MACHINE LEARNING – Continuous Assessment

Team 2: Balaga Kishore Kumar, Cao Guoan, Chan Qingshuang, Chua Lee Siong, Dhilip Kumar Veerapandi, Ei Phyu Khin, Judy Choo Sang San, Lee Chenghao, Liu Xiaolin

Part 1: Supervised Machine Learning

Dataset source:
https://www.kaggle.com/ronitf/heart-disease-uci

Problem statement:
This dataset records the attributes of a group of patients and whether they have heart disease. From this dataset, we would like to be able to predict the presence of heart disease in patients.

Data Dictionary of dataset:

Attribute Name	Type	Definition
age	N	Age of patient in years
sex	N	Gender of patient (1=male;0=female)
cp	N	Chest pain type (4 values)
trestbps	N	Resting blood pressure (in mm Hg on admission to the hospital)
chol	N	Serum cholesterol in mg/dl
fbs	N	Fasting blood sugar > 120 mg/dl (1=true; 0= false)
restecg	N	Resting electrocardiographic results (values 0,1,2)
thalach	N	Maximum heart rate achieved
exang	N	Exercise induced angina (1=yes; 0= no)
oldpeak	N	Oldpeak = ST depression induced by exercise relative to rest
slope	N	The slope of the peak exercise ST segment
ca	N	Number of major vessels (0-3) coloured by flourosopy
thal	N	Thalassemia: 3 = normal; 6 = fixed defect; 7 = reversable defect
target	N	The presence of heart disease (1=no; 0= yes)

N: Numerical

No of records & columns in the dataset:
- 303 records and 14 columns

No of records & columns in the training / test dataset:
- Training – 227 rows, 6 columns (cp, exang, thal, slope, oldpeak, ca)
- Test – 76 rows, 6 columns (cp, exang, thal, slope, oldpeak, ca)
  (Variations of columns/features were tested during model building)

Duration of training for each model:
- Logistic Regression: 0.00699s
- K-NN Classification: 0.00296s
- Support Vector Machines: 0.021941s
- Decision Tree, Random Forest: 0.001247s, 0.060034s

Process Flow:

Data engineering

- The original dataset is clean – it does not have improperly formatted data, missing data or NaN values. Data has also been encoded, e.g. for Thalassemia (3 = normal; 6 = fixed defect; 7 = reversable defect). Hence, we did not perform data cleansing on this dataset.

Feature engineering

Using the SVM Model

- First, we selected features that are most relevant to the analysis, e.g. cholesterol and blood pressure as those describe prime factors for heart disease. We tested a model built on this feature combination, and received an accuracy score of 53.94%.
- Next, we identified the correlation between features using df.corr(), and selected features that have high correlation with the output to build the model, e.g. thalach and oldpeak readings.
- Below are the training duration and accuracy scores of the SVM model to illustrate how these improve after simple feature selection:

feature1	feature2	feature3	feature4	feature5	Training duration (s)	Accuracy Score (%)
chol	trestbps				0.07347	53.94
thalach	oldpeak				0.01558	78.95
thalach	oldpeak	cp			0.01559	81.57
thalach	oldpeak	cp	exang		0.01559	88.16
thalach	oldpeak	cp	exang	ca	0.01561	86.84

Using the Logistic Regression Model

- Using the logistic regression model, we applied a different technique. First, we calculated the accuracy scores for models built on each individual feature.

```
age's accuracy = 0.618421052631579
sex's accuracy = 0.6447368421052632
chest_pain_type's accuracy = 0.7763157894736842
resting_blood_pressure's accuracy = 0.47368421052631576
cholesterol's accuracy = 0.5526315789473685
fasting_blood_sugar's accuracy = 0.5526315789473685
rest_ecg's accuracy = 0.6052631578947368
max_heart_rate_achieved's accuracy = 0.6842105263157895
exercise_induced_angina's accuracy = 0.7763157894736842
st_depression's accuracy = 0.7368421052631579
st_slope's accuracy = 0.7631578947368421
num_major_vessels's accuracy = 0.75
thalassemia's accuracy = 0.7763157894736842
target's accuracy = 1.0
```

- Next, we combined the features with top accuracy scores to find the best combination for the model. For example, a model built with the top 6 features produces the highest accuracy score 90.79%.
- The top 6 features are cp (chest pain type), exang (exercise induced angina), thal (thalassemia), slope (the slope of the peak exercise ST segment), oldpeak (ST depression induced by exercise relative to rest) and ca(number of major vessels).

Conclusion:
Based on this dataset, the presence of heart disease can be predicted best with the following factors:

cp	Chest pain type
exang	Exercise induced angina
thal	Thalessemia
slope	The slope of the peak exercise ST segment
oldpeak	ST depression induced by exercise relative to rest
slope	The slope of the peak exercise ST segment
ca	Number of major vessels coloured by fluoroscopy

What we have learnt:
Features are important for predictive models and will determine the results that we achieve. We need to test and improve features as needed until the model yields the best results. The relevance and quantity of the features will determine the accuracy and speed of the model.

##Note: We have provided the code in Jupyter Notebook and also PDF format.

Part 2: Unsupervised Machine Learning

Dataset source:
https://www.kaggle.com/lava18/google-play-store-apps

Problem statement:
This dataset records the attributes of Android mobile applications in the Google Play Store. From this dataset, we would like to be able to find the best clustering results/optimum number of clusters.

Data Dictionary of dataset:

Attribute Name	Type	Definition
App	A	Name of app
Category	A	Category of app, e.g. Art and Design
Rating	N	Rating scale
Reviews	N	Number of reviews
Size	A	Size of app in KB or MB
Installs	A	Number of installations
Type	A	Type of app, free or paid
Price	A	Price of app in dollars
Content Rating	A	Audience for app, e.g. Everyone
Genres	A	Genre of app, e.g. Health and Fitness
Last Updated	D	Date the app was last updated
Current Version	A	Version of app
Android Version	A	Version of app for android

A: Alphanumerical
N: Numerical

No of records & columns in the dataset:
- 10842 records and 13 columns

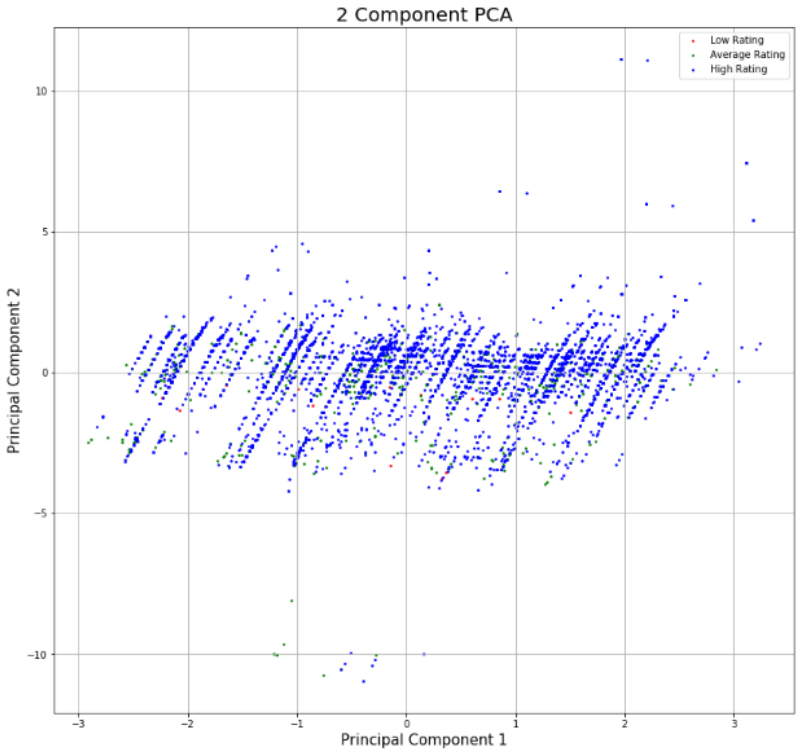
No of records & columns in the training / test dataset:
- 7723 records and 9 columns after data cleansing

Process Flow:

Data engineering
- The first step is to cleanse the data. As this is a fairly large dataset with 10000+ rows, there are several steps that we need to take.
- *Remove all null value rows*: The column with the most null values is the rating columns. We removed all rows containing null values.
- *Drop irrelevant columns*: We removed the columns 'App', 'Last Updated', 'Current Ver' and 'Android Ver' as we believe these will not contribute greatly to the clustering.
- *Perform encoding of values*: As almost all of the columns contain alphanumerical data, we performed encoding for the columns 'Content Rating', 'Category', 'Type', 'Genre', 'Installs'.
- *Remove unnecessary characters*: We removed $ (dollar sign), 'K' and 'M' (kilobyte and megabyte).
- After cleansing, we imported the data to a new CSV file.

PCA

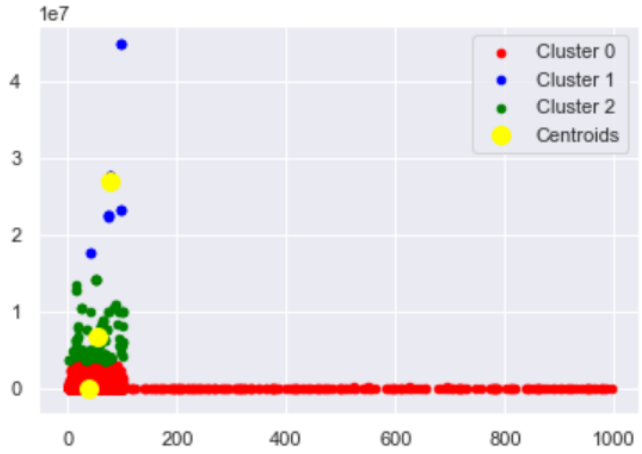
- Next, we conducted PCA (feature reduction) and reduced all the columns to 2 principal components.
- We selected the 'Rating' column as the target. ('Low Rating', 'Average Rating' and 'High Rating'.)



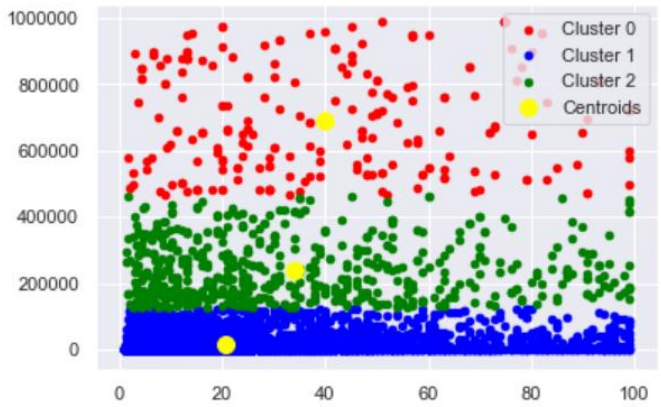
- The explained variance tells us how much information (variance) can be attributed to each of the principal components.
- Together, the first two principal components contain 41.35% of the information. The first principal component contains 22.66% of the variance and the second principal component contains 18.69% of the variance. The remaining principal components contain the rest of the variance of the dataset.

k-Means

- At this point, we attempted to conduct k-Means clustering.
- First, using the elbow method, we found that the optimal number of clusters is 3.
- We ran K-Means with 3 clusters and visualized the clusters by selecting 'Size' and 'Reviews'.



- The resulting clusters did not seem to be clear enough – there is a long tail that disturbs the visualization.
- Hence, we filtered the dataset to 'Size' < 100 and 'Reviews' <= 1000000 to get a better visualization.

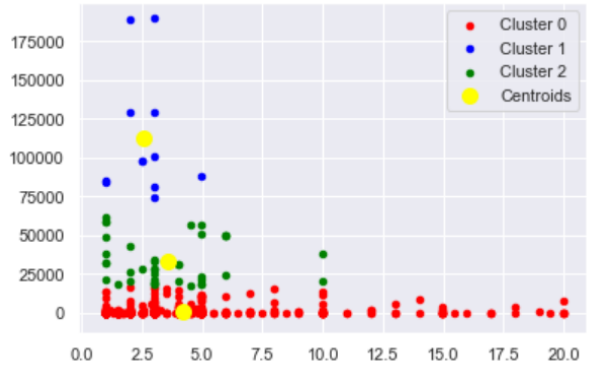


- The resulting cluster is much clearer - we can interpret the segmentation as follows: Cluster 1: Small apps with very limited reviews. Cluster 2: Medium sized apps with some reviews. Cluster 0: Large apps with many reviews.

- We ran a model evaluation - the homogeneity, completeness, v-measure, and ARI scores were all very low, indicating that K-Means is not suitable here. This could be due to the odd shapes of the clusters. Only the silhouette coefficient is high here, indicating good/dense clustering.

```
Homogeneity: 0.029
Completeness: 0.040
V-measure: 0.034
Adjusted Rand Index: -0.085
Adjusted Mutual Information: 0.029
Silhouette Coefficient: 0.852
```

- Next we tried to run K-Means by selecting 'Price and 'Reviews'. For this attempt, we filtered out the free apps, as we assume a business would be interested in paid apps only.



- The resulting clusters are as above. From the chart we can see 3 clusters. Cluster 1: Expensive apps with very limited reviews. Cluster 2: Medium priced apps with some reviews Cluster 0: Cheaper apps with many reviews.

Applying PCA

- We then applied PCA to the k-Means model and ran another model evaluation.

```
Homogeneity: 0.002
Completeness: 0.001
V-measure: 0.001
Adjusted Rand Index: -0.005
Adjusted Mutual Information: 0.001
Silhouette Coefficient: 0.446
```

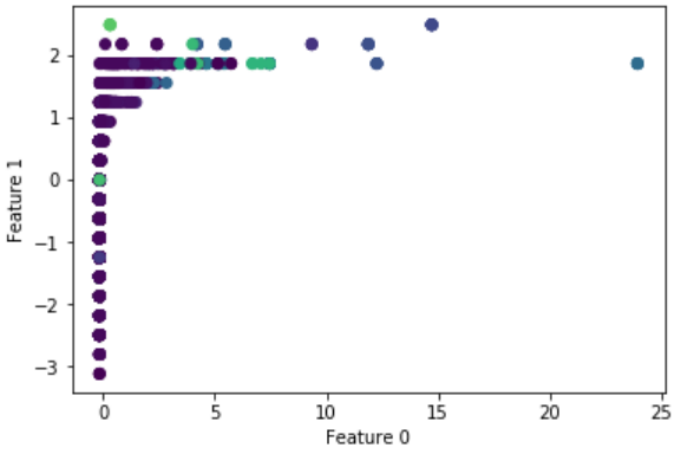
- We conclude that applying PCA did not improve the model evaluation scores.

DBSCAN

- Following on k-Means clustering, we next performed DBSCAN on the dataset.
- From various experiments in adjusting the values of epsilon, minimum samples and columns using a for-loop, we obtained the findings below:-

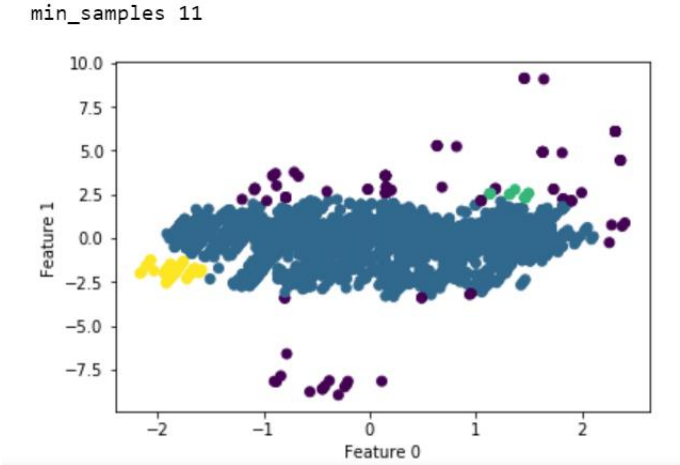
Before PCA

Epsilon Value	Min Samples	Columns	Number of clusters
0.5	2	All columns	46



After PCA

Epsilon Value	Min Samples	Columns	Number of clusters
0.3	11	2 (Principal Columns 1 & 2)	3



- To get a clearer clustering, we identified the optimum epsilon value to be 0.3 and number of samples to be 11.
- We observe that DBSCAN struggles with high dimensionality data. If given data with too many dimensions, DBSCAN suffers – there were 46 clusters before we performed PCA.
- After performing PCA, this helped to separate clusters of high density versus clusters of low density within the dataset. This also helped to handle outliers within the dataset.

Conclusion:
K-Means is more suitable for our current dataset rather than DBSCAN. The optimum number of clusters is 3 – more clearly defined clusters were created when we selected 'Size' and 'Reviews', or 'Price' and 'Reviews'.

What you have learnt:
As our data consisted mainly of categorical values, PCA did not make a big improvement to our models. The expected variance was also too low, hence we did not get a clear diagram. In this case, we had to rely on other methods to get the optimum epsilon values, sample size and k-value.

##Note: We have provided the code in Jupyter Notebook and also PDF format.

Thank you for reading!