exploration of relationships and associations between attributes like age, gender, cabin class, and survival rates. Furthermore, it facilitates the creation of predictive models that can estimate the probability of survival based on passenger attributes.

(d) How the knowledge would be useful in some applications:

Analyzing the Titanic data set provides practical implications in various fields such as risk assessment, emergency planning, and social science research. By comprehending the factors influencing survival, valuable insights can be gained to enhance safety protocols, optimize emergency response strategies, and gain a deeper understanding of social dynamics during catastrophic events. Moreover, this data set holds significant educational value as it can be utilized to teach data analysis and machine learning techniques effectively.

Paper details

Paper Link - <https://dl.acm.org/doi/10.1145/3534678.3539097>

Title - RCAD: Real-time Collaborative Anomaly Detection System for Mobile Broadband Networks

Authors - Azza H. Ahmed, Michael A. Riegler, Steven A. Hicks, and Ahmed Elmokashfi

Paper session and affiliation - KDD '22: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining

What the problem is, and why is it important and challenging

The paper addresses the problem of predicting and detecting anomalies in broadband networks through an automated network management system. It is important because, with the increase in the number of devices connected to the internet, it becomes challenging to identify anomalies using human operators. This near-continuous flux of data has compounded the complexity of network operation and management

Proposed solution

The paper introduces RCAD, a real-time collaborative anomaly detection framework for mobile networks. It consists of two components: an online distributed unsupervised anomaly detection and prediction system, and a collaborative framework for knowledge sharing and exchanging. The framework utilizes hierarchical temporal memory (HTM), an unsupervised machine intelligence algorithm, for anomaly detection. The online nature of HTM enables continuous learning without the need for retraining. To enhance detection accuracy, a collaborative framework allows probes with fewer anomalies to benefit from others' experiences. Two collaboration approaches, threshold-based model replacement and deep reinforcement learning (DRL)-based model replacement, are presented.

How is the proposed solution evaluated?

The dataset used is collected from 10 probes of a mobile operator in Norway. The measurements tracked are RTT, RSSI, RSRP, and RSRQ. The paper uses Precision, Recall, and F1-score to evaluate the prediction and detection of anomalies. The performance metric is calculated by comparing each prediction sample with annotated ground truth. Using the HTM algorithm an F1 score of 0.7 is achieved

Being a classification problem we would primarily be interested in the following metrics:

Accuracy: Accuracy is the most straightforward metric and measures the overall correctness of the predictions by calculating the ratio of correctly classified instances to the total number of instances.

Precision: Precision calculates the ratio of true positives (correctly predicted positive instances) to the sum of true positives and false positives (incorrectly predicted positive instances). It indicates how many of the predicted positive instances are actually relevant. High precision indicates that the model is correctly identifying exoplanets and minimizing false positives.

Recall (Sensitivity or True Positive Rate): Recall calculates the ratio of true positives to the sum of true positives and false negatives (missed positive instances). It indicates the proportion of actual positive instances that are correctly identified. High recall indicates that the model effectively captures exoplanets and minimizes false negatives.

F1 Score: The F1 score is the harmonic mean of precision and recall. It provides a single metric that balances both precision and recall. The F1 score is useful to find an optimal balance between precision and recall.

The accuracy metric is inadequate for evaluating the performance of a planet detection algorithm due to the imbalanced nature of most exoplanet detection datasets. These datasets typically have a larger number of light curves without any planet signal compared to those with a planet. While high precision ensures that the predicted "planet candidates" are mostly true, it is not a very informative metric. This is because achieving high precision can be done by making only a few "planet candidate" predictions and ensuring their correctness. However, this approach may result in missing many potential planet candidates.

Instead, prioritizing recall is more important for assessing the algorithm's performance in planet detection. Recall measures the proportion of actual planet signals that are correctly identified by the algorithm. It is preferable to accept a higher number of false positives (incorrectly identified planet candidates) rather than missing potential planet signals.

The trade-off between precision and recall, commonly known as the precision-recall tradeoff, is significant. A model with high recall may have a lower precision and vice versa. This trade-off is typically evaluated using the F1 score, which combines precision and recall in a single metric.

In this project, we propose to explore and evaluate several machine learning algorithms to classify stellar lightcurves as containing or not containing exoplanet transits.

The groundbreaking detection of the exoplanetary system PSR1257 + 12 in 1992 marked a significant milestone in understanding planets outside our solar system. Since then, the detection of new exoplanetary systems has become crucial for studying the formation and evolution of planetary systems and investigating conditions for life. Transit photometry,

measuring periodic dips in starlight, has been the most successful method for exoplanet detection. Recent advancements in time domain optical surveys have made transit photometry even more promising. In this project, the group aims to explore machine learning algorithms to automate the identification of exoplanet transits in stellar lightcurves, combining the frontiers of exoplanet science and the power of machine learning.

HW3

Q1

$$A) \text{ lift} = P(\text{ski} \cup \text{bike}) / (P(\text{ski}) * P(\text{bike}))$$

$$P(\text{ski} \cup \text{bike}) = 600/4000 = 6/40$$

$$P(\text{ski}) = 2500/4000 = 25/40$$

$$P(\text{bike}) = 1300/4000 = 13/40$$

substituting the above in the lift measure we get

$$\text{lift} = (6/40) / ((25/40) * (13/40)) = 0.738$$

Thus the correlation between bike and ski is 0.738

B) Given the association rule "bike \Rightarrow ski"

$$\text{support for the rule} = \text{occurrence of both bike and ski} = 600/4000 = 15\%$$

$$\text{confidence} = \text{occurrence of ski given bike} = 600/1300 = 46.15\%$$

Since both the support and confidence are greater than or equal to the threshold, the given rule is a strong association rule.

Q2

A) the unique items in all transactions are {A, B, C, D, H, K, M, R, S, T, X, Z} that is 12 unique items.

$$\text{max number of frequent item sets would be } 12C1 + 12C2 + \dots + 12C12 = (2^{12} - 1) = 4095$$

B)

Step 1: First calculate the 1-itemset using the Apriori algorithm

1-itemset	Frequency	Support
A	3	0.6
B	3	0.6
C	1	0.2
D	2	0.4
H	3	0.6
K	1	0.2
M	1	0.2
R	1	0.2
S	2	0.4
T	3	0.6
X	3	0.6
Z	2	0.4

after applying the support threshold of 0.4 only consider the itemsets {A}, {B}, {D}, {H}, {S}, {T}, {X}, {Z} for the next round.

Step 2: next calculate the 2-itemsets

2-itemset	Frequency	Support
{A, B}	1	0.2

{A, D}	1	0.2
{A, H}	1	0.2
{A, S}	1	0.2
{A, T}	2	0.4
{A, X}	2	0.4
{A, Z}	2	0.4
{B, D}	2	0.4
{B, H}	2	0.4
{B, S}	2	0.4
{B, T}	1	0.2
{B, X}	1	0.2
{B, Z}	0	0
{D, H}	1	0.2
{D, S}	1	0.2
{D, T}	1	0.2
{D, X}	1	0.2
{D, Z}	0	0
{H, S}	1	0.2
{H, T}	2	0.4
{H, X}	2	0.4

{H, Z}	1	0.2
{S, T}	0	0
{S, X}	0	0
{S, Z}	0	0
{T, X}	3	0.6
{T, Z}	2	0.4
{X, Z}	2	0.4

after applying the support threshold of 0.4 only consider the itemsets {A, T}, {A, X}, {A, Z}, {B, D}, {B, H}, {B, S}, {H, T}, {H, X}, {T, X}, {T, Z}, {X, Z}

Step 3: next calculate the 3-itemsets

3-itemset	Frequency	Support
{A, T, X}	2	0.4
{A, T, Z}	2	0.4
{A, X, Z}	2	0.4
{A, T, H}	1	0.2
{A, X, H}	1	0.2
{B, D, H}	1	0.2
{B, D, S}	1	0.2
{B, H, S}	1	0.2
{B, H, T}	1	0.2
{B, H, X}	1	0.2

{H, T, X}	2	0.4
{H, T, Z}	1	0.2
{H, X, Z}	1	0.2
{T, X, Z}	2	0.4

after applying the support threshold of 0.4 only consider the itemsets {A, T, X}, {A, T, Z}, {A, X, Z}, {H, T, X}, {T, X, Z}

Step 4: next calculate the 4-itemsets

4-itemset	Frequency	Support
{A, T, X, Z}	2	0.4
{A, T, X, H}	1	0.2

after applying the support threshold of 0.4 only consider the itemsets {A, T, X, Z}

C) Since the algorithm ran for 4 steps the number of database scans is equal to 4.

There are a total of 25 Candidate itemsets - {A}, {B}, {D}, {H}, {S}, {T}, {X}, {Z}, {A, T}, {A, X}, {A, Z}, {B, D}, {B, H}, {B, S}, {H, T}, {H, X}, {T, X}, {T, Z}, {X, Z}, {A, T, X}, {A, T, Z}, {A, X, Z}, {H, T, X}, {T, X, Z}, {A, T, X, Z}

Q3

A) The total number of candidate 3-Itemsets is the total number of itemsets present in the leaf nodes of the tree is 22.

B) There are 10 possible 3-Itemsets of the transaction {1, 2, 5, 6, 9}

1. {1, 2, 5} -> reaches the leaf node L3
2. {1, 2, 6} -> reaches the lead node L4
3. {1, 2, 9} -> reaches the lead node L4

4. {1, 5, 6} -> reaches the lead node L4
5. {1, 5, 9} -> reaches the lead node L4
6. {1, 6, 9} -> reaches the lead node L5
7. {2, 5, 6} -> reaches the lead node L7
8. {2, 5, 9} -> reaches the lead node L7
9. {2, 6, 9} -> reaches the lead node L8
10. {5, 6, 9} -> reaches the lead node L8

Thus, the leaf nodes visited for the transaction {1, 2, 5, 6, 9} are L3, L4, L5, L6, L7, L8

C) The only candidate transaction is {1, 2, 5} at the leaf node L3.

Q4

A) for finding the largest k k-Itemsets let's apply the Apriori algorithm, considering only the item_category from the table

Step 1: calculate the 1-Itemset

1-Itemset	Frequency	Support
Milk	4	1
Bread	4	1
Pie	3	0.75
Cherry	2	0.5
Cheese	3	0.75
Cereal	2	0.5

applying the min support threshold of 0.7 we keep the 1-Itemsets {Milk}, {Bread}, {Pie}, {Cheese}

Step 2: calculate the 2-Itemset

2-Itemset	Frequency	Support
{Milk, Bread}	4	1
{Milk, Pie}	3	0.75
{Milk, Cheese}	3	0.75
{Bread, Pie}	3	0.75
{Bread, Cheese}	3	0.75
{Pie, Cheese}	2	0.5

applying the min support threshold of 0.7 we keep the 2-Itemsets {Milk, Bread}, {Milk, Pie}, {Milk, Cheese}, {Bread, Pie}, {Pie, Cheese}

Step 3: calculate 3-Itemsets

3-itemset	Frequency	Support
{Milk, Bread, Pie}	3	0.75
{Milk, Bread, Cheese}	3	0.75
{Milk, Pie, Cheese}	2	0.5
{Bread, Pie, Cheese}	2	0.5

applying the min support threshold of 0.7 we keep the 3-Itemsets {Milk, Bread, Pie}, {Milk, Bread, Cheese}

Step 4: calculate 4-Itemset

4-itemset	Frequency	Support
{Milk, Bread, Pie, Cheese}	2	0.5

Thus the candidate k-Itemset with the maximum k is k=3 and the Itemsets are {Milk, Bread, Pie}, {Milk, Bread, Cheese}

Next, finding the association rules for the above 2 Itemsets

The possible association rules are:

1. (Milk, Bread) \Rightarrow Pie with confidence of $3/4 = 0.75$
2. (Bread, Pie) \Rightarrow Milk with confidence of $3/3 = 1$
3. (Milk, Pie) \Rightarrow Bread with confidence of $3/3 = 1$
4. (Milk, Bread) \Rightarrow Cheese with confidence of $3/4 = 0.75$
5. (Bread, Cheese) \Rightarrow Milk with confidence of $3/3 = 1$
6. (Milk, Cheese) \Rightarrow Bread with confidence of $3/3 = 1$

After applying the threshold confidence of 0.8 we are left with the following 4 association rules

1. (Bread, Pie) \Rightarrow Milk [0.75, 1]
2. (Milk, Pie) \Rightarrow Bread [0.75, 1]
3. (Bread, Cheese) \Rightarrow Milk [0.75, 1]
4. (Milk, Cheese) \Rightarrow Bread [0.75, 1]

B) Since there are only 3 customers and X belongs to a customer, hence we combine the transaction T100 and T300, hence we now have only in total 3 Transactions.

Step 1: calculate a 1-Itemset of Unique brand-item_category,

1-Itemset	Frequency	Support
Farmers-Milk	3	1
Sunset-Milk	1	0.3
Wonder-Bread	3	1
Best-Bread	1	0.3
Sweet-Pie	3	1

Sunny-Cherry	1	0.3
Goldenfarm-Cherry	1	0.3
Dairyland-Cheese	3	1
Kings-Cereal	1	0.3
Best-Cereal	1	0.3

applying the min support threshold of 0.7 we keep the 1-Itemsets {Farmers-Milk}, {Wonder-Bread}, {Sweet-Pie}, {Dairyland-Cheese}

Step2: calculate 2-Itemset

2-Itemset	Frequency	Support
{Farmers-Milk, Wonder-Bread}	3	1
{Farmers-Milk, Sweet-Pie}	3	1
{Farmers-Milk, Dairyland-Cheese}	3	1
{Wonder-Bread, Sweet-Pie}	3	1
{Wonder-Bread, Dairyland-Cheese}	3	1
{Sweet-Pie, Dairyland-Cheese}	3	1

applying the min support threshold of 0.7 we keep all the 2-Itemsets

Step 3: calculate 3-itemset

3-Itemset	Frequency	Support
{Farmers-Milk, Wonder-Bread, Sweet-Pie}	3	1
{Farmers-Milk, Wonder-Bread,	3	1

Dairyland-Cheese}		
{Farmers-Milk, Sweet-Pie, Dairyland-Cheese}	3	1
{Wonder-Bread, Sweet-Pie, Dairyland-Cheese}	3	1

applying the min support threshold of 0.7 we keep all the 3-Itemsets

Step 4: calculate 4-itemset

4-itemset	Frequency	Support
{Farmers-Milk, Wonder-Bread, Sweet-Pie, Dairyland-Cheese}	3	1

So the largest k-frequent item is {Farmers-Milk, Wonder-Bread, Sweet-Pie, Dairyland-Cheese} with k= 4.

Q5

A) For Heart Disease $P(\text{Heart Disease} = \text{"yes"}) = 8/12$ and $P(\text{Heart Disease} = \text{"no"}) = 4/12$

Thus the entropy is $\text{Info}(\text{Heart Disease}) = -(4/12) \cdot \log(4/12) - (8/12) \cdot \log(8/12) = 0.918$

Next, calculate the Information-Gain for each of the 4 attributes

1. Attribute - Diabetes

Diabetes	p (positive)	n (negative)	Info(p, n)
Yes	3	2	0.97
No	5	2	0.87

$\text{Info}(\text{Diabetes}) = (5/12) \cdot \text{Info}(3,2) + (7/12) \cdot \text{Info}(5,2) = 0.911$

$\text{Information-Gain}(\text{Diabetes}) = \text{Info}(\text{Heart Disease}) - \text{Info}(\text{Diabetes}) = 0.07$

2. Attribute - High Blood Pressure

High Blood Pressure	p	n	Info(p,n)
Yes	4	1	0.722
No	4	3	0.985

$$\text{Info(High Blood Pressure)} = (5/12)*0.722 + (7/12)*0.985 = 0.871$$

$$\text{Information-Gain(High Blood Pressure)} = \text{Info(Heart Disease)} - \text{Info(High Blood Pressure)} = 0.046$$

3. Attribute - Smoking

Smoking	p	n	Info(p,n)
Non-smoker	2	1	0.918
Occasional smoker	0	3	0.311
Former smoker	3	0	0.311
Frequent smoker	3	0	0.311

$$\text{Info(Smoking)} = (3/12)*0.918 + 3*(3/12)*0.311 = 0.462$$

$$\text{Information-Gain(Smoking)} = \text{Info(Heart Disease)} - \text{Info(Smoking)} = 0.046$$

4.

Attribute - Exercise

Exercise	p	n	Info(p,n)
Yes	4	4	1
No	4	0	0

$$\text{Info(Exercise)} = (8/12)*1 + (4/12)*0 = 0.67$$

$$\text{Information-Gain(Exercise)} = \text{Info(Heart Disease)} - \text{Info(Exercise)} = 0.248$$

Thus, the maximum Information-Gain is achieved when the split is done based on the attribute "Smoking", thus this would be the root of the decision tree.

B) Using the Gain-ratio to evaluate the split using the tables in the previous part

1. Attribute - Diabetes

$$\text{SplitInfo(Diabetes)} = -(5/12) \cdot \log(5/12) - (7/12) \cdot \log(7/12) = 0.98$$

$$\text{GainRatio(Diabetes)} = \text{InformationGain(Diabetes)} / \text{SplitInfo(Diabetes)} = 0.07/0.98 = 0.071$$

2. Attribute - High Blood Pressure

$$\text{SplitInfo(High Blood Pressure)} = -(5/12) \cdot \log(5/12) - (7/12) \cdot \log(7/12) = 0.98$$

$$\text{GainRatio(High Blood Pressure)} = \text{InformationGain(High Blood Pressure)} / \text{SplitInfo(High Blood Pressure)} = 0.046/0.98 = 0.0469$$

3. Attribute - Smoking

$$\text{SplitInfo(Smoking)} = -4 \cdot (3/12) \cdot \log(3/12) = 0.602$$

$$\text{GainRatio(Smoking)} = \text{InformationGain(Smoking)} / \text{SplitInfo(Smoking)} = 0.455/0.602 = 0.755$$

4. Attribute - Exercise

$$\text{SplitInfo(Exercise)} = -(8/12) \cdot \log(8/12) - (4/12) \cdot \log(4/12) = 0.918$$

$$\text{GainRatio(Exercise)} = \text{InformationGain(Exercise)} / \text{SplitInfo(Exercise)} = 0.248/0.918 = 0.27$$

Since the GainRatio for the attribute Smoking is the highest the 1st Split should be done based on it similar to part-A

C) Let C1 indicate a classification of Heart Disease as "Yes" and C2 as "No"

We are given the tuple $X(\text{Diabetes} = \text{"No"}, \text{High Blood Pressure} = \text{"No"}, \text{Smoking} = \text{"Non-smoker"}, \text{and Exercise} = \text{"Yes"})$ then we evaluate which of $P(X|C1)$ or $P(X|C2)$ is max and make the final classification.

First, computing the prior probability

$$P(\text{Heart Disease} = \text{"Yes"}) = 8/12 = 2/3$$

$$P(\text{Heart Disease} = \text{"No"}) = 4/12 = 1/3$$

Next, computing the conditional probabilities

$$P(\text{Diabetes} = \text{"No"} \mid \text{Heart Disease} = \text{"Yes"}) = 5/8$$

$$P(\text{Diabetes} = \text{"No"} \mid \text{Heart Disease} = \text{"No"}) = 2/4 = 1/2$$

$$P(\text{High Blood Pressure} = \text{"No"} \mid \text{Heart Disease} = \text{"Yes"}) = 4/8 = 1/2$$

$$P(\text{High Blood Pressure} = \text{"No"} \mid \text{Heart Disease} = \text{"No"}) = 3/4$$

$$P(\text{Smoking} = \text{"Non-smoker"} \mid \text{Heart Disease} = \text{"Yes"}) = 2/8 = 1/4$$

$$P(\text{Smoking} = \text{"Non-smoker"} \mid \text{Heart Disease} = \text{"No"}) = 1/4$$

$$P(\text{Exercise} = \text{"Yes"} \mid \text{Heart Disease} = \text{"Yes"}) = 4/8 = 1/2$$

$$P(\text{Exercise} = \text{"Yes"} \mid \text{Heart Disease} = \text{"No"}) = 4/4 = 1$$

Using these probabilities we obtain,

$$\begin{aligned} P(X \mid \text{Heart Disease} = \text{"Yes"}) &= P(\text{Diabetes} = \text{"No"} \mid \text{Heart Disease} = \text{"Yes"}) * P(\text{High Blood Pressure} = \text{"No"} \mid \text{Heart Disease} = \text{"Yes"}) * P(\text{Smoking} = \text{"Non-smoker"} \mid \text{Heart Disease} = \text{"Yes"}) \\ &* P(\text{Exercise} = \text{"Yes"} \mid \text{Heart Disease} = \text{"Yes"}) \end{aligned}$$

$$= 5/8 * 1/2 * 1/4 * 1/2 = 5/128$$

Similarly,

$$\begin{aligned} P(X \mid \text{Heart Disease} = \text{"No"}) &= P(\text{Diabetes} = \text{"No"} \mid \text{Heart Disease} = \text{"No"}) * P(\text{High Blood Pressure} = \text{"No"} \mid \text{Heart Disease} = \text{"No"}) * P(\text{Smoking} = \text{"Non-smoker"} \mid \text{Heart Disease} = \text{"No"}) * \\ &P(\text{Exercise} = \text{"Yes"} \mid \text{Heart Disease} = \text{"No"}) \end{aligned}$$

$$= 1/2 * 3/4 * 1/4 * 1 = 3/32$$

To find class C_i that maximizes $P(X | C_i) * P(C_i)$, we compute

$$P(X | \text{Heart Disease} = \text{"Yes"}) * P(\text{Heart Disease} = \text{"Yes"}) = 5/128 * 2/3 = 0.026$$

$$P(X | \text{Heart Disease} = \text{"No"}) * P(\text{Heart Disease} = \text{"No"}) = 3/32 * 1/3 = 0.031$$

Since the value of $P(X | \text{Heart Disease} = \text{"No"}) * P(\text{Heart Disease} = \text{"No"})$ is more the Bayesian Classifier would classify X as Heart Disease = "No"

HW4

Q1

$$\text{Mean of error rates in } M1 \text{ } errM1 = (30.4+32.1+20.7+22.6+31.5+41.0+27.5+25.4+21.5+26.1) / 10 = 27.88$$

$$\text{Mean of error rates in } M2 \text{ } errM2 = (22.7+14.2+22.9+20.3+21.7+22.4+20.1+19.1+16.2+32.0) / 10 = 21.16$$

Since a total of 10 rounds are done $k = 10$ and the degrees of freedom = $10 - 1 = 9$

Next calculating the variance,

$$var(M_1 - M_2) = \frac{1}{k} \sum_{i=1}^k [err(M_1)_i - err(M_2)_i - (\overline{err}(M_1) - \overline{err}(M_2))]^2.$$

the difference in means is $(errM1 - errM2) = 27.88 - 21.16 = 6.72$

$$var(M1 - M2) = 1/10 * [(30.4-22.7-6.72)^2 + (32.1-14.2-6.72)^2 + (20.7-22.9-6.72)^2 + (22.6-20.3-6.72)^2 + (31.5-21.7-6.72)^2 + (41-22.4-6.72)^2 + (27.5 - 20.1-6.72)^2 + (25.4-19.1-6.72)^2 + (21.5-16.2-6.72)^2 + (26.1-32-6.72)^2]$$

$$= 1/10 * [537.596]$$

= 53.76

Next calculating the t-statistic

$$t = \frac{\overline{err}(M_1) - \overline{err}(M_2)}{\sqrt{\text{var}(M_1 - M_2)/k}},$$

t = (errM1 - errM2) /

t = 6.72 / (

t = 2.89

sig = 1% = 0.01 and degrees of freedom = 10 - 1 = 9






















2-sided test $1 - \text{sig}/2 = 1 - 0.01/2 = 0.995$

Check T-table for (9, 0.995) = 3.250

Thus $t=2.89 < 3.250$ and hence the null hypothesis cannot be rejected. Thus, there is no significant difference between the two models and any difference between the error rates of M1 and M2 can be attributed to chance.

1st Iteration of the K-means algorithm

Point	Distance from Centroid A(1, 2)	Distance from Centroid B (3,3)	Distance from Centroid C (5,5)	Cluster assignment
p_1 (1,2)				A

p_2 (2,1)				A
p_3 (3,2)				B
p_4 (3,3)				B
p_5 (4,1)				B
p_6 (5,2)				B
p_7 (5,5)				C
p_8 (6,4)				C

The new centroids are:

Centroid A = $((1+2)/2, (2+1)/2) = (1.5, 1.5)$ taking mean of p_1 and p_2

Centroid B = $((3+3+4+5)/4, (2+3+1+2)/4) = (3.75, 2)$ taking mean of p_3, p_4, p_5 and p_6

Centroid C = $((5+6)/2, (5+4)/2) = (5.5, 4.5)$

Q4

One of the tools I have used is the Pandas library in Python. I have used Pandas along with Numpy to analyze a historical broadband metrics dataset that is tracked by a US government body called FCC, this was part of a research project I'm working on. The pandas library in Python is a powerful tool for data mining and analysis, particularly for working with structured data. It provides easy-to-use data structures and data manipulation functions that are highly efficient and optimized for performance. I have mainly used pandas to load the data from CSV format into dataframes and then perform some aggregations and transformations to it.

Some of the key strengths of the pandas library are:

1. **Data Manipulation:** Pandas provide a wide range of functions and methods for data manipulation, including filtering, sorting, grouping, merging, reshaping, and aggregating data. It allows for efficient data wrangling and transformation, making it easier to clean and preprocess datasets for data mining tasks.
2. **Data Structures:** The pandas library offers two primary data structures, namely Series and DataFrame. Series is a one-dimensional labeled array, and DataFrame is a two-dimensional labeled data structure, similar to a table or a spreadsheet. These data structures provide a flexible and intuitive way to store, analyze, and manipulate data, including handling missing values and performing arithmetic operations.
3. **Data Integration:** pandas integrates well with other data analysis and data mining libraries in Python, such as NumPy, scikit-learn, and Matplotlib. This integration allows for seamless data interchange between different tools and enhances the functionality and capabilities of pandas.

Possible Limitations of the pandas library:

1. **Memory Usage:** pandas DataFrames store data in memory, which can be a limitation when dealing with extremely large datasets that do not fit into memory. In such cases, alternative tools like Apache Spark or Dask, which support distributed computing or out-of-memory processing, may be more suitable.
2. **Performance for Certain Operations:** While pandas is generally efficient for most data manipulation tasks, some operations, such as iterating over large DataFrames, can be slower compared to optimized alternatives like vectorized operations using NumPy. In such cases, it is recommended to use pandas' built-in functions or explore other approaches to improve performance.

Q2