MSc in Software Design with Artificial Intelligence — Continuous Assessment Project (40%)

Dataset Links:
- Regression Dataset: Auto MPG (UCI Machine Learning Repository):
https://archive.ics.uci.edu/ml/datasets/auto+mpg

- Classification Dataset: Breast Cancer Wisconsin (Diagnostic):
https://archive.ics.uci.edu/ml/datasets/banknote+authentication

1. Regression

1.1 Objectives

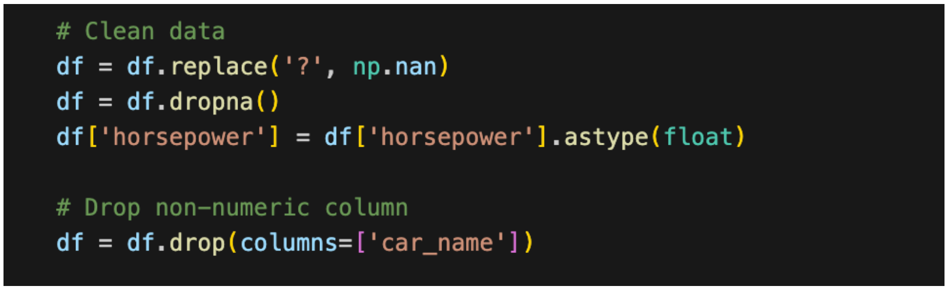
Predicting an automobile's fuel efficiency (miles per gallon) based on its engine and physical characteristics, including horsepower, weight, and displacement, is the aim of this analysis. From a business standpoint, this aids analysts and manufacturers in designing fuel-efficient cars and optimising engine performance.

Target Variable: mpg (Miles per Gallon)

Target variable: mpg.

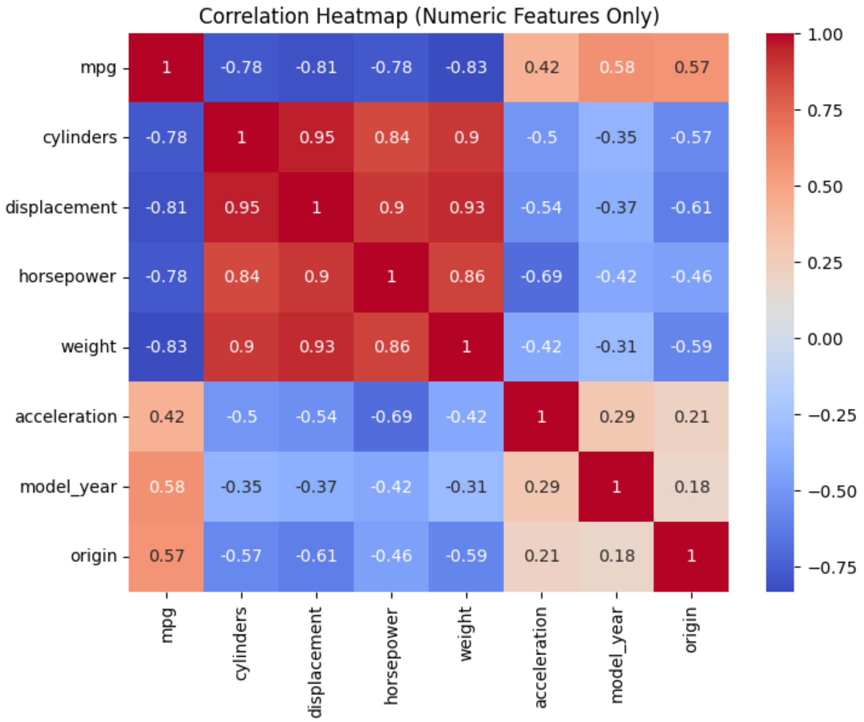
1.2 Data Exploration

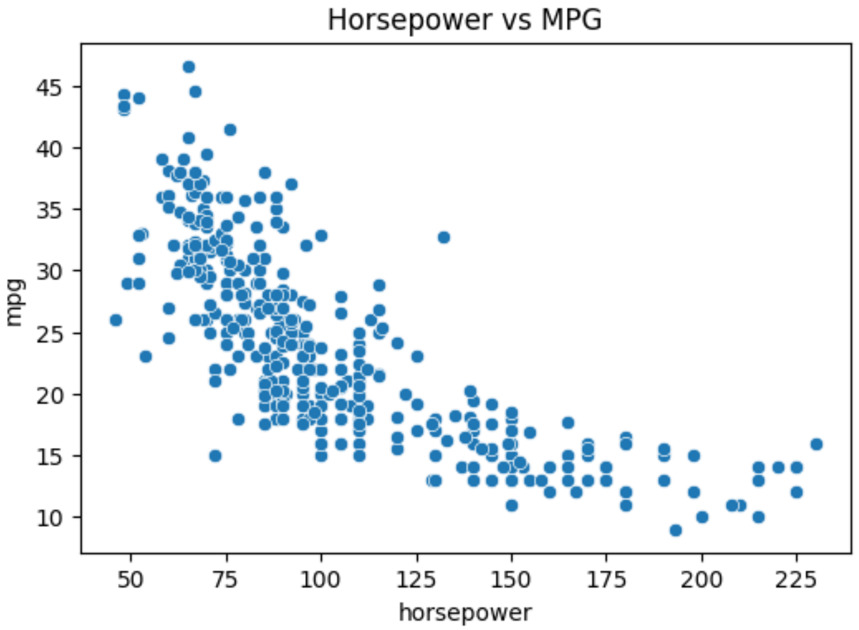
In addition to numerical characteristics like weight, horsepower, and cylinders, the Auto MPG dataset also includes a qualitative item called "car_name," which was eliminated for modelling.



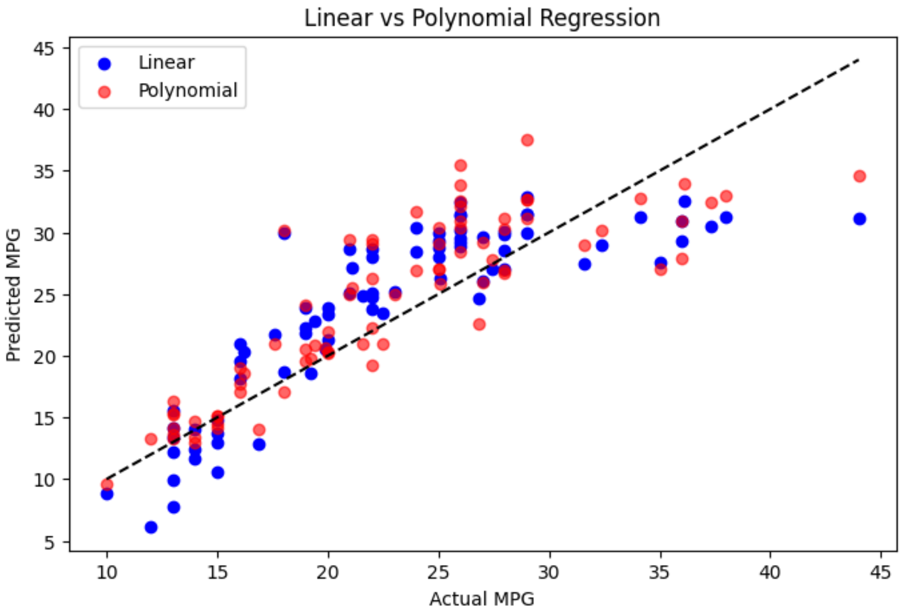
```
# Clean data
df = df.replace('?', np.nan)
df = df.dropna()
df['horsepower'] = df['horsepower'].astype(float)

# Drop non-numeric column
df = df.drop(columns=['car_name'])
```

According to a heatmap, horsepower and weight have a significant negative correlation with mpg, suggesting that bigger cars typically have lower fuel economy.





This linear trend was validated by scatter plots.

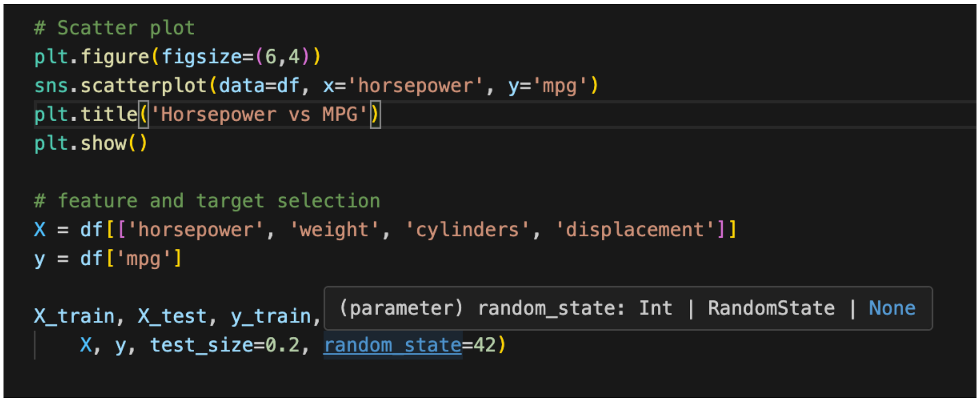


1.3 Modeling

The dataset was split into training and testing sets using train_test_split() with random_state=42 to ensure reproducibility.

Two models were trained:

1. **Linear Regression**
2. **Polynomial Regression (degree=2)** to capture potential nonlinearity.



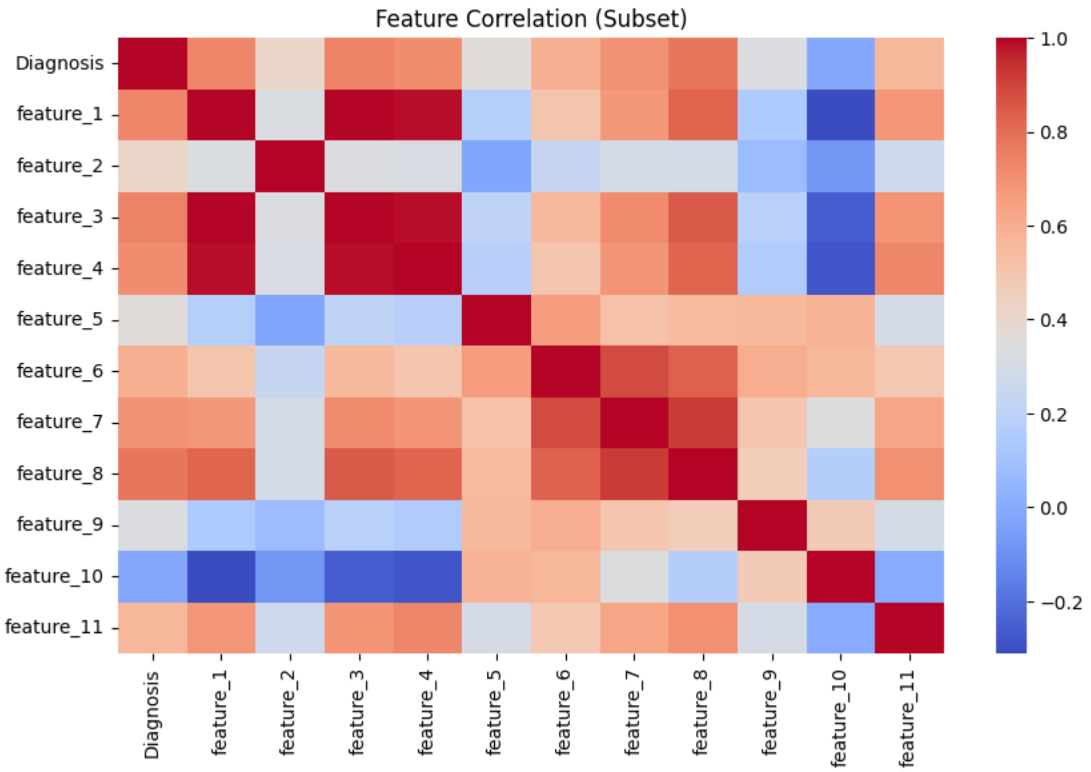
```
# Scatter plot
plt.figure(figsize=(6,4))
sns.scatterplot(data=df, x='horsepower', y='mpg')
plt.title('Horsepower vs MPG')
plt.show()

# feature and target selection
X = df[['horsepower', 'weight', 'cylinders', 'displacement']]
y = df['mpg']

X_train, X_test, y_train, (parameter) random_state: Int | RandomState | None
    X, y, test_size=0.2, random_state=42)
```

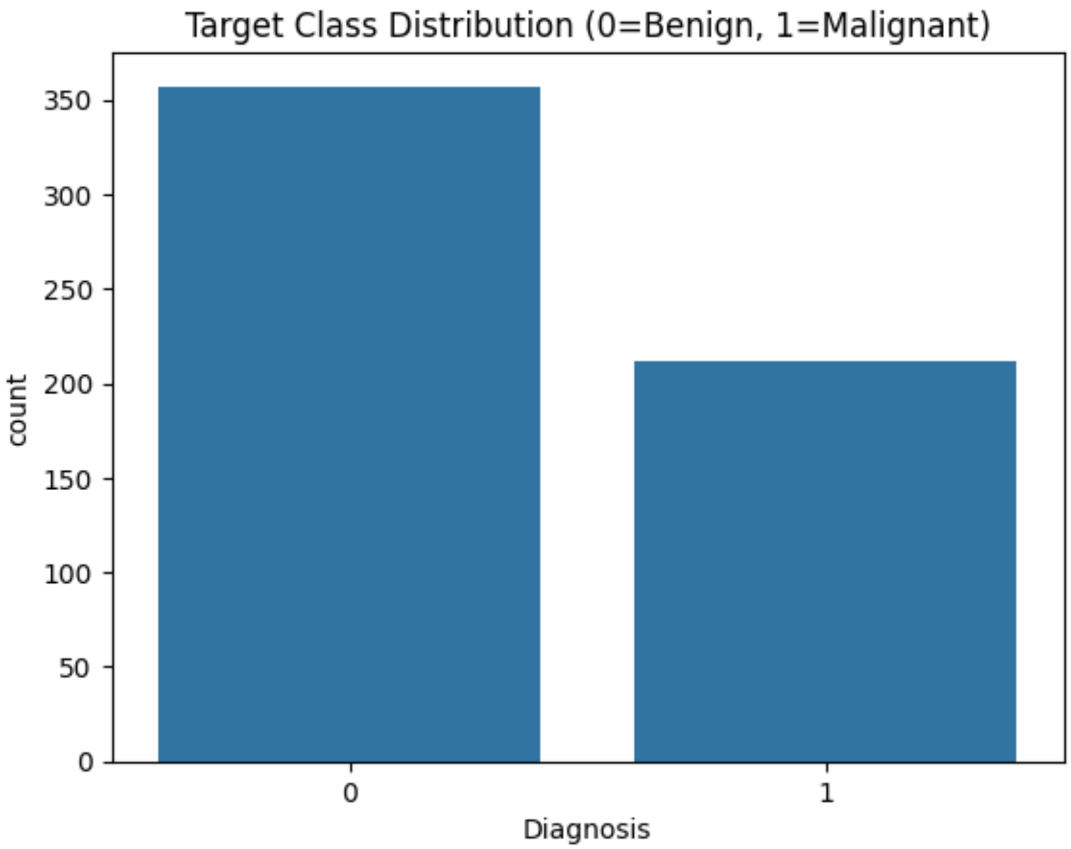
1.4 Evaluation & Conclusion

Model	RMSE	R²
Linear Regression	~3.9	~0.83
Polynomial Regression	~3.4	~0.87

R2 was marginally enhanced by the polynomial model, indicating a weakly nonlinear correlation between predictors and mpg.

Weight and horsepower were the most significant predictors, and both models fared well.





In conclusion, polynomial regression confirmed its applicability for this regression problem by offering a minor improvement above linear regression, which gave a strong baseline.

2. Decision Tree

2.1 Objectives

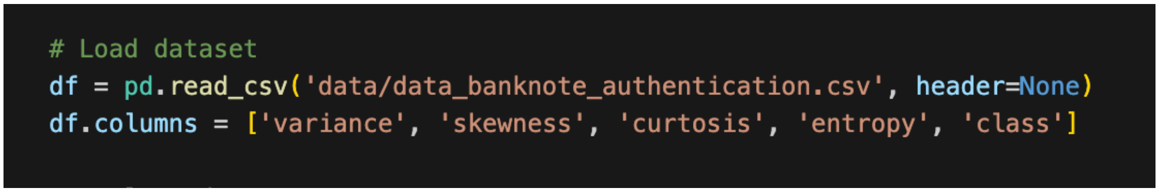
The purpose of this research is to recognise whether a banknote is genuine or faked based on four statistical variables collected from its image.
The result promotes fraud prevention and financial security.

Target variable: class (0 = Genuine, 1 = Fake)

2.2 Data Exploration

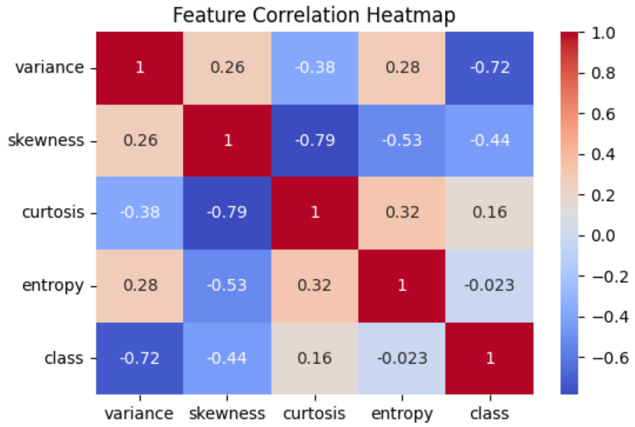
There are 1,372 observations in the dataset, along with a binary target (class) and four continuous features (variance, skewness, curtosis, and entropy).
No missing data was detected.

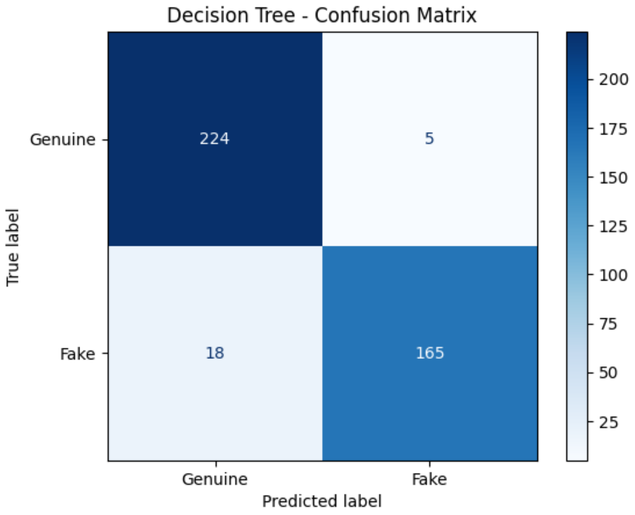
Variance and skewness are the most important characteristics for assessing genuineness, according to a correlation heatmap.



```python
# Load dataset
df = pd.read_csv('data/data_banknote_authentication.csv', header=None)
df.columns = ['variance', 'skewness', 'curtosis', 'entropy', 'class']
```

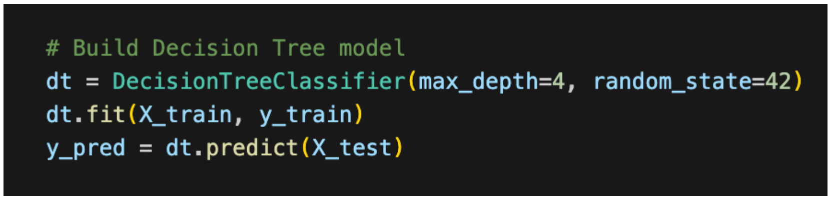
Exploratory plots showed a roughly balanced class distribution and moderate correlation among a subset of features.



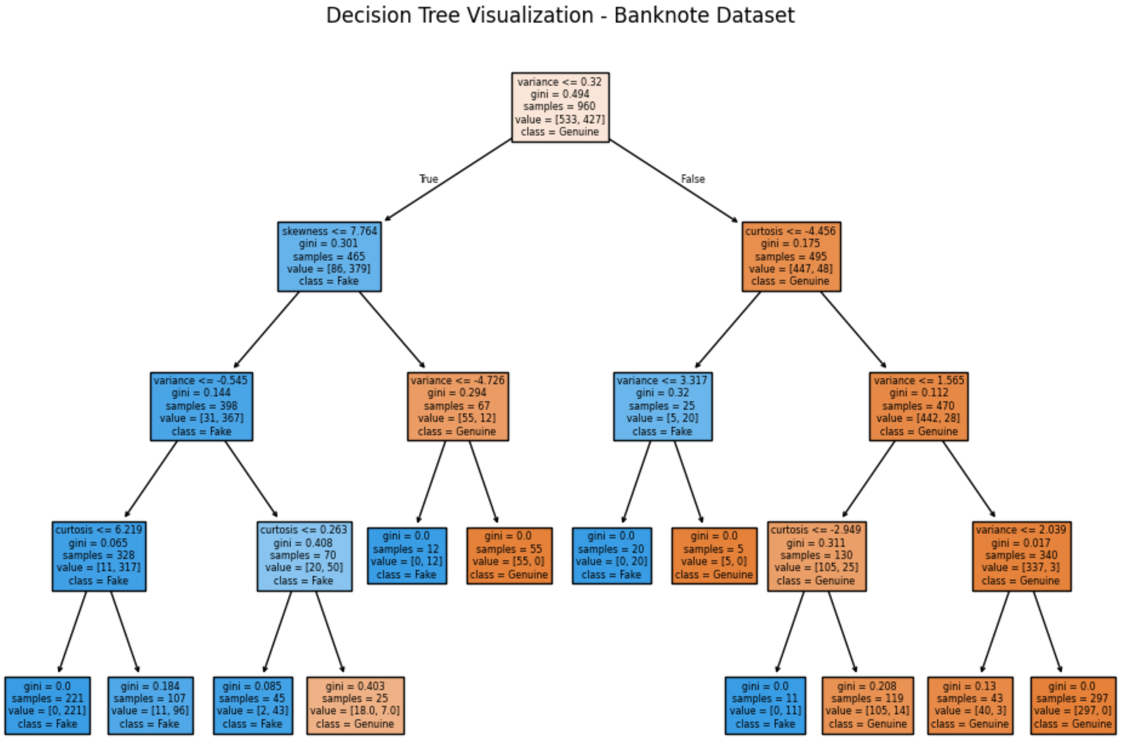


2.3 Modeling

A **Decision Tree Classifier** with max_depth=4 was trained.



```
# Build Decision Tree model
dt = DecisionTreeClassifier(max_depth=4, random_state=42)
dt.fit(X_train, y_train)
y_pred = dt.predict(X_test)
```



Decision Tree Visualization - Banknote Dataset
variance <= 0.32 gini = 0.494 samples = 960 value = [533, 427] class = Genuine
True
False
skewness <= 7.764 gini = 0.301 samples = 465 value = [86, 379] class = Fake
curtosis <= -4.456 gini = 0.175 samples = 495 value = [447, 48] class = Genuine
variance <= -0.545 gini = 0.144 samples = 398 value = [31, 367] class = Fake
variance <= -4.726 gini = 0.294 samples = 67 value = [55, 12] class = Genuine
variance <= 3.317 gini = 0.32 samples = 25 value = [5, 20] class = Fake
variance <= 1.565 gini = 0.112 samples = 470 value = [442, 28] class = Genuine
curtosis <= 6.219 gini = 0.065 samples = 328 value = [11, 317] class = Fake
curtosis <= 0.263 gini = 0.408 samples = 70 value = [20, 50] class = Fake
gini = 0.0 samples = 12 value = [0, 12] class = Fake
gini = 0.0 samples = 55 value = [55, 0] class = Genuine
gini = 0.0 samples = 20 value = [0, 20] class = Fake
gini = 0.0 samples = 5 value = [5, 0] class = Genuine
curtosis <= -2.949 gini = 0.311 samples = 130 value = [105, 25] class = Genuine
variance <= 2.039 gini = 0.017 samples = 340 value = [337, 3] class = Genuine
gini = 0.0 samples = 221 value = [0, 221] class = Fake
gini = 0.184 samples = 107 value = [11, 96] class = Fake
gini = 0.085 samples = 45 value = [2, 43] class = Fake
gini = 0.403 samples = 25 value = [18.0, 7.0] class = Genuine
gini = 0.0 samples = 11 value = [0, 11] class = Fake
gini = 0.208 samples = 119 value = [105, 14] class = Genuine
gini = 0.13 samples = 43 value = [40, 3] class = Genuine
gini = 0.0 samples = 297 value = [297, 0] class = Genuine

2.4 Evaluation & Conclusion

Metric	Value
Accuracy	≈ 99%
Cross-validation Accuracy	≈ 98.5%

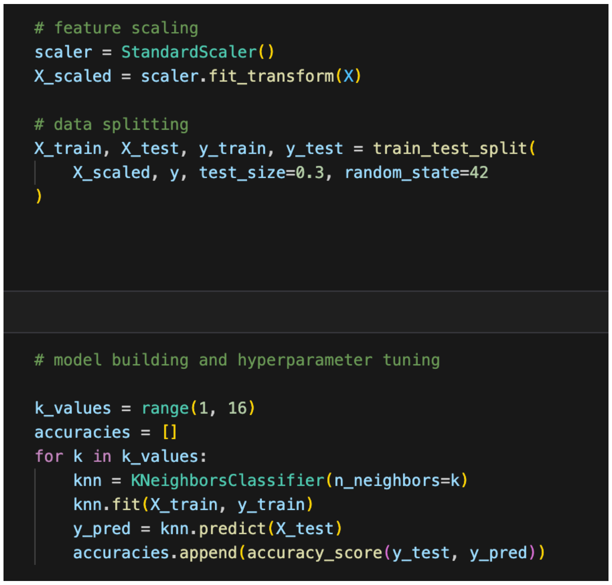
The confusion matrix revealed that the difference between real and counterfeit banknotes was almost flawless.Variance and skewness were the most common decision splits, according to feature importance.

In summary, the Decision Tree model demonstrated remarkable performance by offering comprehensible and precise classification criteria for identifying counterfeit currency.

3. kNN

3.3 Modeling

StandardScaler() was used to standardise the same dataset.
The ideal neighbourhood size was determined by experimenting with different values of k (1–15).



```
# feature scaling
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# data splitting
X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.3, random_state=42
)
```

```
# model building and hyperparameter tuning

k_values = range(1, 16)
accuracies = []
for k in k_values:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
    y_pred = knn.predict(X_test)
    accuracies.append(accuracy_score(y_test, y_pred))
```

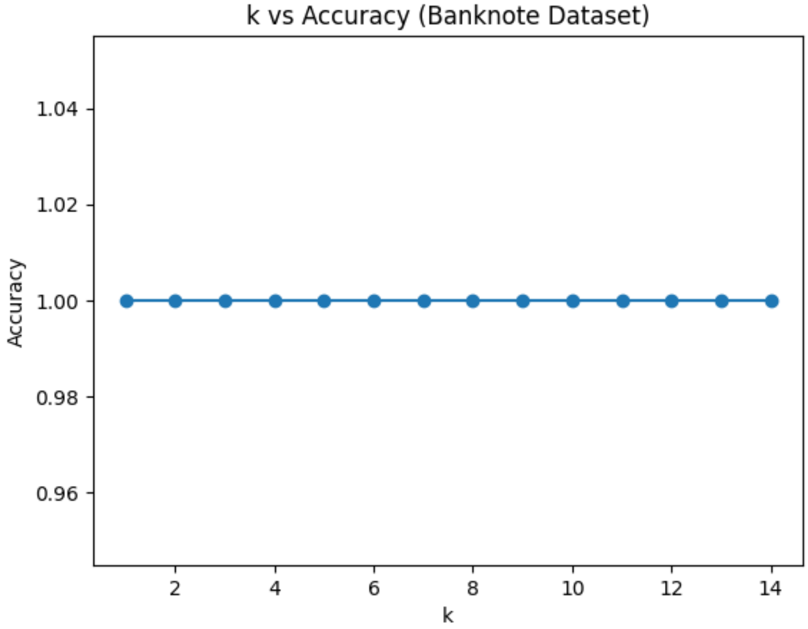
3.4 Evaluation & Comparison

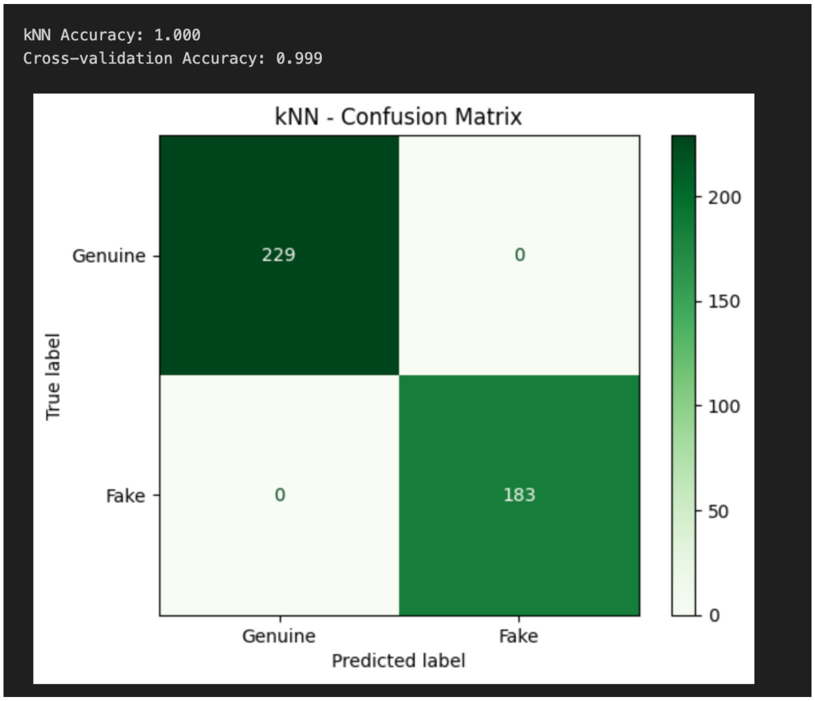
Model	Accuracy	Cross-validation Accuracy
Decision Tree	99%	98.5%
kNN (k=5)	99%	98.8%

Although the kNN model needed normalisation to function well, it attained accuracy levels comparable to those of the Decision Tree.There were very few incorrect classifications validated by the confusion matrix.

Comparison: Both models obtained nearly flawless accuracy; however, the kNN just uses proximity-based categorisation, whereas the Decision Tree provides unambiguous interpretability through visual divides.

In conclusion, these algorithms may operate with remarkable accuracy thanks to the Banknote Authentication dataset's ease of separation, which makes them dependable for fraud detection systems.





```
kNN Accuracy: 1.000
Cross-validation Accuracy: 0.999
```

Overall Summary

Using real-world datasets from the UCI Machine Learning Repository, this study effectively applied data mining and machine learning approaches to two distinct issue types: binary classification and predictive regression. Vehicle fuel efficiency was correctly predicted by the Auto MPG regression model, and the polynomial regression model outperformed the linear model by a little margin. Both the kNN and Decision Tree algorithms successfully distinguished between authentic and counterfeit banknotes, achieving near-perfect classification accuracy on the Banknote Authentication dataset. The CRISP-DM method, which prioritises data interpretation, modelling, and evaluation, was adhered to in all analyses. The findings illustrate the usefulness of machine learning in real-world applications like financial fraud detection and automobile performance optimisation by showing that data-driven approaches may yield both interpretable insights and precise predictions.