# Cure Deliverable 4

## Due: Nov 19, 2023

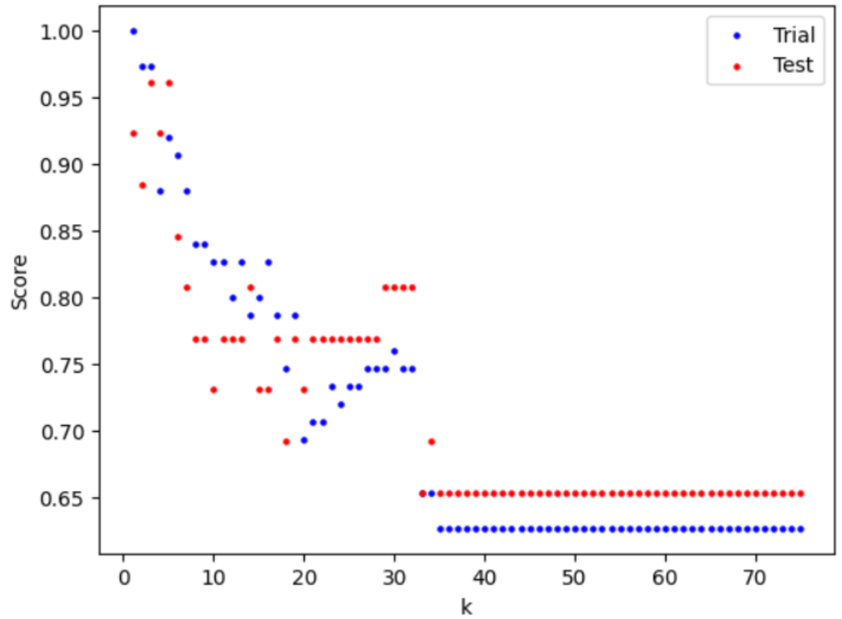
## Made By:

Group 103

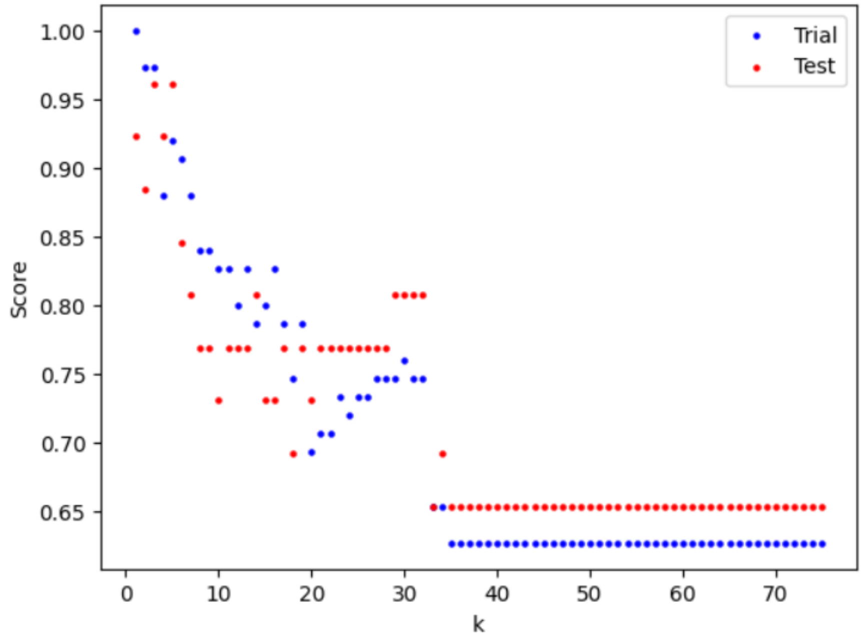
## Editors:

Opeyemi Sanyaolu (30156143)
Mendjemo Ngangom Gerard Ledoux (30192835)
Kosy Onyejemezi (30189156)
Chidera Awiaka (30165302)

(a) The graphs of model accuracy versus k values for different scenarios described in Task 2



Performance Comparison of Model Accuracy with Varying k Values
(MinMaxScaler)



Performance Comparison of Model Accuracy with Varying k Values
(StandardScaler)

(b) Final model parameters and scaler and accuracy:

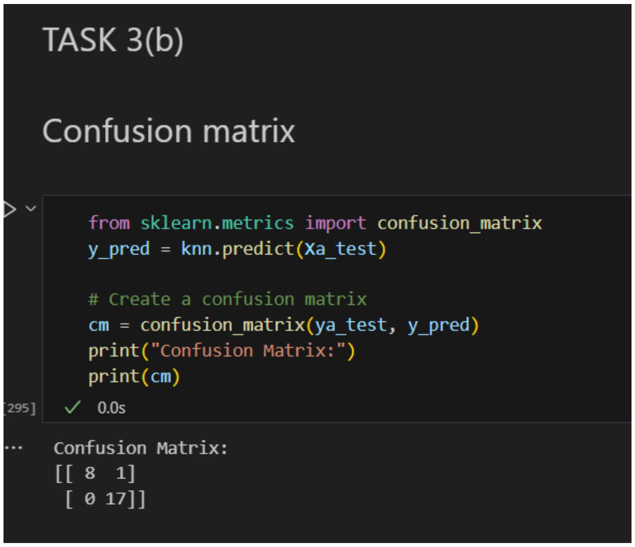
k = 3

Scaler used for attribute preprocessing: MinMaxScalar()

Model accuracy in the training set = 0.9733333333333334

Model accuracy in the test set = 0.9615384615384616

Confusion matrix:



TASK 3(b)

Confusion matrix

```
from sklearn.metrics import confusion_matrix
y_pred = knn.predict(Xa_test)

# Create a confusion matrix
cm = confusion_matrix(ya_test, y_pred)
print("Confusion Matrix:")
print(cm)
```

```
Confusion Matrix:
[[ 8  1]
 [ 0 17]]
```

Brief interpretations of confusion matrix:

A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. It's a useful tool to understand the performance of a classification algorithm by displaying the counts of the true positive, true negative, false positive, and false negative predictions made by the classifier.

The matrix is of the form:

[[true_negative false_positive]
 [false_negative true_positive]]

Where:

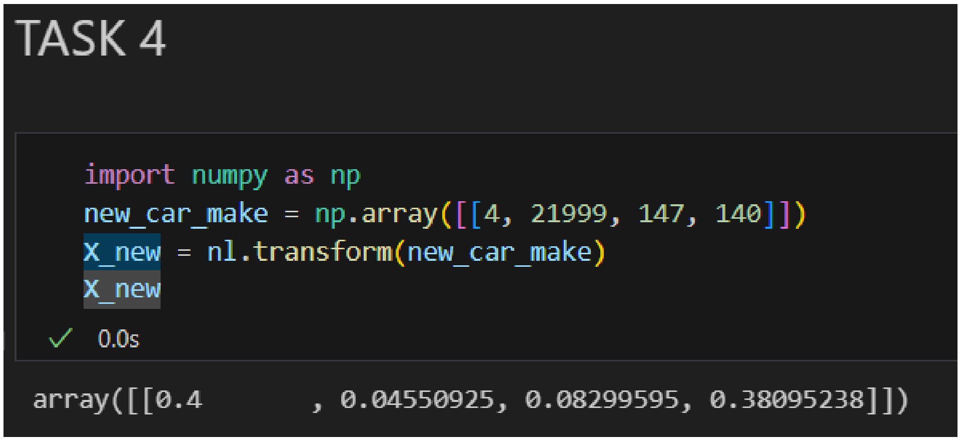
- true_negative: Number of instances correctly predicted as negative.
- true_positive: Number of instances correctly predicted as positive.
- false_negative: Number of instances wrongly predicted as negative.
- false_positive: Number of instances wrongly predicted as positive.

(c) Ans of task 4.

New instance attribute values:

Due to the car_make column of the original excel datasheet not being numerical, the values have been modified to give numerical values to the car brand (the car make is divided into the car brand and model). This generalisation makes predictions more accurate for a small data set.

4 represents a Hyundai in the updated dataframe.



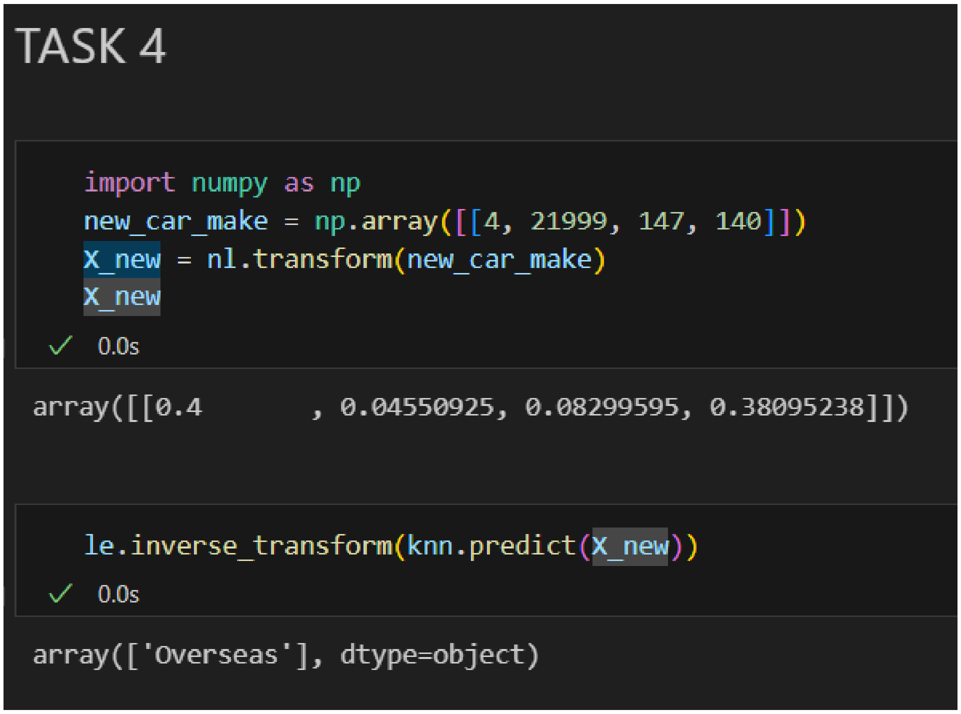
TASK 4

```
import numpy as np
new_car_make = np.array([[4, 21999, 147, 140]])
X_new = nl.transform(new_car_make)
X_new
```

✓ 0.0s

```
array([[0.4       , 0.04550925, 0.08299595, 0.38095238]])
```

Predicted class:



TASK 4

```
import numpy as np
new_car_make = np.array([[4, 21999, 147, 140]])
X_new = nl.transform(new_car_make)
X_new
```

✓ 0.0s

```
array([[0.4       , 0.04550925, 0.08299595, 0.38095238]])
```

```
le.inverse_transform(knn.predict(X_new))
```

✓ 0.0s

```
array(['Overseas'], dtype=object)
```

The model predicted the right Category (Overseas).