PLAGIARISM DECLARATION

1. I confirm that this assignment is my own work and is not copied from another person's work or from any form of sources.
2. I acknowledge that copying someone else's assignment, or part of it, constitutes a form of plagiarism.
3. I have not allowed anyone to copy my work or part of it, with the intention of passing it off as their own work.

Name: Eden Will Sng Jin Xuan

Admin: 201520M

Signature: Eden

Date: 19-Jan-2024

School Of Information Technology

Foundation of AI Assignment

Admin No & Name:	201520M: Eden Will Sng Jin Xuan
PEM Group:	SF2102
Module:	IT310C
Tutor:	Lim Sing Tat

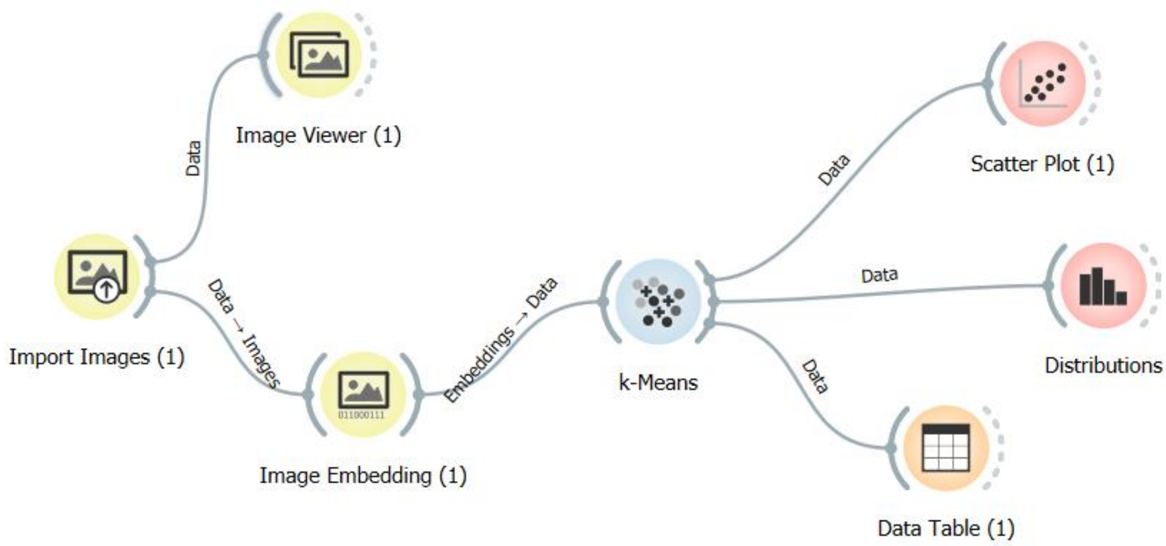
Table of Contents

Question 1 (6m)

Question Part A (1m)

Apply the best algorithm in Python's Orange library to group together images that share similar features.

The best algorithm is:



K-Means Clustering, it is good at finding clusters of 2d & 3d images.

Other Explanation: (Extras)

Hierarchical clustering, Louvain Method and DB scan did not offer good enough accuracy as compared to K Means.

While Louvain Method offers the second based on the clusters. its best suited for graphical data which is not what the dataset is given. Louvain method is also a graph-based clustering method or community detection method. So, for this practical we will use K-Means Clustering

Sources:

1. cRNA Python Workshop Clustering (n.d.) retrieved 2024 Jan 7. From the Chan Zuckerberg GitHub website: https://chanzuckerberg.github.io/scRNA-python-workshop/analysis/04-clustering.html

2. Louvain Method (n.d.) Retrieved 2024, Jan 7. From the Wikipedia Website: https://en.wikipedia.org/wiki/Louvain_method

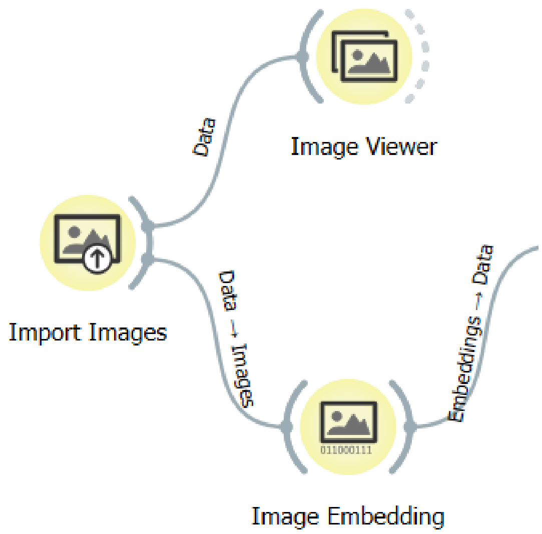
Question 1 Part B (3M)

Describe how the clustering algorithm used in Step a) can group images with similar. features into the same cluster.

Explanation:

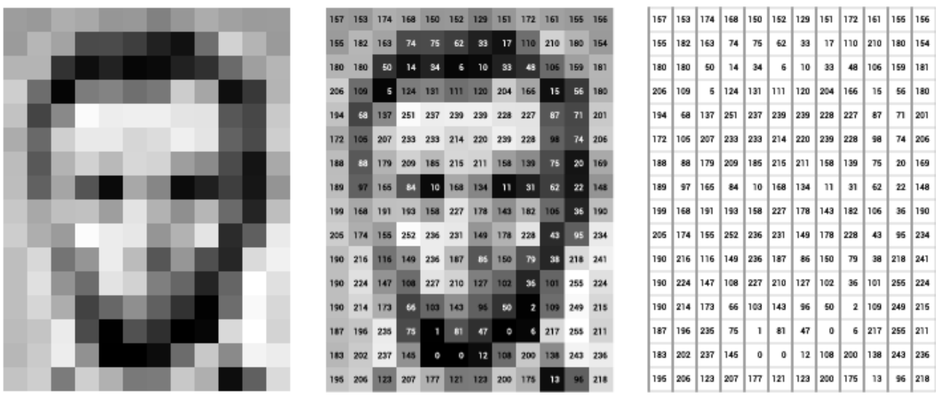
First Images by itself can’t be understood by the machine.

So, the image must be converted into embedded data first for the machine to understand it. This embedded image data will also convert the features of the image into datapoints.



Example of what it will look like in Python orange.

This will be needed for the K-Means algorithm to work.



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Credits: Stanford Asst. Prof, Serena Yeung (n.d.). https://ai.stanford.edu/~syyeung/ Retrieved 2024, Jan 7. Image taken from This textbook:
Barak, M. (2020). Teaching Problem-Solving in the Digital Era. In: Williams, P.J., Barlex, D. (eds) Pedagogy for Technology Education in Secondary Schools. Contemporary Issues in Technology Education. Springer, Cham. https://doi.org/10.1007/978-3-030-41548-8_13

Here’s a visual representation of how the images is converted.

K means is a partition clustering method that group images in our dataset based on common and similar features.

How K-Means work is simple, for example given 2 groups. The K-Means will be assumed as 2.

Next, 2 random data points will be chosen on the vector space. The points could be called K1 Point or K2 Point.

The embedded images will then be measured by how close it to is either K1 point or K2 Point through one of the many distance measurements.

In this case Euclidean distance could be used.

The closest embedded image to one of the cluster data points will be grouped to them.

So, at the end of our first K-means algorithm run, we should have our images grouped into 2 groups.

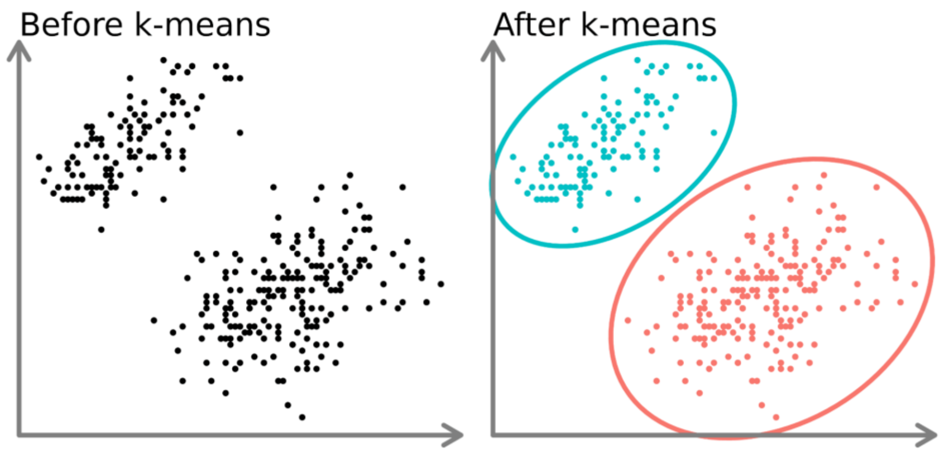


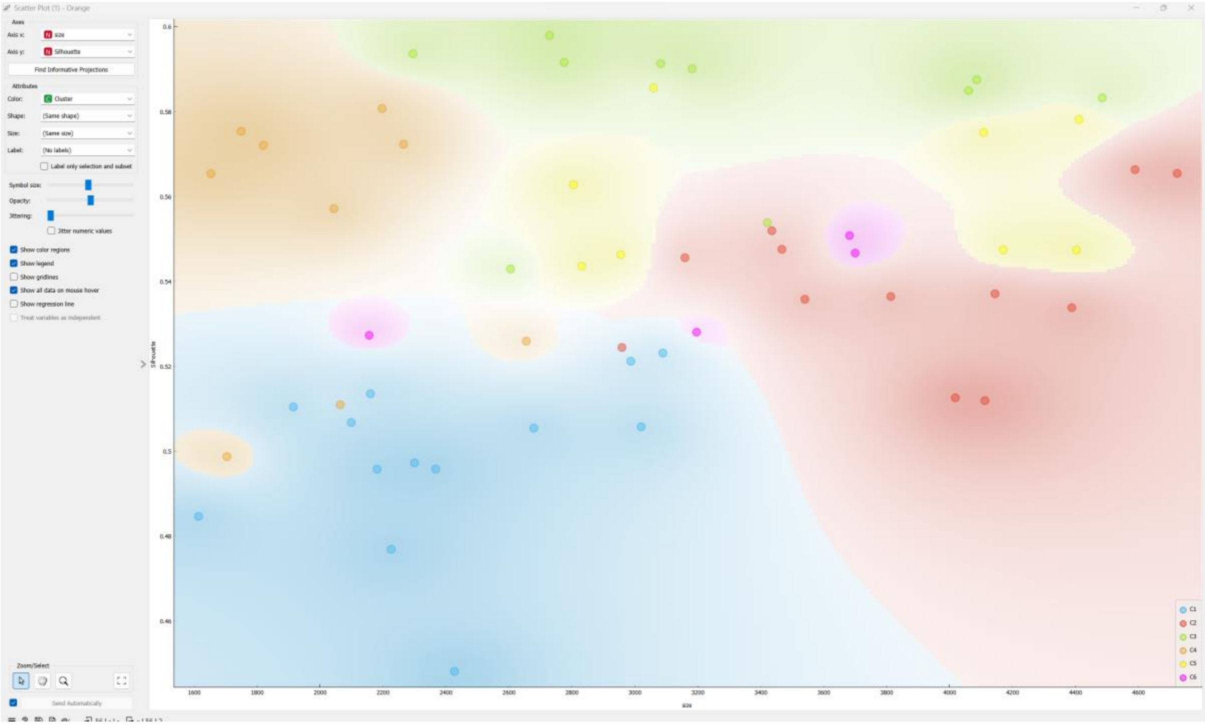
Image source: Babitz,K (2023, Mar 23). Introduction to k-Means Clustering with scikit-learn in Python. Retrieved 2024, Jan 7 From the Data Camp website:
https://www.datacamp.com/tutorial/k-means-clustering-python

Here's a visual representation of my description.

However, this is not accurate as the data points chosen for K-means is arbitrary. So, based on the current datapoints grouped to K1 or K2.

We will do a summation & calculation using K-means algorithm to determine the next K1 and K2 coordinates. Once we have the new K1 & K2 Coordinates, we will group the datapoints again into different clusters.

We will iterate through this process multiple times until the K-means data points do not change significantly or the operator has defined a fixed number of iterations.



X-Axis: Size of Image

Y-Axis: Silhouette generated from K-Means Clustering

Colour: Cluster

At the End of multiple iterations, the image dataset will then be grouped. For our practical assignment we have 6 groupings. Here is a visual representation of the cluster groups where X axis is the size of the image and Y axis is the silhouette of the image. The Silhouette variable is derived from K means clustering to show how similar the images are to each cluster.

The results are as follows.

BMW:

8/8 in Cluster 3

Chrome:

4/10 in Cluster 6

4/10 in Cluster 1

2/10 in Cluster 2

Apple:

9/10 in Cluster 4

1/10 in Cluster 2

Google:

2/10 in Cluster 3

4/10 in Cluster 2

4/10 in Cluster 1

HP:

8/10 in Cluster 5

1/10 in Cluster 2

1/10 in Cluster 1

Coca-Cola

4/8 in Cluster 2

4/8 in Cluster 1

In General, the images which were consistently grouped correctly were Apple, HP & BMW because the images are generally distinctive.

While the groups Coca-Cola, Chrome & Google are extremely similar. I hypothesize it's due to the image size and the fact that google chrome and Coca-Cola have similar colors.

Therefore, K-Means clustering has difficulty differentiating these 3 brands into separate groups. hence, they are often grouped together.

This holds true for hierarchical clustering. Though K Means is more accurate.

Appendix A

BMW_1	BMW_1.jpg	4485	70	70	C3	0.583252
BMW_2	BMW_2.jpg	4060	70	70	C3	0.584971
BMW_3	BMW_3.jpg	4086	70	70	C3	0.587515
BMW_4	BMW_4.jpg	3182	70	70	C3	0.590124
BMW_5	BMW_5.jpg	3082	56	56	C3	0.591309
BMW_6	BMW_6.jpg	2729	56	56	C3	0.597977
BMW_7	BMW_7.jpg	2775	56	56	C3	0.591608
BMW_8	BMW_8.jpg	2295	56	56	C3	0.59366
Chrome_1	Chrome_1.jpg	3196	70	70	C6	0.52806
Chrome_10	Chrome_10.jpg	2181	56	56	C1	0.495755
Chrome_2	Chrome_2.jpg	3700	70	70	C6	0.546684
Chrome_3	Chrome_3.jpg	3159	70	70	C2	0.545587
Chrome_4	Chrome_4.jpg	3813	70	70	C2	0.536437
Chrome_5	Chrome_5.jpg	3682	70	70	C6	0.550811
Chrome_6	Chrome_6.jpg	2226	56	56	C1	0.47689
Chrome_7	Chrome_7.jpg	2156	56	56	C6	0.527299
Chrome_8	Chrome_8.jpg	1614	56	56	C1	0.484626
Chrome_9	Chrome_9.jpg	2300	56	56	C1	0.497262
CocaCola_1	CocaCola_1.jpg	4018	70	70	C2	0.512575
CocaCola_2	CocaCola_2.jpg	4589	70	70	C2	0.566338
CocaCola_3	CocaCola_3.jpg	4723	70	70	C2	0.565444
CocaCola_4	CocaCola_4.jpg	4388	70	70	C2	0.533818
CocaCola_5	CocaCola_5.jpg	2367	56	56	C1	0.495791
CocaCola_6	CocaCola_6.jpg	2987	56	56	C1	0.521186
CocaCola_7	CocaCola_7.jpg	3089	56	56	C1	0.523108
CocaCola_8	CocaCola_8.jpg	3020	56	56	C1	0.505751
apple_1	apple_1.jpg	2265	70	70	C4	0.572313
apple_10	apple_10.jpg	1704	56	56	C4	0.498715
apple_2	apple_2.jpg	2655	70	70	C4	0.525931
apple_3	apple_3.jpg	2044	70	70	C4	0.557139
apple_4	apple_4.jpg	2197	70	70	C4	0.580744
apple_5	apple_5.jpg	2959	70	70	C2	0.524439
apple_6	apple_6.jpg	1820	56	56	C4	0.572061
apple_7	apple_7.jpg	2064	56	56	C4	0.510938
apple_8	apple_8.jpg	1653	56	56	C4	0.565382
apple_9	apple_9.jpg	1749	56	56	C4	0.575385
google_1	google_1.jpg	3421	70	70	C3	0.553838
google_10	google_10.jpg	2099	56	56	C1	0.50675
google_2	google_2.jpg	4144	70	70	C2	0.537087
google_3	google_3.jpg	3435	70	70	C2	0.551931
google_4	google_4.jpg	3540	70	70	C2	0.535793
google_5	google_5.jpg	3467	70	70	C2	0.547555
google_6	google_6.jpg	2605	56	56	C3	0.542939
google_7	google_7.jpg	2679	56	56	C1	0.505434
google_8	google_8.jpg	1915	56	56	C1	0.510427
google_9	google_9.jpg	2160	56	56	C1	0.513525

47	hp_1	hp_1.jpg	4170	70	70	C5	0.547437
48	hp_10	hp_10.jpg	4411	70	70	C5	0.578156
49	hp_2	hp_2.jpg	4403	70	70	C5	0.547393
50	hp_3	hp_3.jpg	4108	70	70	C5	0.575087
51	hp_4	hp_4.jpg	4112	70	70	C2	0.511917
52	hp_5	hp_5.jpg	2831	56	56	C5	0.543595
53	hp_6	hp_6.jpg	2955	56	56	C5	0.546294
54	hp_7	hp_7.jpg	2805	56	56	C5	0.562785
55	hp_8	hp_8.jpg	2427	56	56	C1	0.44811
56	hp_9	hp_9.jpg	3059	56	56	C5	0.585597

Question 1 Part C (2m)

Describe 2 applications that you can use as the above clustering result.

1 - Malware Detection

Source: Bodnar, C. (2013, October 29). A Malware Classification. Retrieved 2024, Jan 7. From the Kaspersky Website: https://www.kaspersky.com/blog/a-malware-classification/3037/

Since most Anti Malware Scanners use virus signatures to detect malware. It may not work if the malware mutates or changes itself to evade detection.

However, malware does perform similarly and have similar characteristics. Therefore, we can cluster the files based on the features of what a malware would behave or feature into datapoints. These datapoints could then be clustered into groups of normal software and malware software.

Clustering is good in this case as malware is changing faster than we could add virus signature to the Anti malware software. Unsupervised learning helps to defeat new and unknown threats in the environment because it could spot patterns, we humans can’t see or have the technical skills to do so.

Supporting Sources:

1. Saha,A. (2021, Aug 21). K-MEANS CLUSTER AND IT'S USE CASE IN CYBER SECURITY…. Retrieved 2024, Jan 8. From the Medium Website: https://arnabsaha1.medium.com/k-means-cluster-and-its-use-case-in-cyber-security-3abfaab2ec09
2. Mosharrat, N., Sarker, I.H., Anwar, M.M., Islam, M.N., Watters, P., Hammoudeh, M. (2022). Automatic Malware Categorization Based on K-Means Clustering Technique. In: Arefin, M.S., Kaiser, M.S., Bandyopadhyay, A., Ahad, M.A.R., Ray, K. (eds) Proceedings of the International Conference on Big Data, IoT, and Machine Learning. Lecture Notes on Data Engineering and Communications Technologies, vol 95. Springer, Singapore. https://doi.org/10.1007/978-981-16-6636-0_49

2 - Email Spam filtering

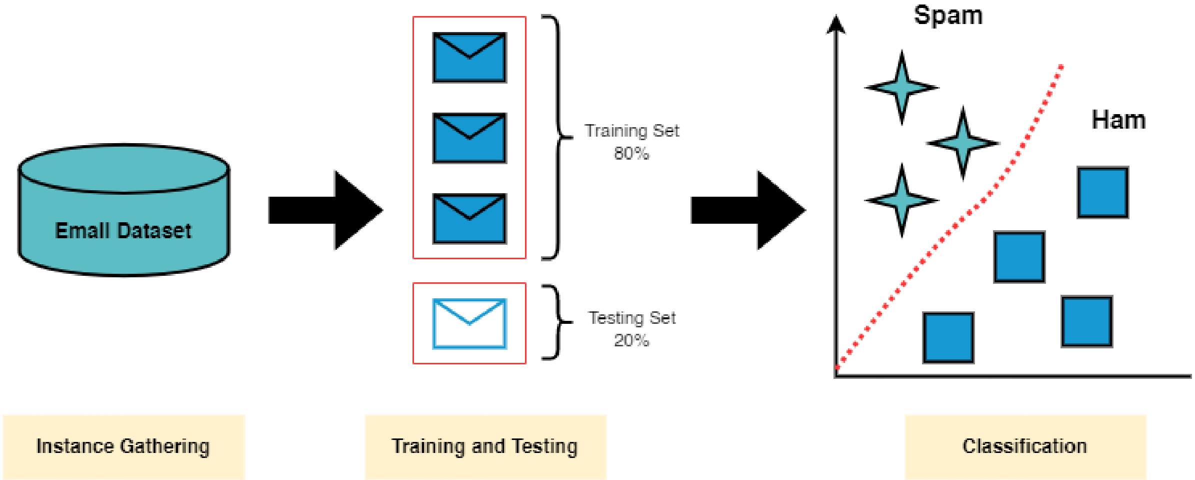


Image Source: Tabish, A. (2022, Aug 23). Machine Learning Techniques for Spam Detection in Email. Retrieved 2024, Jan 8. From the Medium Website:
https://medium.com/@alinatabish/machine-learning-techniques-for-spam-detection-in-email-7db87eb11bc2

Phishing emails are becoming more sophisticated and dangerous. Even a well-educated user in cybersecurity can still fall prey to phishing emails. Using supervised learning for phishing emails may not be effective due to the rapidly changing landscape of phishing techniques. Some phishing techniques are specialized and customized to completely evade the traditional phishing email detectors.

However, since all phishing emails have similar characteristics that separates it from normal emails.

Example Phishing Email features includes.
1. high probability of the word money
2. suspicious email links
3. embedded viruses inside attachments

We can group these into datapoints that could be clustered to either spam or ham emails.

Supporting Articles Sources:

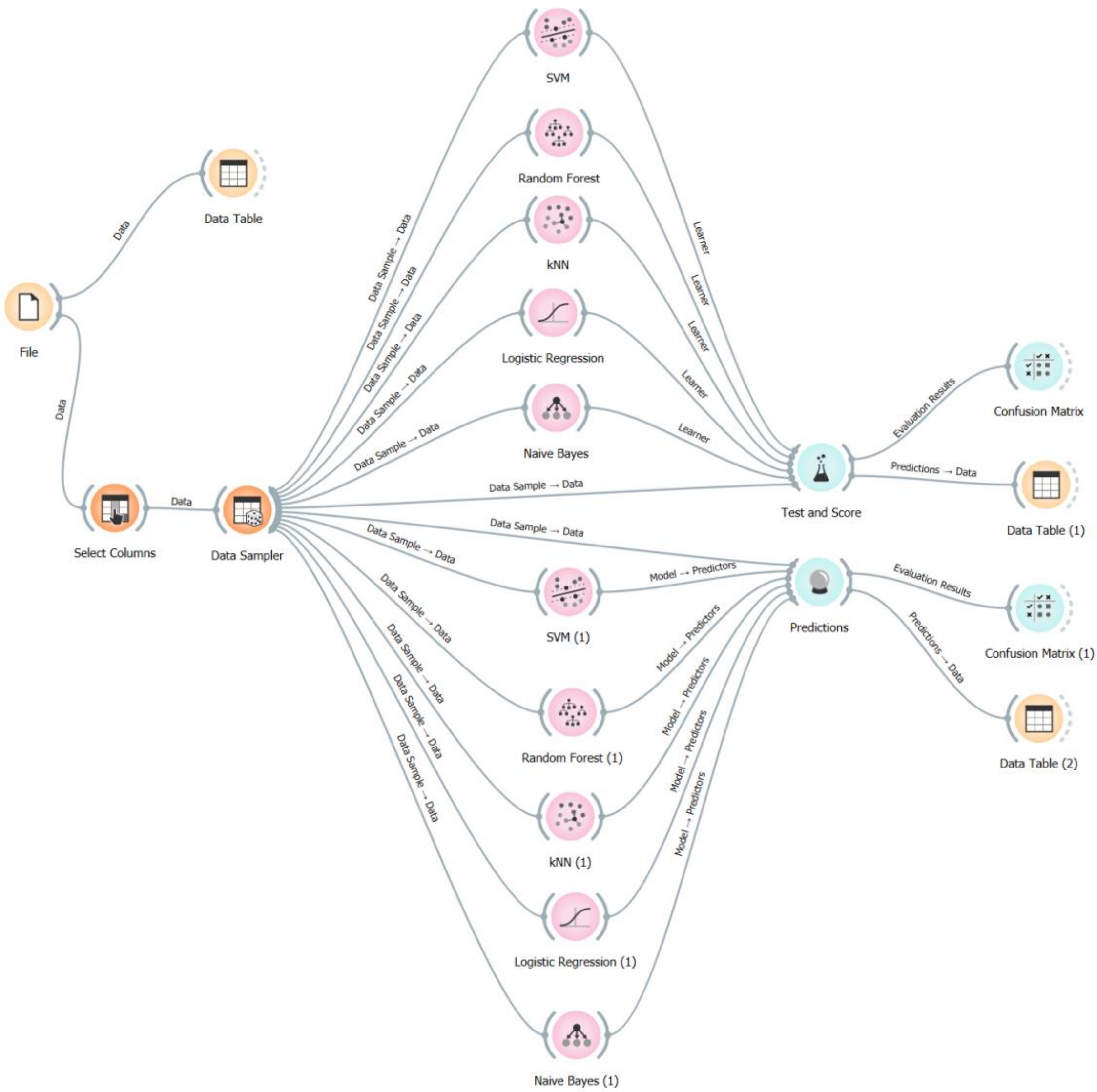
1. Surbhi (2021, July 20). K-means Clustering and its use-cases in Security Domains. Retrieved 2024, Jan 9 from the LinkedIn website: https://www.linkedin.com/pulse/k-means-clustering-its-use-cases-security-domains-surbhi-/

2. Tabish, A. (2022, Aug 23). Machine Learning Techniques for Spam Detection in Email. Retrieved 2024, Jan 8. From the Medium Website: https://medium.com/@alinatabish/machine-learning-techniques-for-spam-detection-in-email-7db87eb11bc2

Question 2 Classification of Notes Authenticity (5marks)

Question 2 Part A)

Develop a Python Orange program to train machine learning classifiers to classify the mobile phone price range. The classifier must achieve an F1 score of at least 0.9 on the test dataset. (2 marks)



Here is the python orange program I have created to classify the Mobile Price Ranges.

I used the following Algorithms for testing:

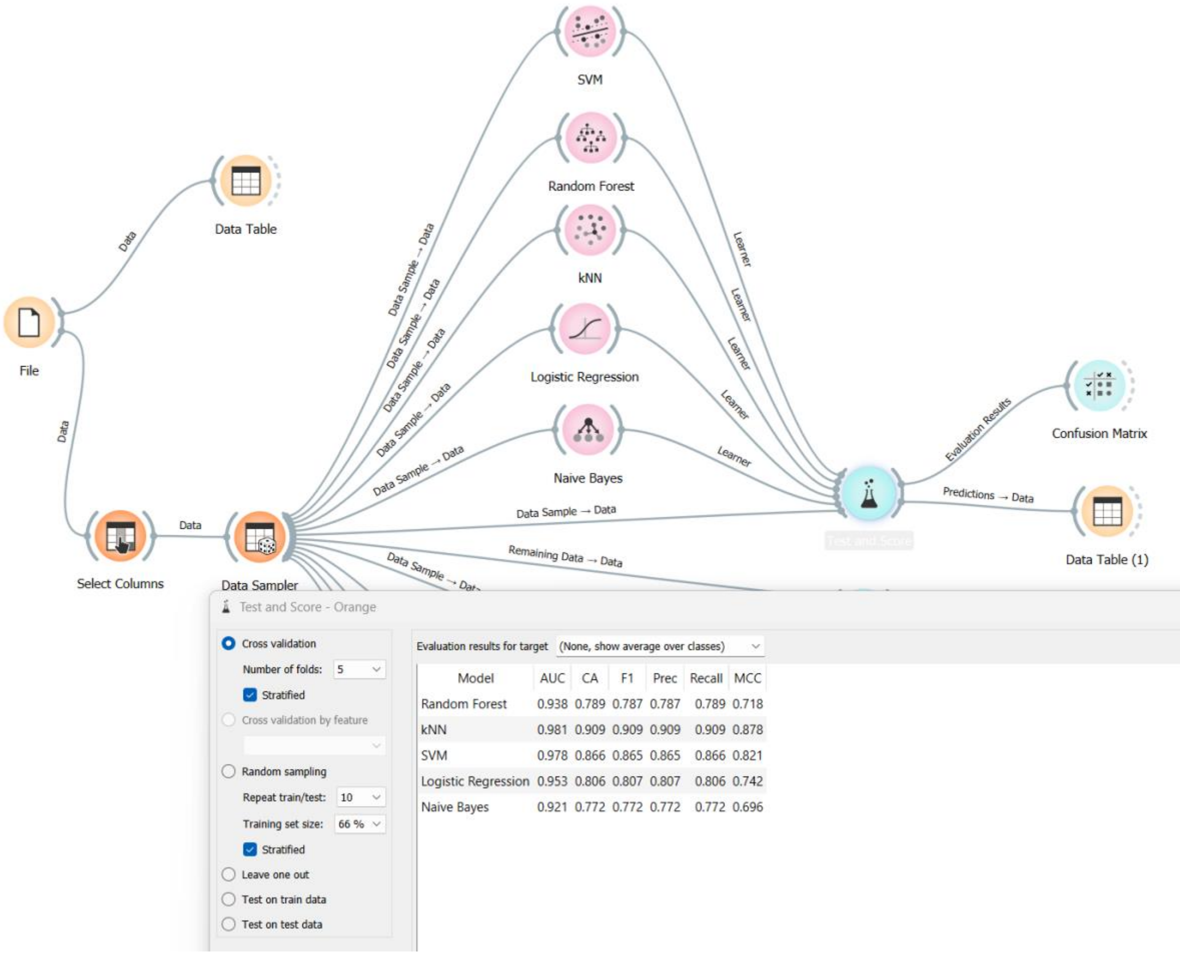
- SVM
- Random Forest
- kNN
- Logistic Regression
- Naïve Bayes

In the first phase, I want to ensure that the algorithms used are suitable for classifying the phone range.

I have split my data into two sets: 70% Training and 30% testing Data.

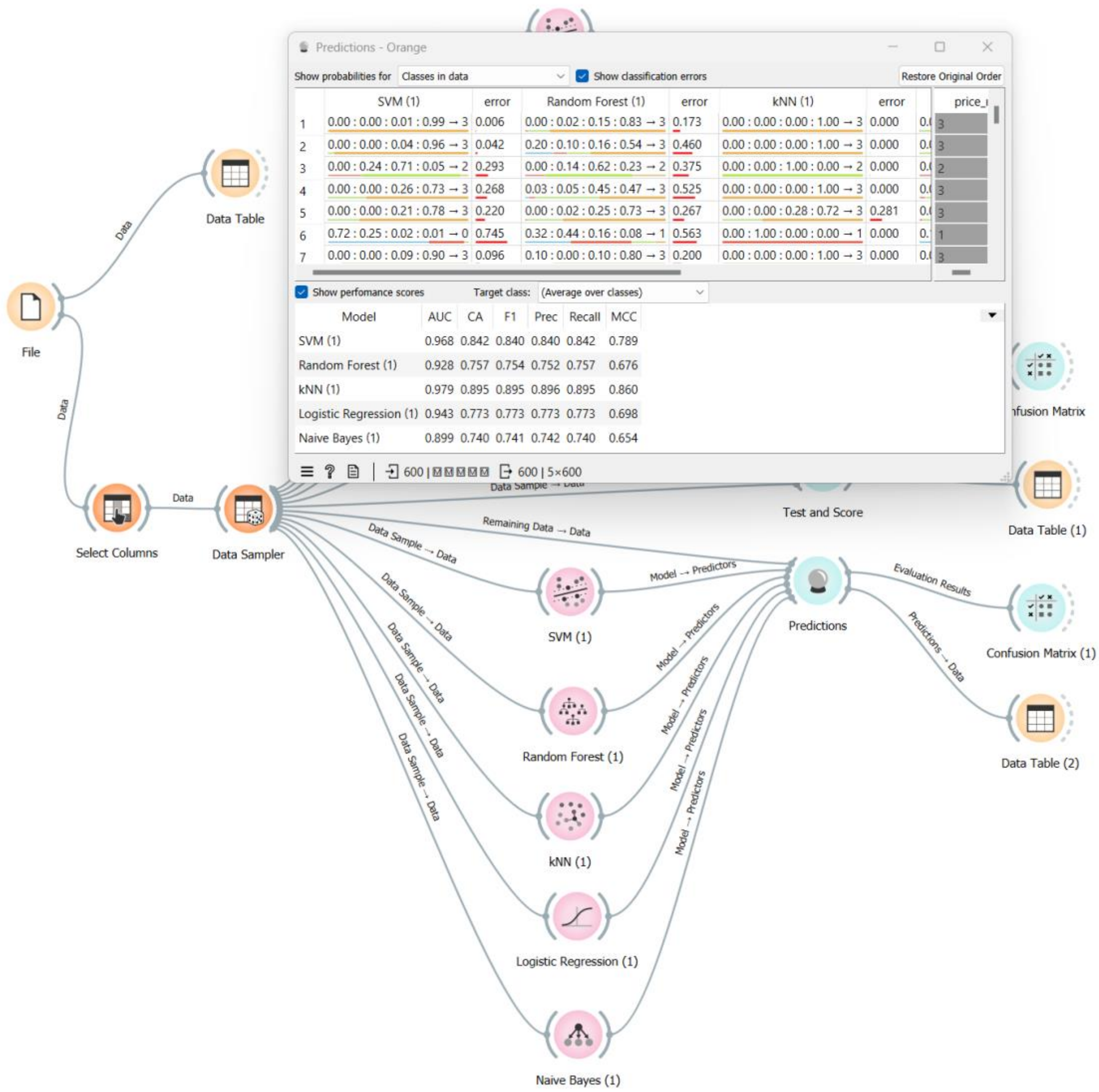
To do so, we will use the test and score module and cross validation checked to validate that the Machine Learning Model is suitable for this Task.

The Classifier that scored with F1 Score beyond 0.9 is K-Nearest Neighbors (KNN).

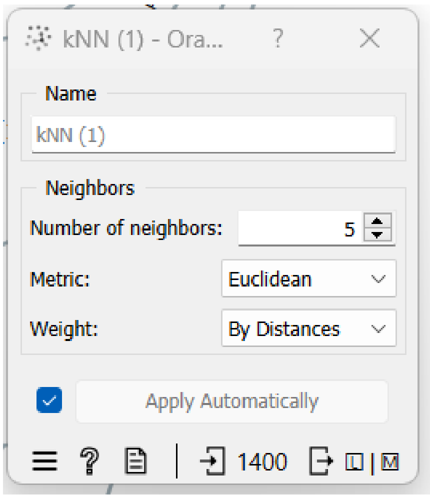


Model	AUC	CA	F1	Prec	Recall	MCC
Random Forest	0.938	0.789	0.787	0.787	0.789	0.718
kNN	0.981	0.909	0.909	0.909	0.909	0.878
SVM	0.978	0.866	0.865	0.865	0.866	0.821
Logistic Regression	0.953	0.806	0.807	0.807	0.806	0.742
Naive Bayes	0.921	0.772	0.772	0.772	0.772	0.696

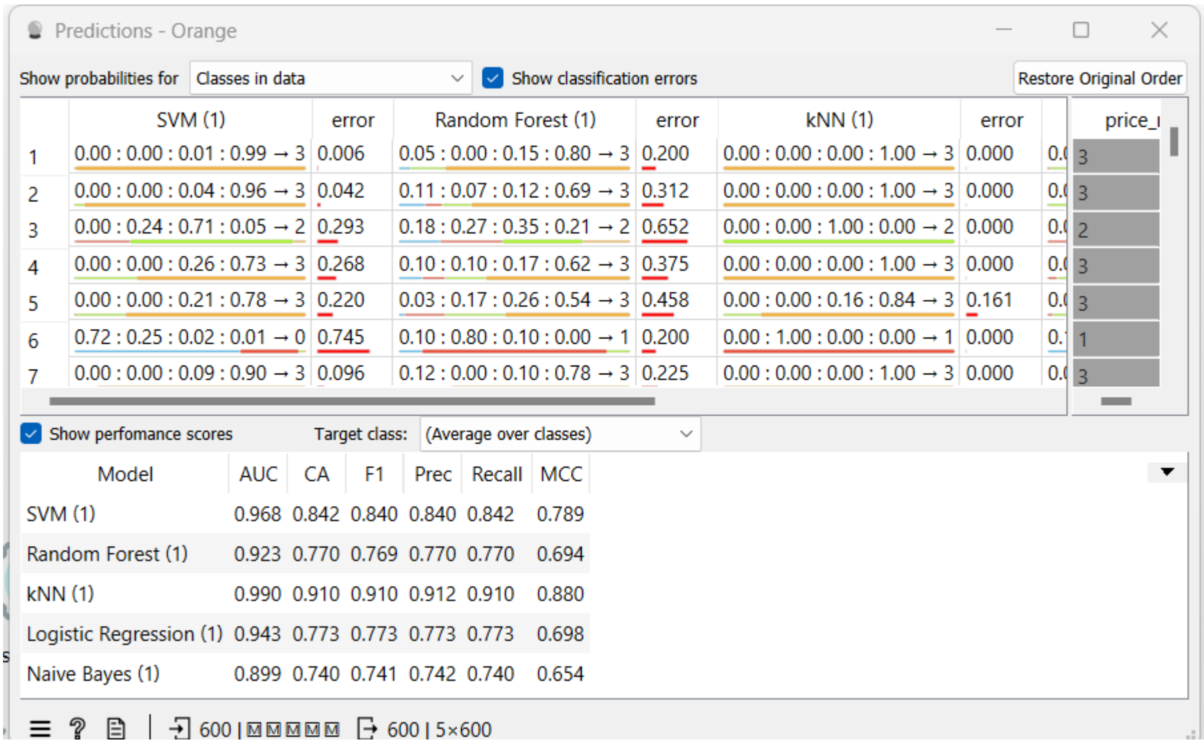
Now we will evaluate the algorithms through the prediction, the prediction mode uses the actual model to test whether it can classify the mobile phone price ranges.
We will be using the remaining data for the Test Data set.



The scores here are unsatisfactory, with kNN F1-Score being 0.895%.
This suggests we will need to do some hyperparameter tuning to increase the F1 Score to our desired target of 0.9.

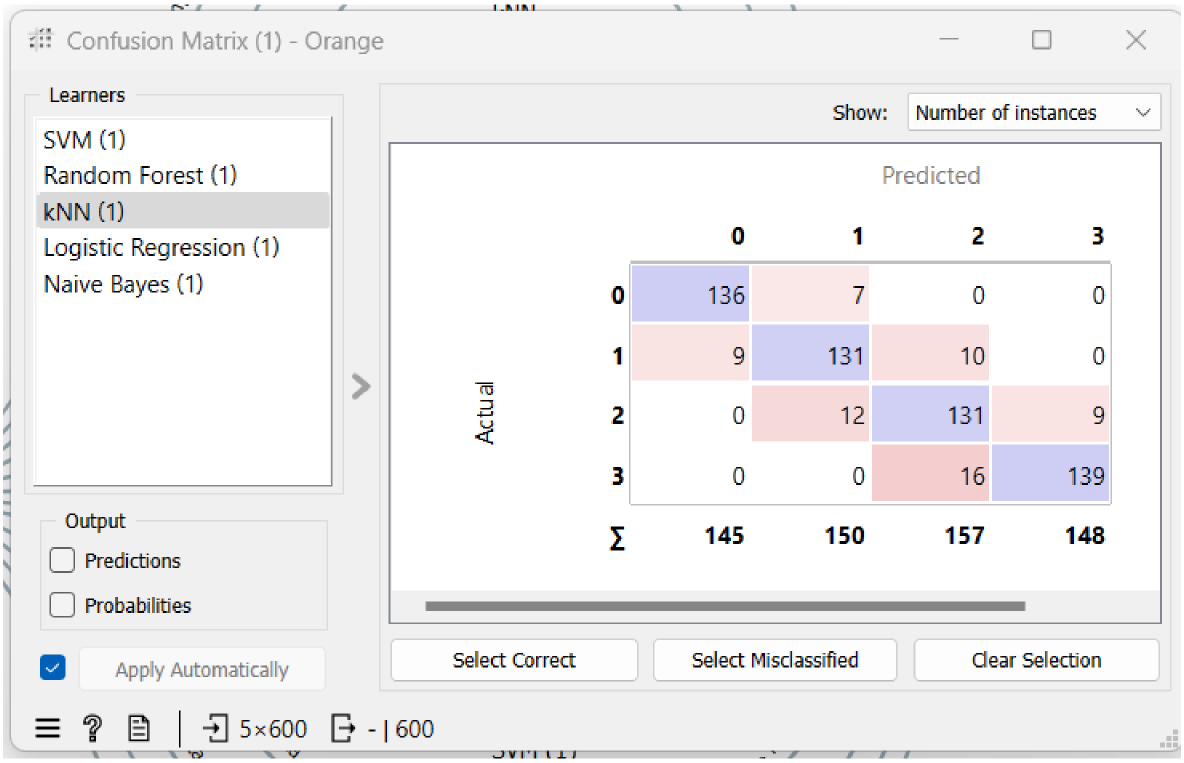


I have changed the Number of Neighbors to 5 and the Metric to Euclidean Distance. Let's look at our updated scores.

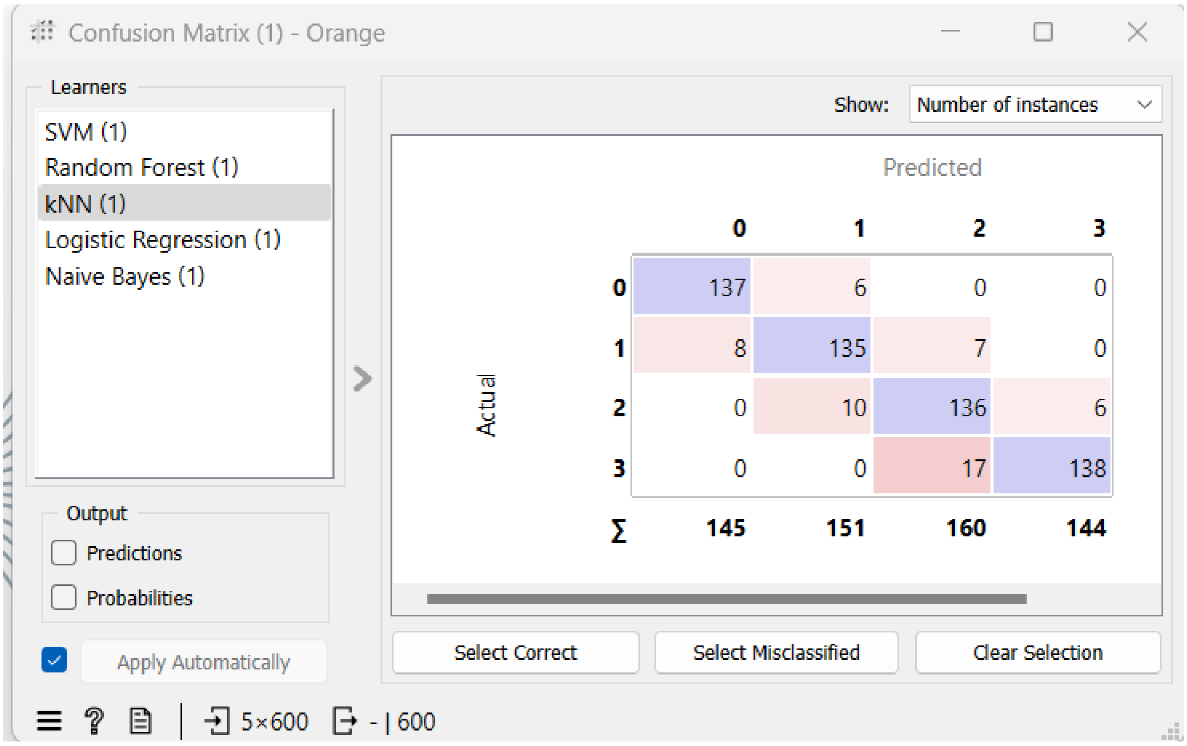


As we can see here the F1 Score of kNN is now 0.91 which is above our desired score.
Let's look at the Confusion Matrix to see how these fares.

Before parameter tuning:



After Parameter Tuning:



As we can see here, the accuracy of the F1 score directly corresponds to a more accurate prediction and actual score of the dataset!

Question 2 Part B)

Explain the purpose of the training, validation, and test data in the machine learning workflow.

(1 mark)

Here's a good visualisation of the purpose of training, validation and test data in machine learning workflow.

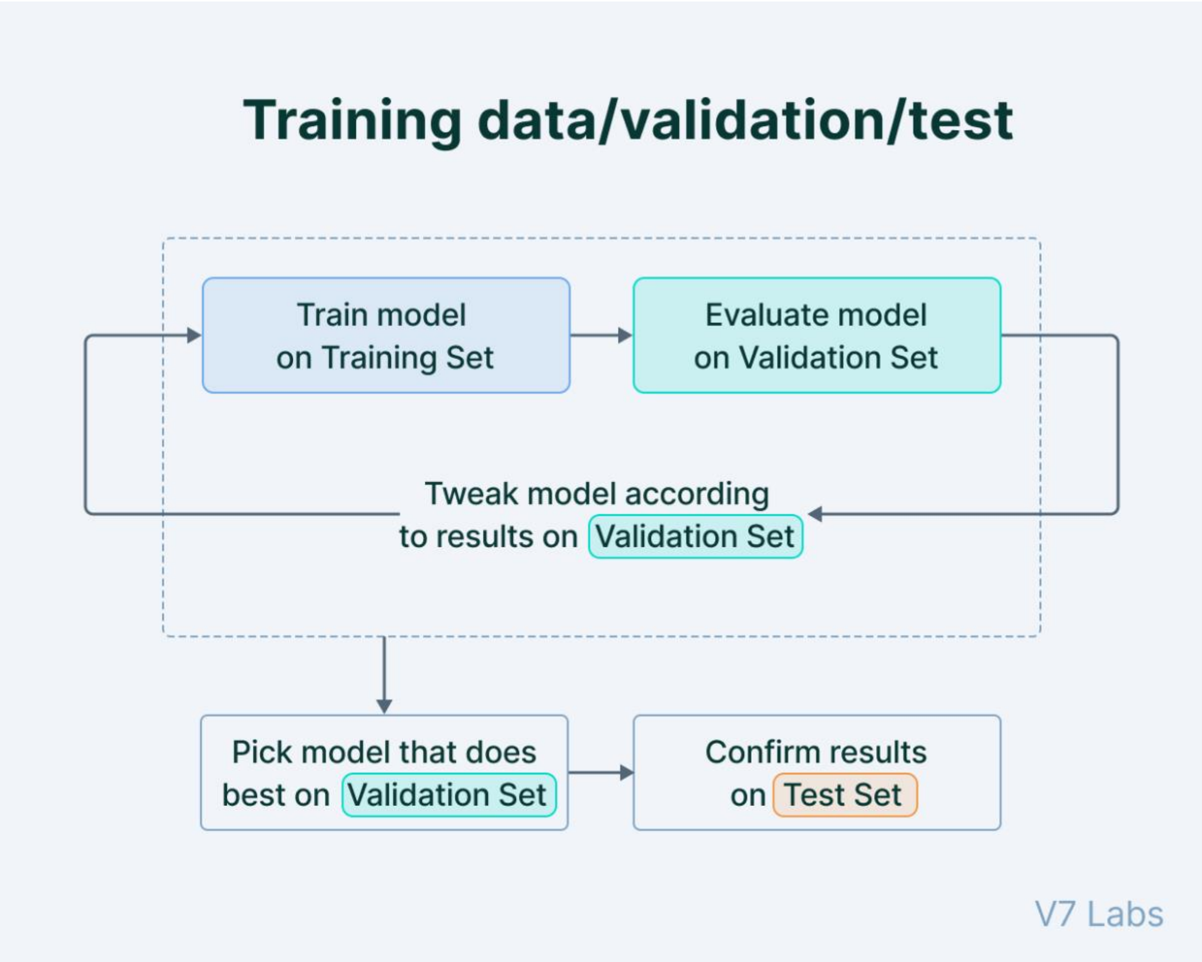


Image source: Baheti, P. (2021, September 21). Train Test Validation Split: How To & Best Practices [2023]. Retrieved 2024, Jan 7. From the V7 Labs website: https://www.v7labs.com/blog/train-validation-test-set

Training Data:

Training dataset is used to train the model on the problems we are looking to solve.

Validation Dataset:

This dataset is used to evaluate the training model and to introduce hyper parameter changes if the results does not meet the desired outcome. The validation dataset gets more biased when used more during the training process.

Testing Dataset:

Testing dataset is the final step in the machine learning workflow. The training dataset help gives us an unbiased reference verifies that the model can solve unknown problems that aren't introduced to it during the training dataset.

Question 2 Part C)

Suppose you are only allowed to use one feature to train the classifier. Which feature would you choose, and explain why this feature was selected? (2 marks)

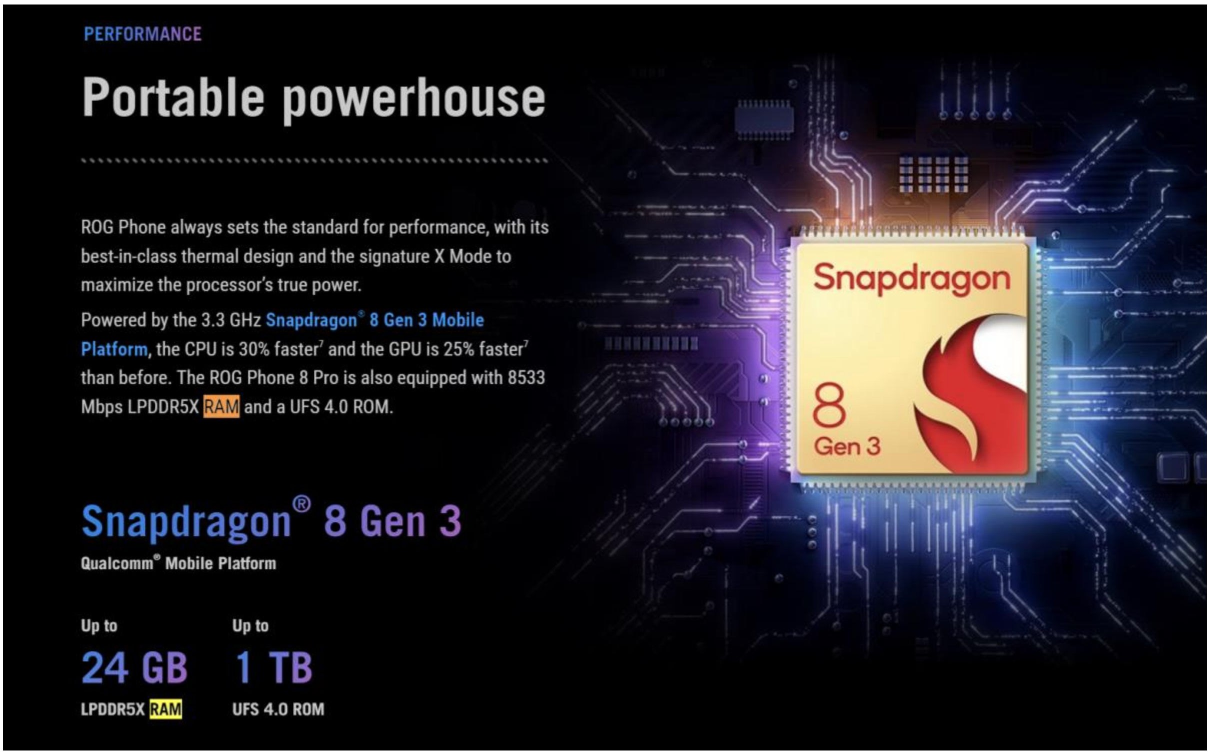


Image source: Asus (n.d.) Rog Phone 8. Retrieved 2024, Jan 10. From the Asus Website: https://rog.asus.com/phones/rog-phone-8-pro/

Hypothesis:

Only Feature: RAM

Usually ram cost is linearly matched with the phone price range. In most mobile phones, the capacity of ram is linearly increase with the price range of phones. Higher capacity ram is always tied to the high-end specs of phone processors.

Verification:

Let's verify that this is the case,

Evaluation results for target (None, show average over classes)

Model	AUC	CA	F1	Prec	Recall	MCC
Random Forest	0.885	0.692	0.692	0.693	0.692	0.590
kNN	0.851	0.690	0.689	0.689	0.690	0.587
SVM	0.824	0.657	0.643	0.654	0.657	0.550
Logistic Regression	0.929	0.764	0.764	0.764	0.764	0.686
Naive Bayes	0.889	0.763	0.763	0.763	0.763	0.684

It seems like the F1 Scores in general is around 0.6-0.7 which suggest high correlation between ram and phone price range.

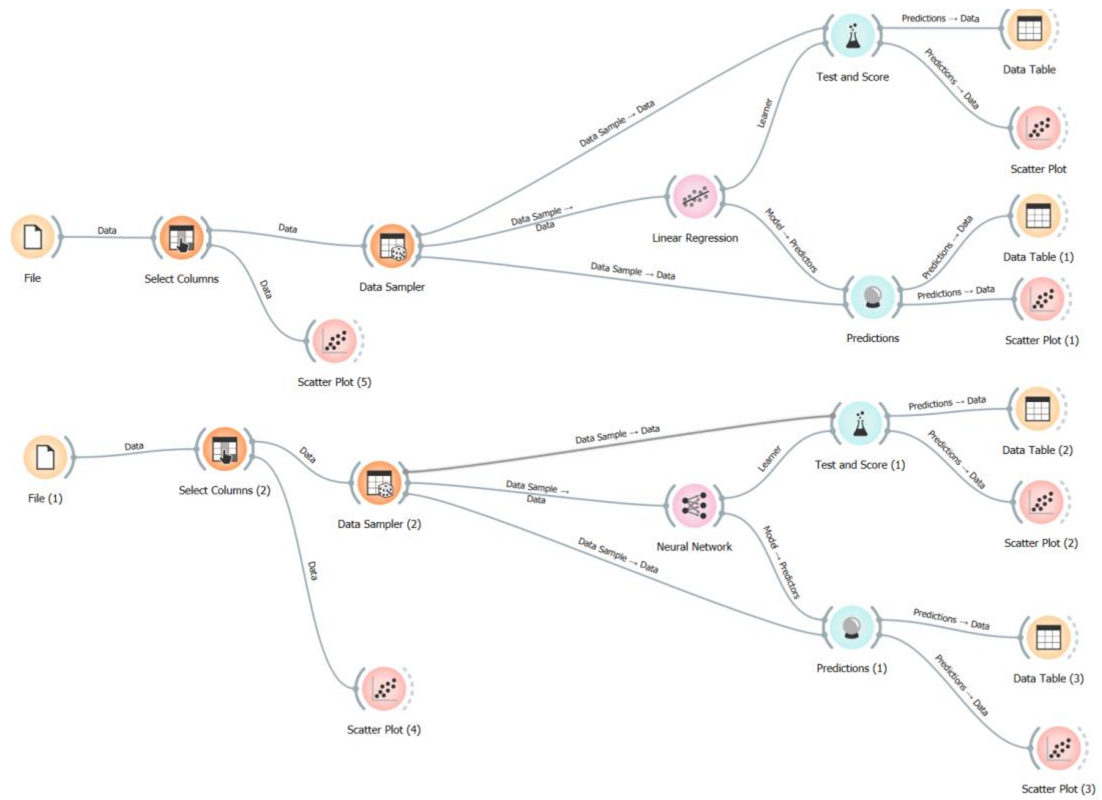
I have tested other features but they all hover around 0.2-0.3. This is to be expected since most phones features besides ram are similar. Usually, the constant differentiator would be memory ram capacity.

Question 3 Prediction of insurance price using Regression models (8 marks)

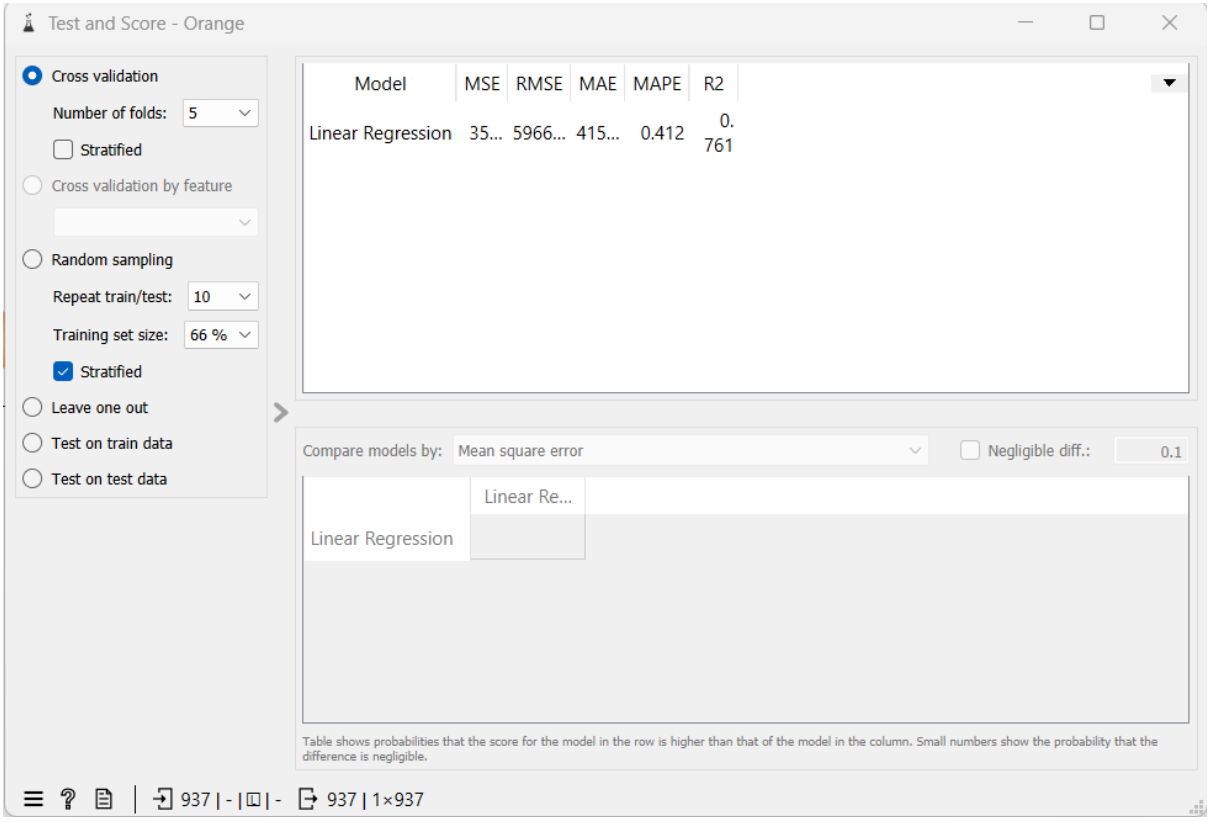
Question 3 Part A)

Develop a prototype with Machine Learning Model. Split the dataset into a training set and a testing set. Train the model using the training dataset, then evaluate its performance using the testing dataset. You must achieve an R2 score of at least 0.7 on the testing dataset. Take a screenshot of the result and include it in the submission document. (2 marks)

Model Development:

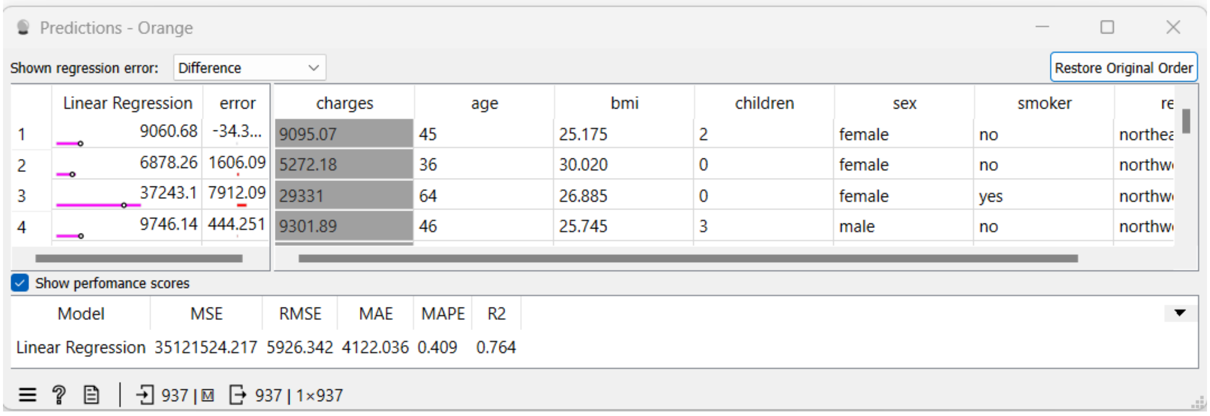


Test & Scoring of Model Performance: (Training Dataset, tested using Cross Validation)



R2 Score: 0.761

Prediction Performance: (Testing Dataset, Predicted Outcome)



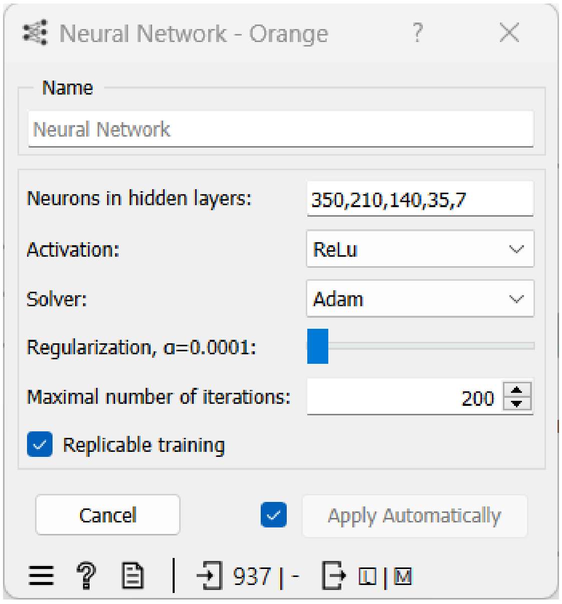
R2 Score: 0.764

Question 3 Part B)

Develop a prototype with Deep Learning Model.Split the dataset into a training set and a testing set. Train the model using the training dataset, then evaluate its performance using the testing dataset. You must achieve an R2 score of at least 0.8 on the testing dataset. Take a screenshot of the result and include it in the submission document. (2 marks)

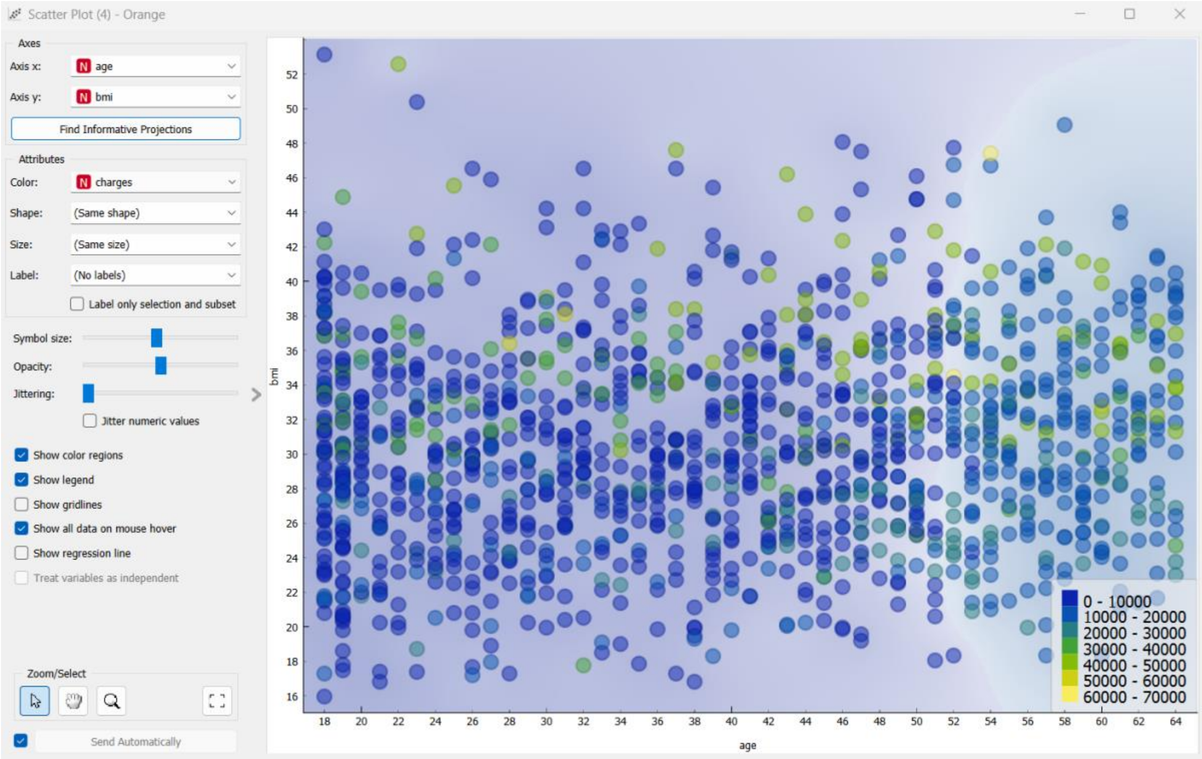
Development process:

Change the neural engine to these number of networks of nodes to allow the neural networks to form the relationships of the attributes.



Why did I use 7 as the final output node?

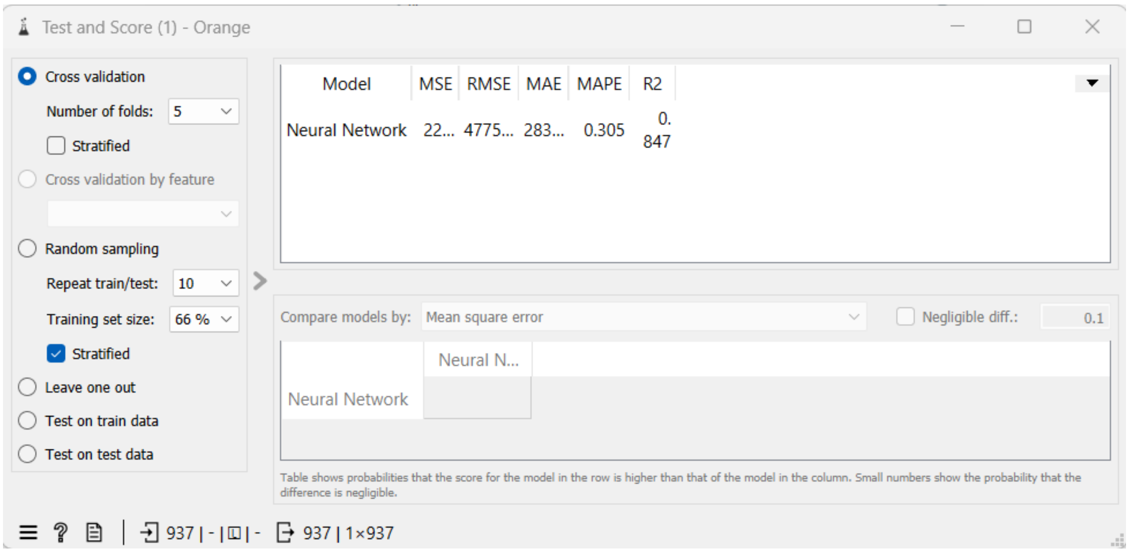
This is a good question let’s look at a scatter plot graph to understand the charges the insurance did give.



Notice how there are 7 tiers of insurance charges in the graph. Likely our dataset could have 7 tiers of insurance charges. For us to make a neural network that sorts the clients to the 7 classes the output node could be 7 nodes!

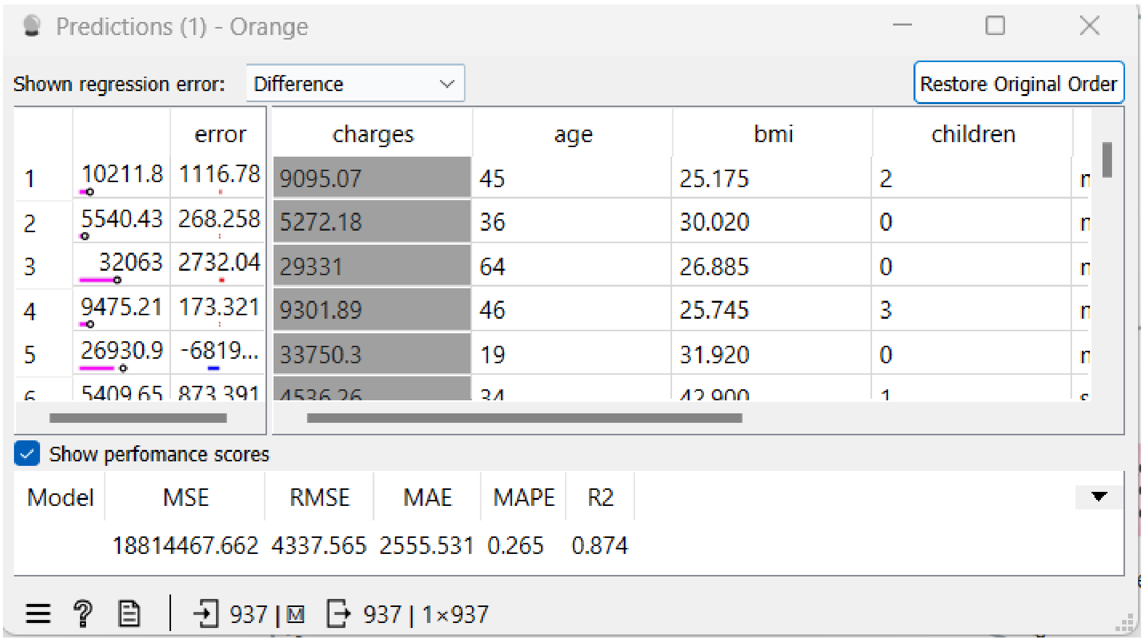
Based on all my testing this 7 nodes output gave me the highest R2 score amongst all the node combinations.

Test & Scoring of Model Performance:



R2 Score: 0.847

Prediction Performance:



R2 Score: 0.874

Question 3 Part C)

Evaluate the fairness of the insurance.csv dataset using AI Ethics fairness principles. Identify and justify 2 potential fairness issues that could arise when using this dataset to develop a machine learning/deep learning prediction application. What are the of causes the unfairness for each of the case? (4 marks)

Potential Issues:

In the AI Fairness Ethics Principle, we should not discriminate or factor gender, socioeconomic status, and ethnicity.

As AI is trained with real world data, these data may contain biasness from areas with less diversity or existing systemic biasness. Without applying any filter or controls this biasness will also spread to the AI.

For this Dataset, it is charging individuals health insurance. Health insurance should be marked by objective information of an individual risk profile and not use extremely narrow indicators to define their potential cost.

Here are the following potential Issues.

1 - Gender biases:

In our training of the insurance charges, we put assigned sex as a deciding factor. There is a possibility of hidden biasness that favours one over the other. Therefore, it should not be considered as part of the AI decision making process. Having assigned sex as a decision-making factor may create disparities between individuals. It may also reinforce gender stereotypes through AI.

18	female	25.08	0	no	northeast	2196.473
18	male	25.46	0	no	northeast	1708.001

Here are some rows I eyeballed in the dataset, while not exactly empirical, it helps us to understand the potential issue that may arise from using very few indicators to decide insurance price

18	male	30.14	0	no	southeast	1131.507
18	female	31.13	0	no	southeast	1621.883

For example, in the dataset given, if all factors are nearly equal besides gender, the AI may mistakenly assume that gender is a factor in giving a price difference. This is discriminatory because it does not evaluate the person insurance charge based on health risk.

2-Region Biases:

Regions may have hidden biases. Certain regions have lower socio-economic groups or a concentration of a single ethnicity. By using regions in our insurance parameters. We may be discriminating these individuals even though socio economic or ethnicity is not included.

26	18	female	30.115	0	no	northeast	2203.472
27	18	female	31.35	0	no	southeast	1622.189

Here is data that shows that almost all statistics are equal besides region. The AI if not applied with fairness may assume that region is justified for higher insurance price without considering any risk profiling. Individuals shouldn't be penalised based on where they are from. A holistic assessment is needed to better evaluate the insurance charge.

Overall, the AI should be evaluating the individual on a more personalised plan based on factors and parameters that does not reinforce society discriminations. Perhaps a better way to approach the insurance dataset is to use diverse range of individuals and different profiling for risks factors.

To enhance the dataset, perhaps including various health risks would better diversify the dataset.

Article I read to help me understand potential issues in this dataset:

1. Ownesens, J. (n.d.). Removal of Gender Bias in Insurance. Retrieved 2024, Jan 7. From the wns website: https://www.wns.com/perspectives/articles/articledetail/759/removal-of-gender-bias-in-insurance-creates-need-for-new-pricing-strategy-in-europe

Questions 4 Topics Classification from the text (6 Marks)

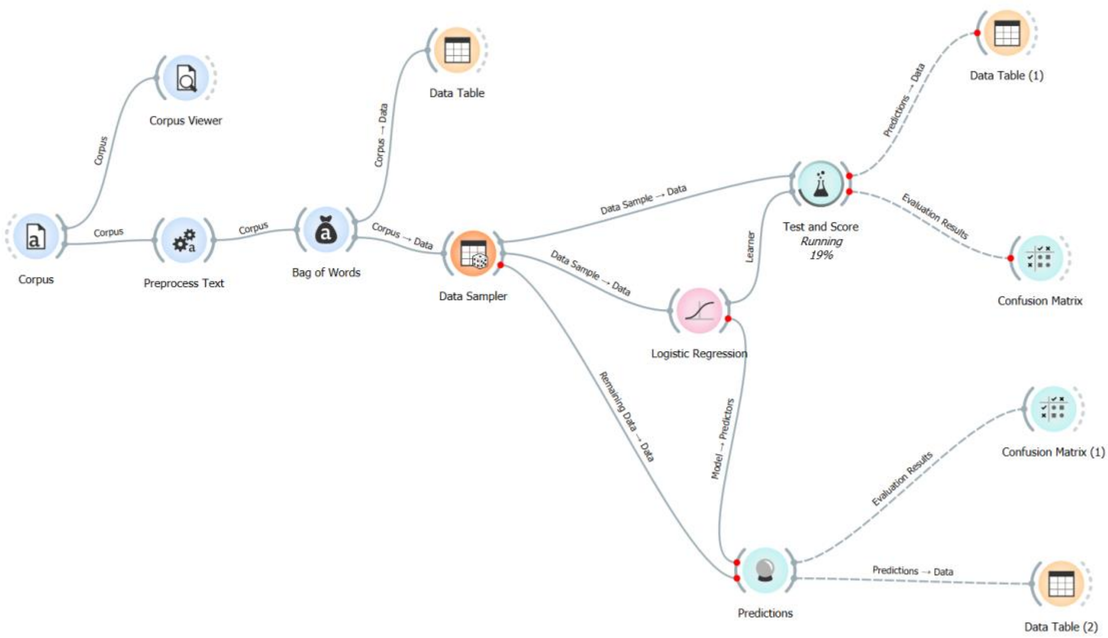
Given a dataset named "topics_dataset.tab" which includes sentences and corresponding labels indicating the topic of each sentence, complete the following tasks:

Question 4 Part A)

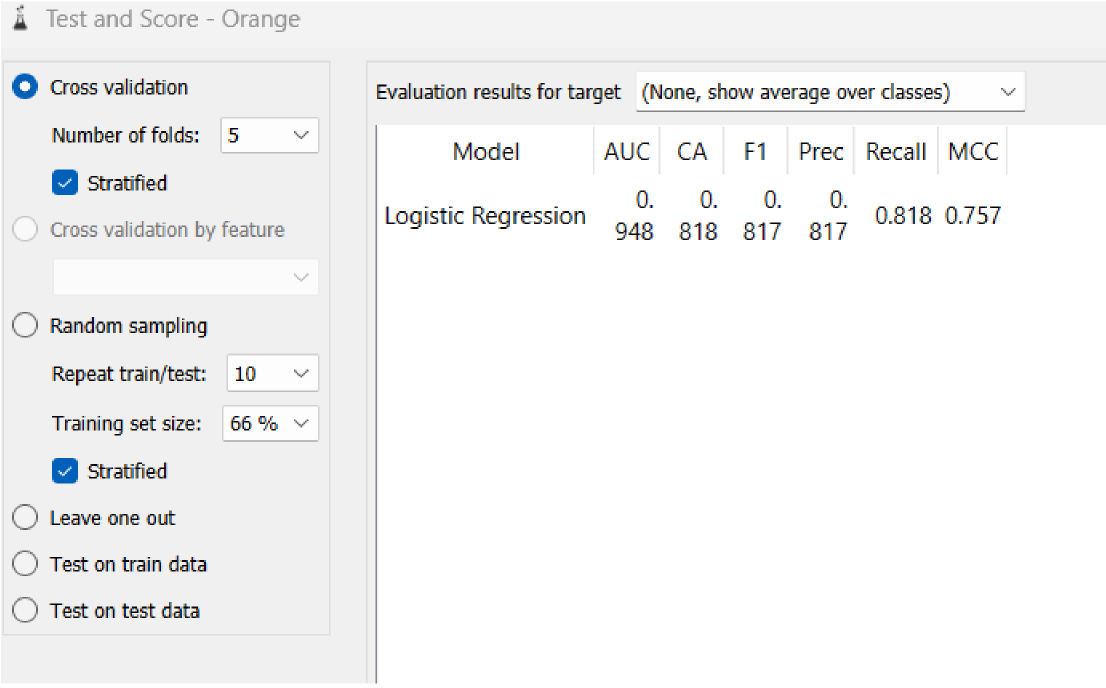
Format the text dataset into the Bag of Words using Python's Orange library.

Then, use the data to train Logistic Regression classifiers capable of classifying the four topics: 1-World, 2 Sports, 3-Business, and 4-Sci/Tech. The classifiers should have a F1-score of at least 0.8 and AUC score of at least 0.8 based on the test data. (2 marks)

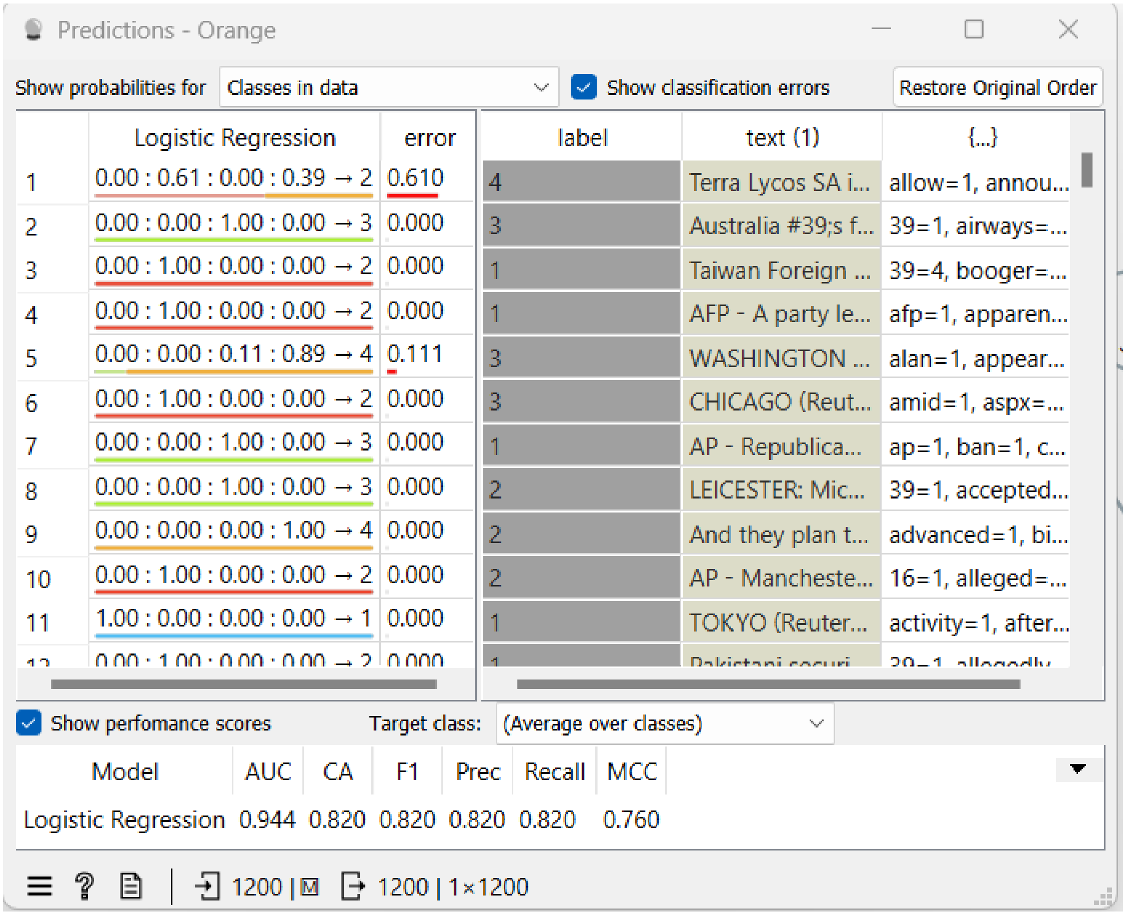
Model Development



Training Model Test and score:



Prediction Scores:



Question 4 Part B)

Explain the importance of text preprocessing in machine learning.

Provide three specific preprocessing techniques commonly used in natural language processing (NLP). (4 marks)

The importance of Preprocessing in Machine learning is to allow for any given document of text. The NLP can standardize the format of the text into a format which the NLP algorithm can easily understand. Next Preprocessing speeds up the performance of the machine learning model. Preprocessing also helps to denoise the context and to allow the machine learning to focus on what is needed to be trained on.

Here are the following Preprocessing techniques used.

Lower Casing

This involves baselining all words into lower case.

Take for example,

“FOUNDATION” and “foundation”

In a vector space model, these are considered as two different words.

By Baselining to all lowercase, we can group these similar words as one category instead of 2.

Lemmatization

Finding the root word of words, this is to reduce the inflection of repeated words.

Example, Before Lemmatization:

e.g. Happiness, Happy, Happiest

After lemmatization:

Happy

In a vector space, the above is considered as 3 separate categories. However, the context it is used all mean the same.

By lemmatizing them, we can group these frequency as one category, Happy.

Tokenization

Split the sentence into words which would be treated like tokens. In a normal context, the parameter may be put as a string input. However, in order for the machine learning to process the data it will need to be tokenized.

Before:

“This is a sentence”

After Tokenization:

“This”, “is”, “a”, “Sentence”.

This will be needed to tabulate the frequency of these tokens being used in each sentence.

Combining all three techniques is necessary to allow NLP to function efficiently and effectively.

Source used:
1. Harsith. (2019, Nov 21). Text Preprocessing in Natural Language Processing. Retrieved 2024, Jan 11. From the Towards Data Science Website: https://towardsdatascience.com/text-preprocessing-in-natural-language-processing-using-python-6113ff5decd8

2. De Silva, M. (2023, Apr 30). Preprocessing Steps for Natural Language Processing (NLP): A Beginner’s Guide. Retrieved 2024, Jan 11. From the Towards Data Science Website: https://medium.com/@maleeshadesilva21/preprocessing-steps-for-natural-language-processing-nlp-a-beginners-guide-d6d9bf7689c9

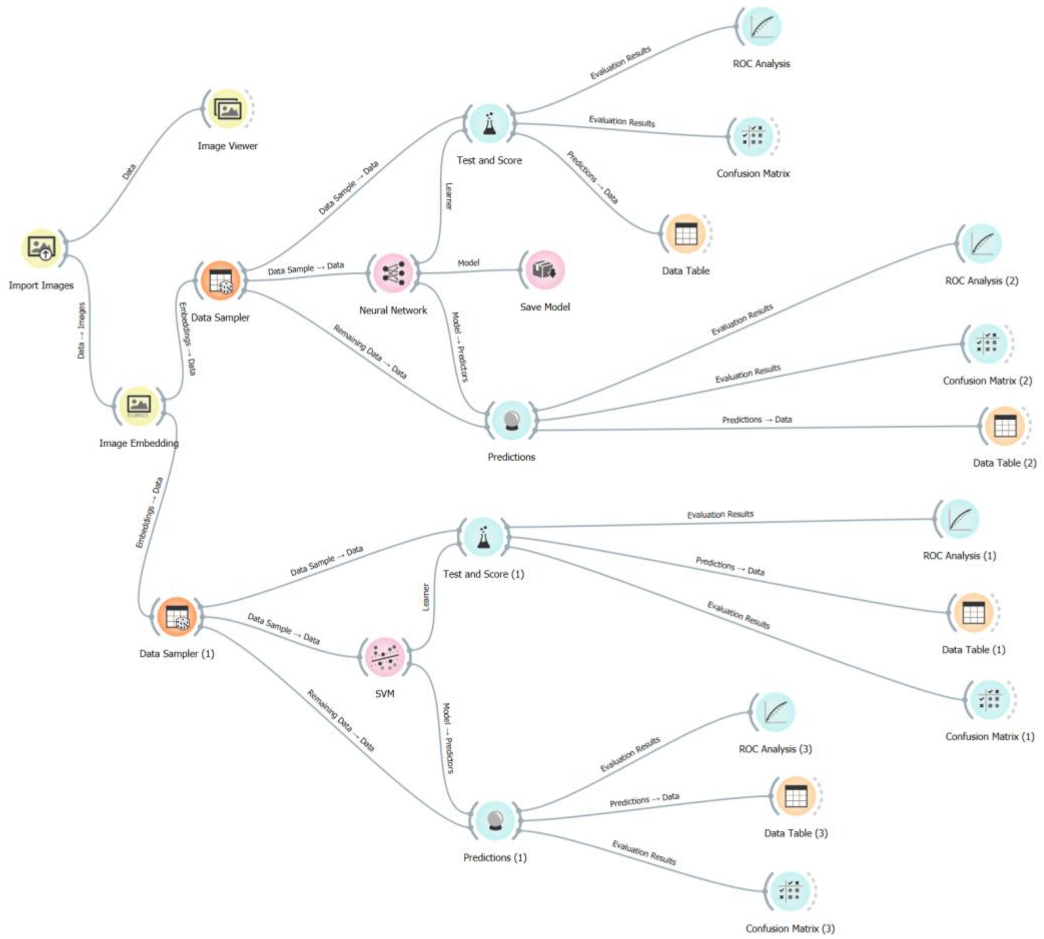
Questions 5 Image Classification Problems (5 Marks)

The folder contains 11 subfolders of different hand sign images namely a,b,c,d,e,i,m,n,s,x and y.

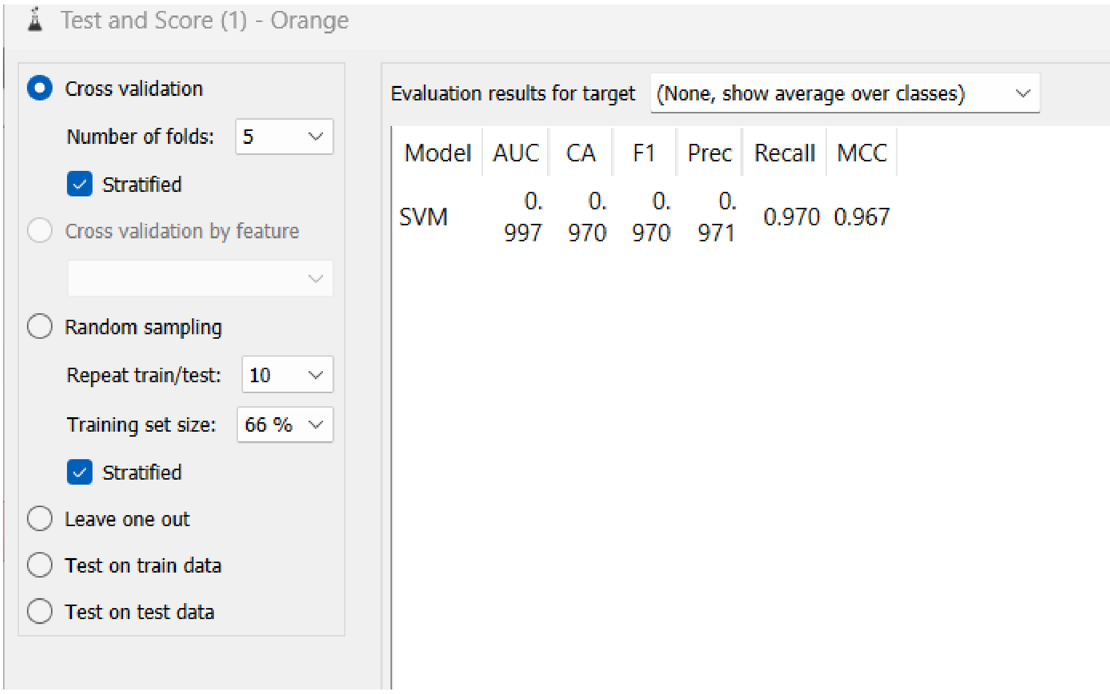
Question 5 Part A)

Use the image data in the directory to train a machine learning hand sign classifier and a neutral networker hand sign classifier. Both the models test data F1 score must be at least 0.9. Screen capture the result and include it in the submission document (2 marks)

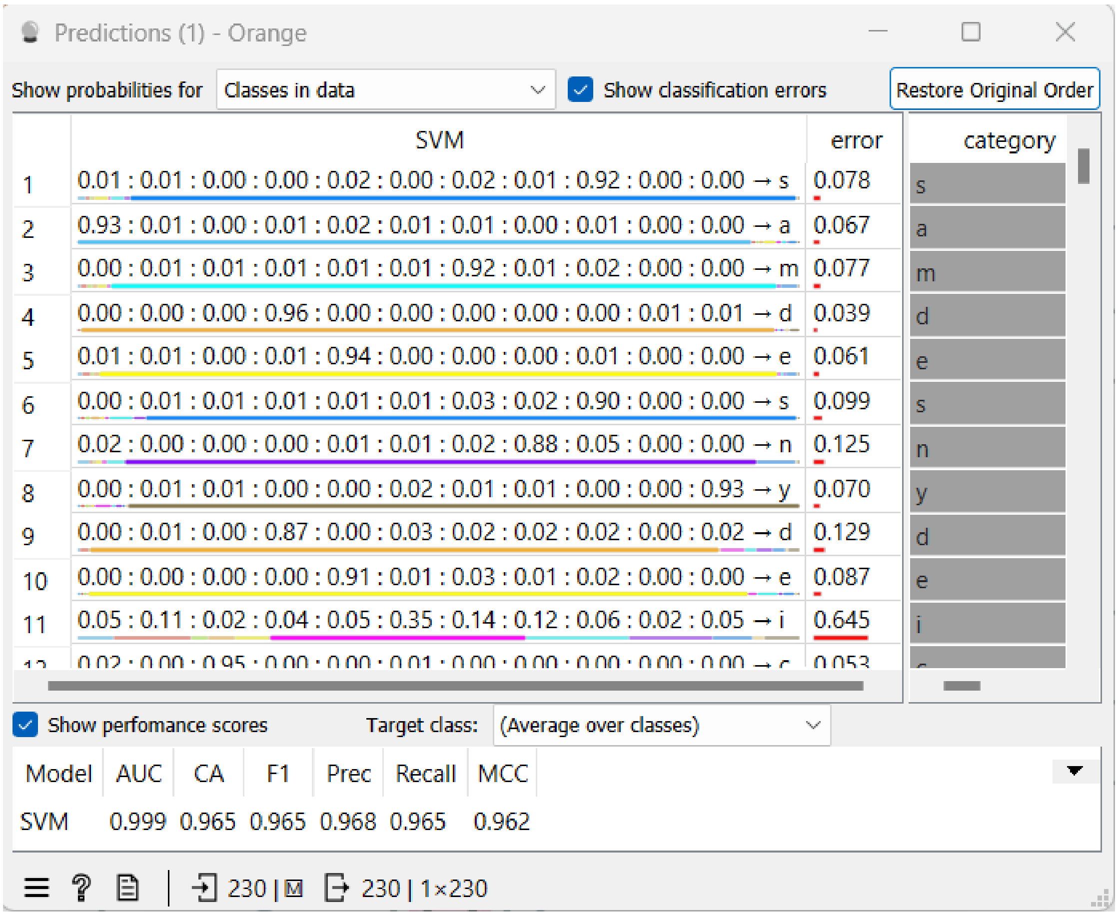
Model Development:



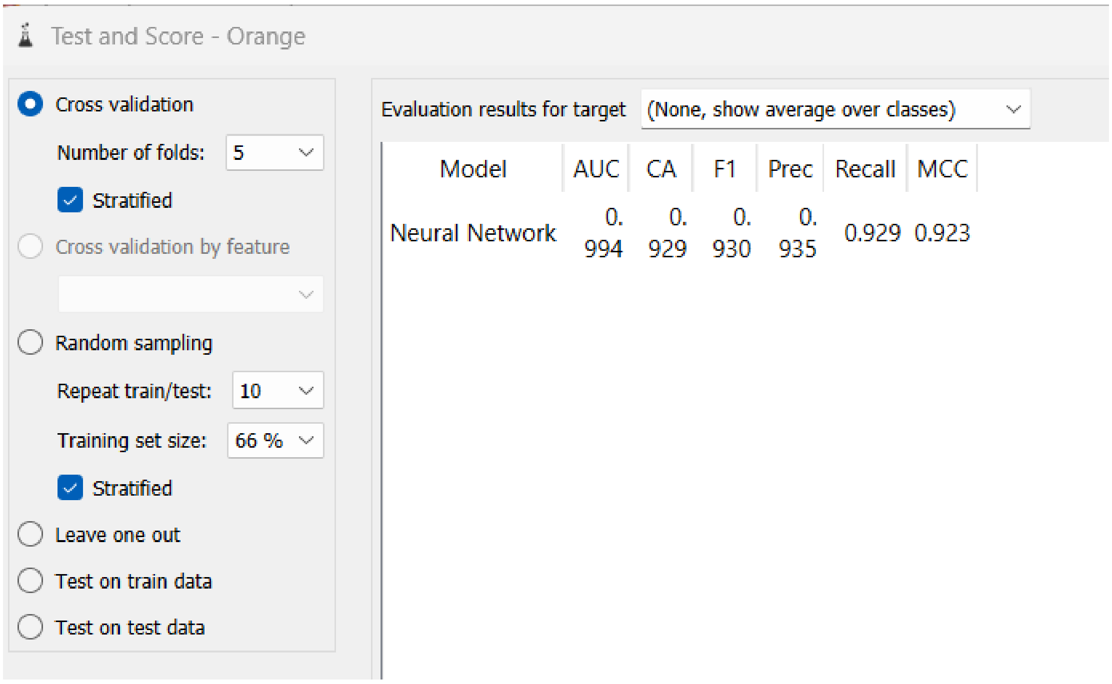
Machine Learning SVM Training scores:



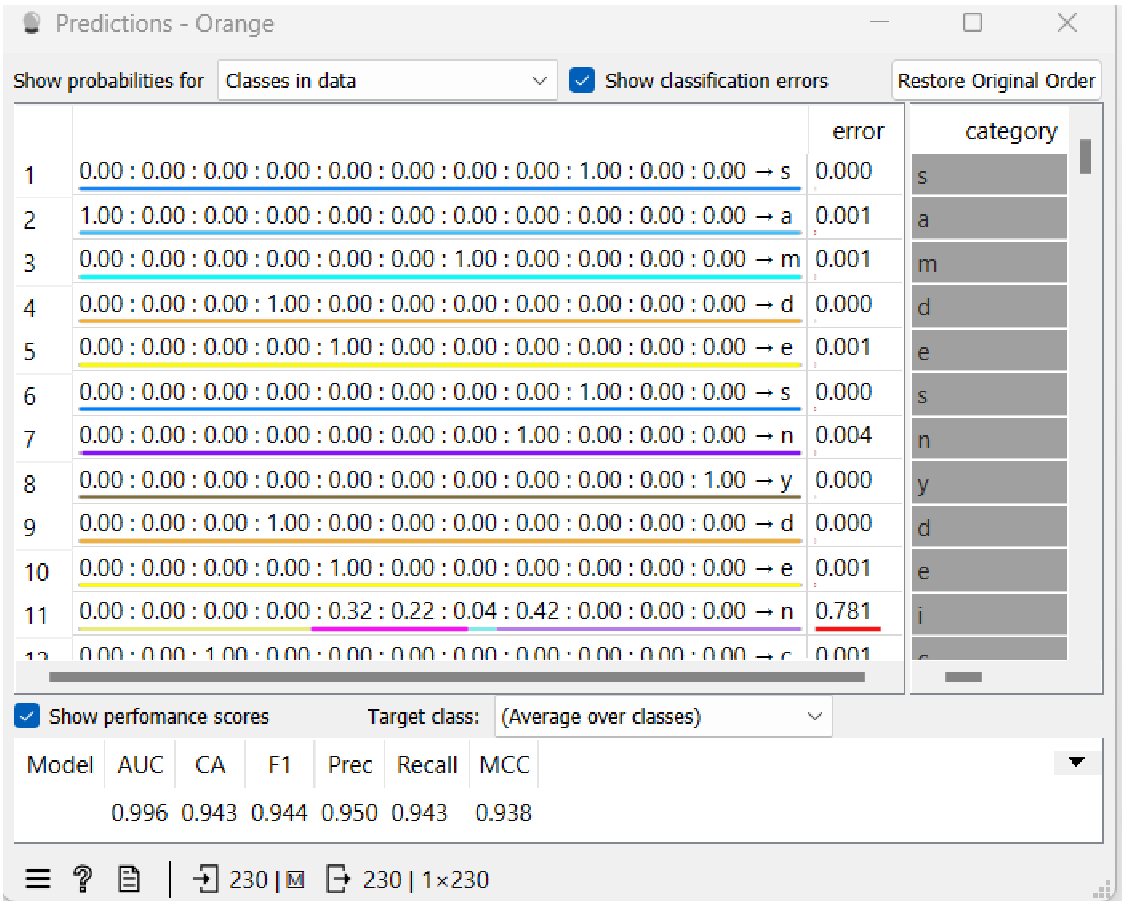
Machine Learning SVM Prediction Scores:



Neural Network Training Scores:



Neural Network Scores Prediction:



Question 5 Part B)

Suggest another performance metric to measure the performance of the 2 trained models. Show the result of the performance metric and use it to explain the model's performance. Suggest reason that contributed to the result (The result must be clearly included in the submission document) (3 marks)

Confusion Matrix is a performance metric that could be used to measure the performance of the 2 trained models. Confusion Matrix works by testing the actual and predicted classification in a true positive, false positive, true negative, false negative table. The greater the ratio of true positives to the false negative & false positive the more accurate the Trained models are.

Here is the result of the performance matrix.

Testing of Neural Network Model performance:

Confusion Matrix - Orange

Learners

Neural Network

Predicted

Actual	a	b	c	d	e	i	m	n	s	x	y	∑
a	58	0	0	0	0	0	2	0	0	0	0	60
b	0	39	3	0	0	0	0	6	0	0	0	48
c	0	0	47	0	0	0	0	0	0	2	0	49
d	0	0	0	43	1	0	2	1	0	0	0	47
e	0	0	0	0	44	0	0	1	0	0	0	45
i	0	1	0	1	0	40	1	3	0	0	0	46
m	0	0	0	0	1	1	46	2	0	0	0	50
n	0	0	0	0	2	0	1	46	1	0	0	50
s	0	0	0	0	0	0	1	1	42	1	0	45
x	0	0	0	2	0	0	0	0	0	45	0	47
y	0	0	0	0	0	0	0	1	0	0	51	52
∑	58	40	50	46	48	41	53	61	43	48	51	539

Prediction of Neural Network Model Performance:

Confusion Matrix (2) - Orange

Learners

Predicted (columns) / Actual (rows)

	a	b	c	d	e	i	m	n	s	x	y	∑
a	18	0	0	0	0	0	0	0	0	0	0	**18**
b	0	18	1	1	0	0	0	1	1	0	0	**22**
c	0	0	21	0	0	0	0	0	0	0	0	**21**
d	0	0	0	21	0	0	0	1	0	0	0	**22**
e	1	0	0	0	15	0	0	0	0	0	0	**16**
i	0	0	0	0	0	22	2	1	0	0	0	**25**
m	0	0	0	0	0	0	20	0	0	0	0	**20**
n	0	0	0	0	0	0	0	20	0	0	0	**20**
s	0	0	0	0	0	0	0	2	22	1	0	**25**
x	0	0	0	0	0	0	0	0	0	23	0	**23**
y	0	0	0	0	0	0	1	0	0	0	17	**18**
∑	**19**	**18**	**22**	**22**	**15**	**22**	**23**	**25**	**23**	**24**	**17**	**230**

Performance of Machine Learning SVM model

Confusion Matrix (1) - Orange

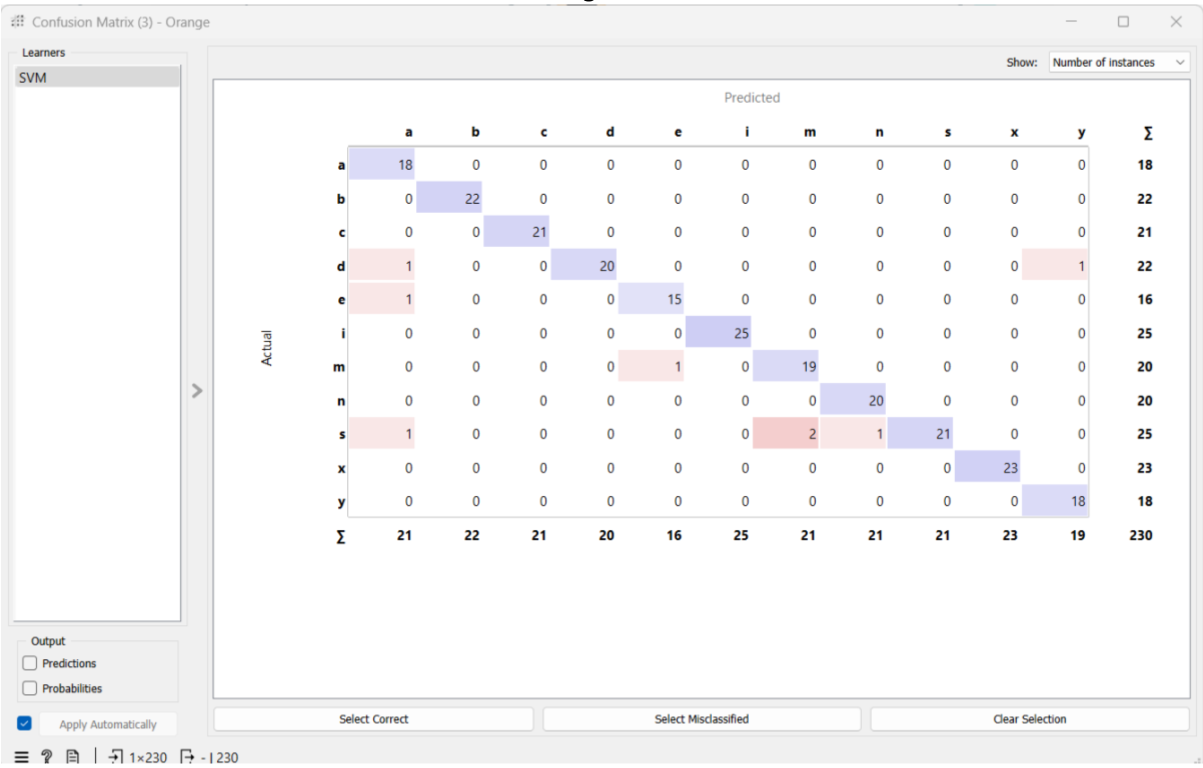
Learners

SVM

Predicted (columns) / Actual (rows)

	a	b	c	d	e	i	m	n	s	x	y	∑
a	59	0	0	0	0	0	1	0	0	0	0	**60**
b	0	47	0	0	0	0	1	0	0	0	0	**48**
c	0	0	48	0	0	0	0	0	0	1	0	**49**
d	0	1	0	46	0	0	0	0	0	0	0	**47**
e	3	0	0	0	42	0	0	0	0	0	0	**45**
i	0	1	0	1	0	42	0	0	0	0	2	**46**
m	0	0	0	0	1	1	48	0	0	0	0	**50**
n	0	0	0	0	0	0	0	50	0	0	0	**50**
s	1	0	0	0	0	0	1	1	42	0	0	**45**
x	0	0	0	0	0	0	0	0	0	47	0	**47**
y	0	0	0	0	0	0	0	0	0	0	52	**52**
∑	**63**	**49**	**48**	**47**	**43**	**43**	**51**	**51**	**42**	**48**	**54**	**539**

Prediction Performance of SVM Machine Learning Model:



Actual \ Predicted	a	b	c	d	e	i	m	n	s	x	y	Σ
a	18	0	0	0	0	0	0	0	0	0	0	18
b	0	22	0	0	0	0	0	0	0	0	0	22
c	0	0	21	0	0	0	0	0	0	0	0	21
d	1	0	0	20	0	0	0	0	0	0	1	22
e	1	0	0	0	15	0	0	0	0	0	0	16
i	0	0	0	0	0	25	0	0	0	0	0	25
m	0	0	0	0	1	0	19	0	0	0	0	20
n	0	0	0	0	0	0	0	20	0	0	0	20
s	1	0	0	0	0	0	2	1	21	0	0	25
x	0	0	0	0	0	0	0	0	0	23	0	23
y	0	0	0	0	0	0	0	0	0	0	18	18
Σ	21	22	21	20	16	25	21	21	21	23	19	230

In general, the Deep Learning Neural Network Model tends to have less false positives than the SVM model. But what we are more curious about is what lead to the model to make a false positive?

A good example would be looking at the Neural Network letter I data table and see for the false positive!

51	i	u5_l8	i/u5_l8.jpg	19392	320	240	m

Here the image labelled u5_l8, from the I folder is falsely labelled as m.

The False positive image from I folder, U5_l8

Sample M hand sign image.

The false positive from my observation is due to the Pinky finger being ignored as the ai looks for similarity between L and M. Notice how the knuckles are arched so similarly from the L image to the M. It is likely the high similarity % of the hand sign allowed the AI to make a mistake.

Notice how the Pinky skin colour has a higher contrast to the person knuckles. The AI may have mistaken the Pinky as white noise during this comparison.