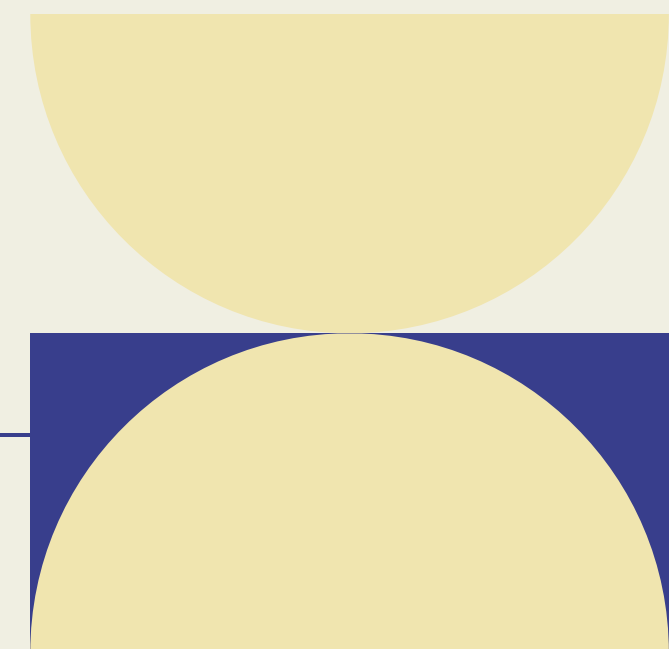


Simple Classification of Wine Dataset Using Decision Tree Method

Danica Alana Sjurjahady



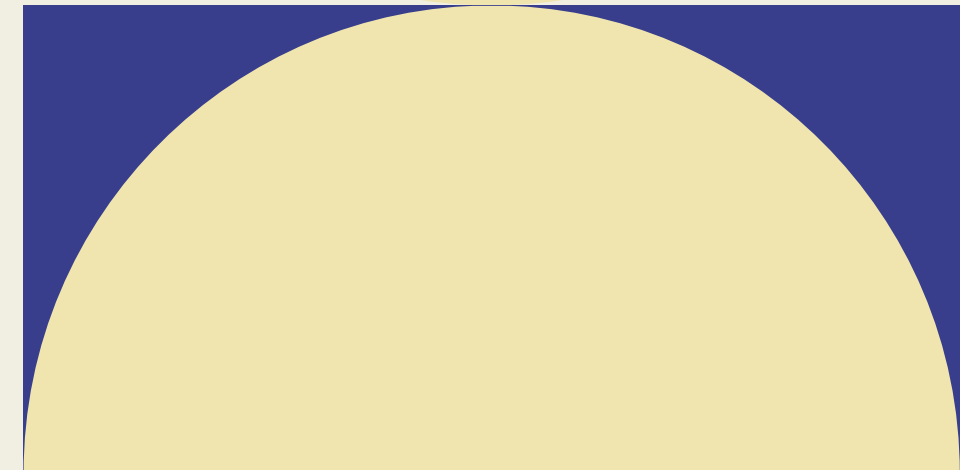
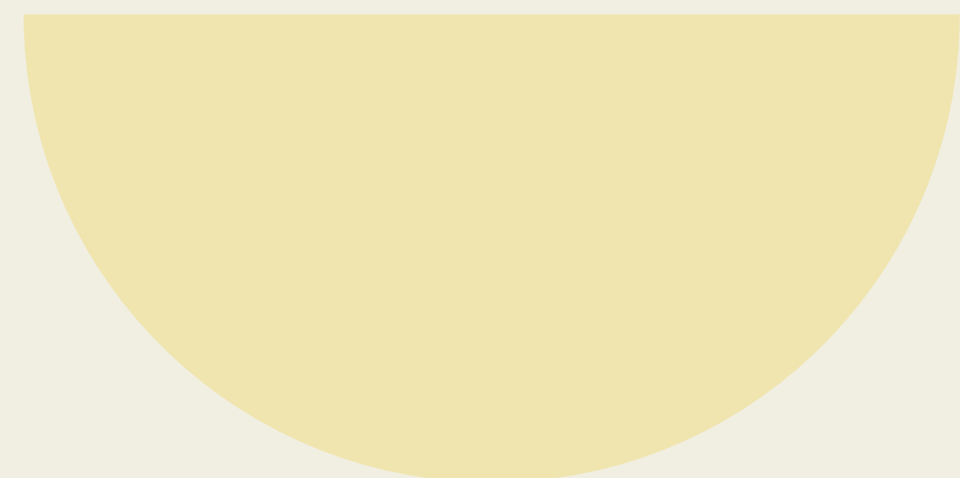
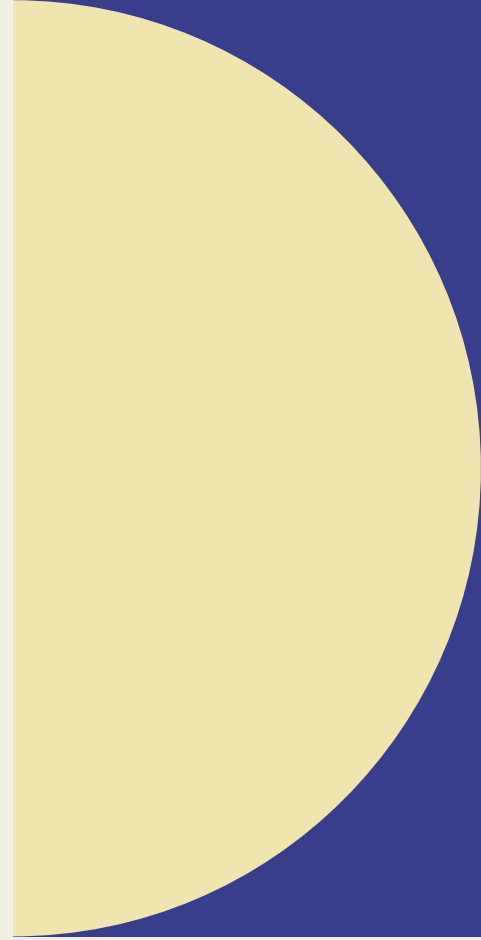
Fresh graduate majoring in Agro-Industrial Technology from University of Darussalam Gontor, who have a strong interest in data analysis. Have a good understanding of basic concepts of statistics and machine learning. Skilled in using MySQL, Python, Google Looker Studio, Google Colab, and Microsoft Power BI. Ready to learn and grow in the role of Data Analyst.

ABOUT ME



INTRODUCTION

This dataset is from UC Irvine Machine Learning Repository. The data is the results of a chemical analysis of wines grown in the same region in Italy by three different cultivators. There are thirteen different measurements taken for different constituents found in the three types of wine.



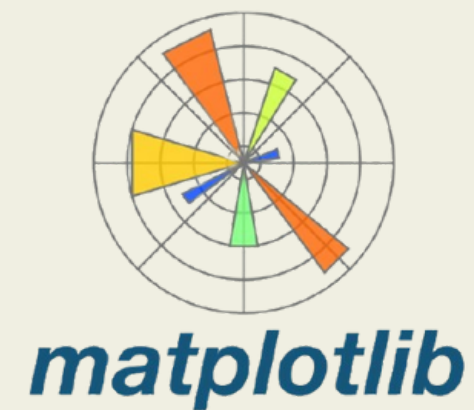
CHARACTERISTICS

Number of Instances: 178

Number of Attributes: 13 numeric, predictive attributes and the class

Attribute Information: Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, Proline, and class (class_0, class_1, class_2)

TOOLS USED



EXPLORATORY DATA ANALYSIS (EDA)

```
[12] import pandas as pd
      from sklearn import datasets

      # Load the Wine dataset from scikit-learn and convert it to a DataFrame
      wine = datasets.load_wine()

      x = wine.data    # inputs for machine learning
      y = wine.target  # desired output of machine learning

      # Convert feature and target data into a DataFrame
      df_x = pd.DataFrame(x, columns = wine.feature_names)
      df_y = pd.Series(y, name = 'target')

      # Combine features and targets in one DataFrames
      df = pd.concat([df_x, df_y], axis = 1)

      df.head(10)
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	target
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065.0	0
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050.0	0
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185.0	0
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480.0	0
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735.0	0
5	14.20	1.76	2.45	15.2	112.0	3.27	3.39	0.34	1.97	6.75	1.05	2.85	1450.0	0
6	14.39	1.87	2.45	14.6	96.0	2.50	2.52	0.30	1.98	5.25	1.02	3.58	1290.0	0
7	14.06	2.15	2.61	17.6	121.0	2.60	2.51	0.31	1.25	5.05	1.06	3.58	1295.0	0
8	14.83	1.64	2.17	14.0	97.0	2.80	2.98	0.29	1.98	5.20	1.08	2.85	1045.0	0
9	13.86	1.35	2.27	16.0	98.0	2.98	3.15	0.22	1.85	7.22	1.01	3.55	1045.0	0

EXPLORATORY DATA ANALYSIS (EDA)

```
[3] # View basic information about the data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 14 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   alcohol                               178 non-null    float64
1   malic_acid                           178 non-null    float64
2   ash                                   178 non-null    float64
3   alcalinity_of_ash                    178 non-null    float64
4   magnesium                            178 non-null    float64
5   total_phenols                        178 non-null    float64
6   flavanoids                           178 non-null    float64
7   nonflavanoid_phenols                 178 non-null    float64
8   proanthocyanins                      178 non-null    float64
9   color_intensity                      178 non-null    float64
10  hue                                   178 non-null    float64
11  od280/od315_of_diluted_wines         178 non-null    float64
12  proline                               178 non-null    float64
13  target                                178 non-null    int64
dtypes: float64(13), int64(1)
memory usage: 19.6 KB
```

EXPLORATORY DATA ANALYSIS (EDA)

```
[4] # Identifies all the different numbers that appear in the 'target' column
df['target'].unique()
```

```
➞ array([0, 1, 2])
```

```
[5] # View a statistical description of the data
df.describe()
```

```
➞
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity
count	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000
mean	13.000618	2.336348	2.366517	19.494944	99.741573	2.295112	2.029270	0.361854	1.590899	5.058000
std	0.811827	1.117146	0.274344	3.339564	14.282484	0.625851	0.998859	0.124453	0.572359	2.318200
min	11.030000	0.740000	1.360000	10.600000	70.000000	0.980000	0.340000	0.130000	0.410000	1.280000
25%	12.362500	1.602500	2.210000	17.200000	88.000000	1.742500	1.205000	0.270000	1.250000	3.220000
50%	13.050000	1.865000	2.360000	19.500000	98.000000	2.355000	2.135000	0.340000	1.555000	4.690000
75%	13.677500	3.082500	2.557500	21.500000	107.000000	2.800000	2.875000	0.437500	1.950000	6.200000
max	14.830000	5.800000	3.230000	30.000000	162.000000	3.880000	5.080000	0.660000	3.580000	13.000000

DATA MODELLING

```
[17] from sklearn.model_selection import train_test_split

# Split the data into train and test
x_train, x_test, y_train, y_test = train_test_split(df_x, df_y, test_size = 0.2, random_state = 42)
```

```
[18] from sklearn.tree import DecisionTreeClassifier

# Create and train a Decision Tree model
model = DecisionTreeClassifier(random_state = 42)
model.fit(x_train, y_train)
```



▼ DecisionTreeClassifier ⓘ ?

DecisionTreeClassifier(random_state=42)

```
[23] from sklearn.metrics import accuracy_score

# Predict and evaluate the model
y_pred = model.predict(x_test)

accuracy = accuracy_score(y_test, y_pred)

print("Classification Report:")
print(f"Accuracy: {accuracy * 100:.2f}%")
```

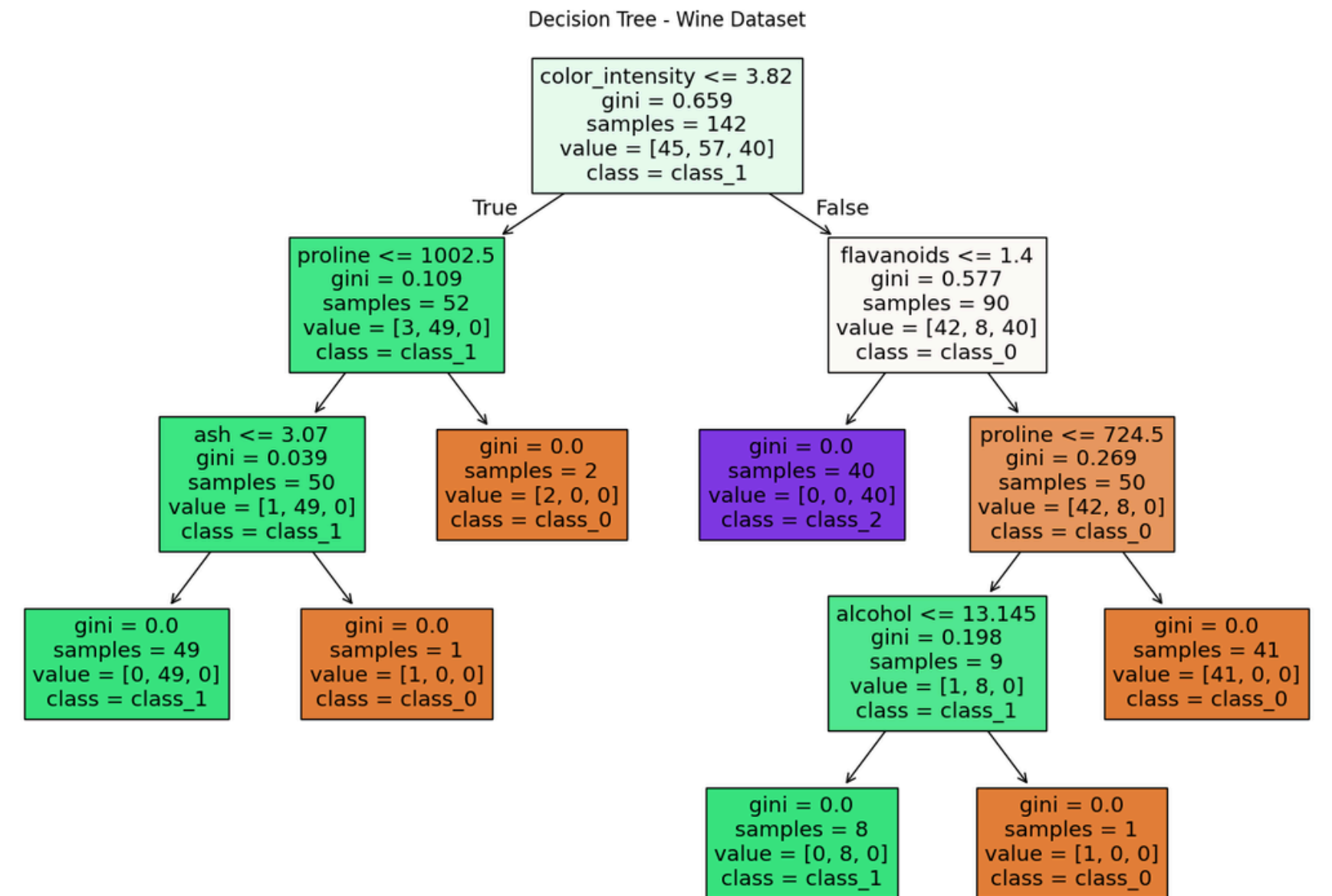


Classification Report:
Accuracy: 94.44%

DATA VISUALIZATION

```
[31] import matplotlib.pyplot as plt
      from sklearn import tree

      # Visualisasi Decision Tree
      plt.figure(figsize = (15,10))
      tree.plot_tree(model,
                     feature_names=wine.feature_names,
                     class_names=wine.target_names,
                     filled=True)
      plt.title("Decision Tree - Wine Dataset")
      plt.show()
```



If you have any questions, suggestions or feedbacks,
please do not hesitate to reach me through the
contacts below:



rsgame99@gmail.com



github.com/DanicaAlana



[linkedin.com/in/danica-alana-
sjurjahady-85b124211/](https://www.linkedin.com/in/danica-alana-sjurjahady-85b124211/)

THANK YOU