ibimbing

“Harnessing the power of Machine Learning : Transforming Data into Actionable insights”

DSF 35.0-DATA SCIENCE

TOOLS

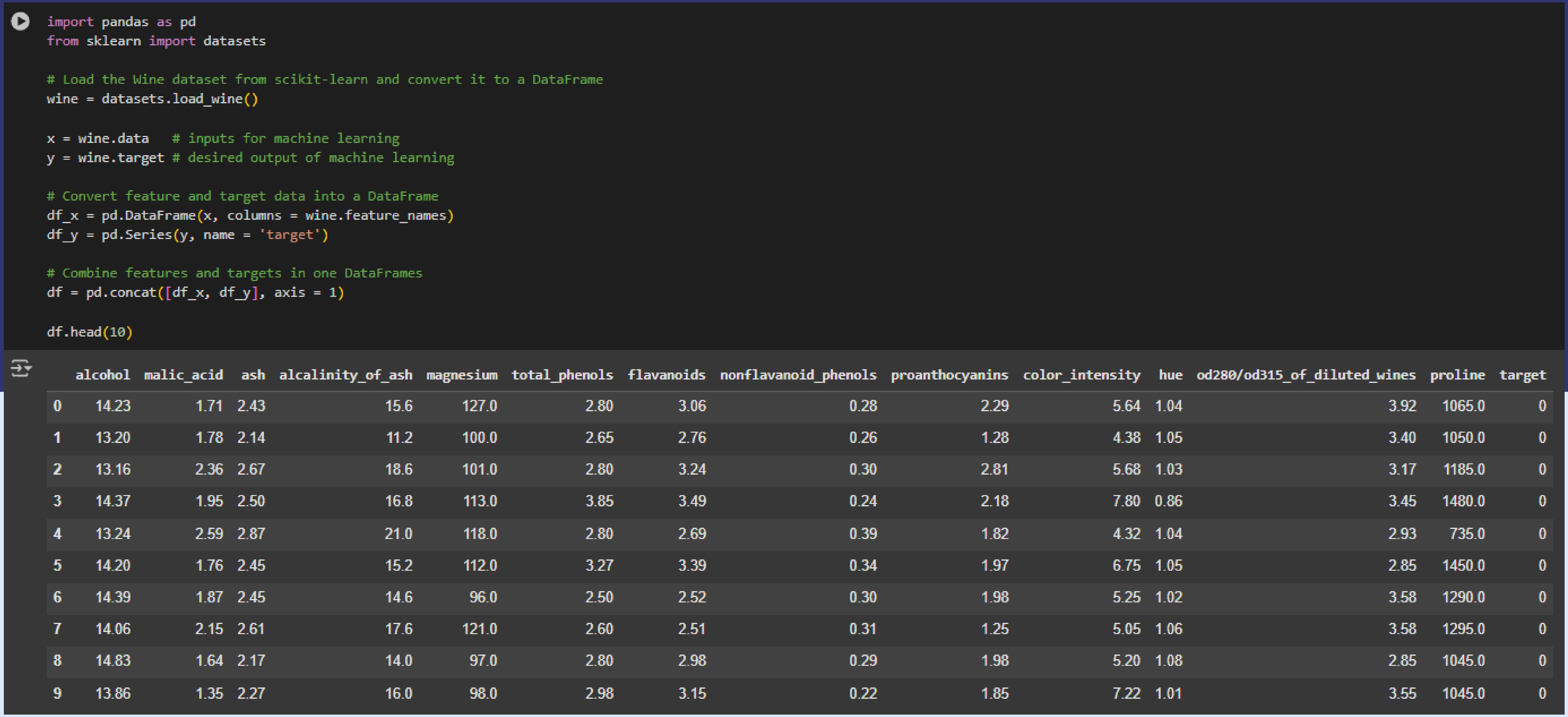
scikit learn

matplotlib

pandas

seaborn

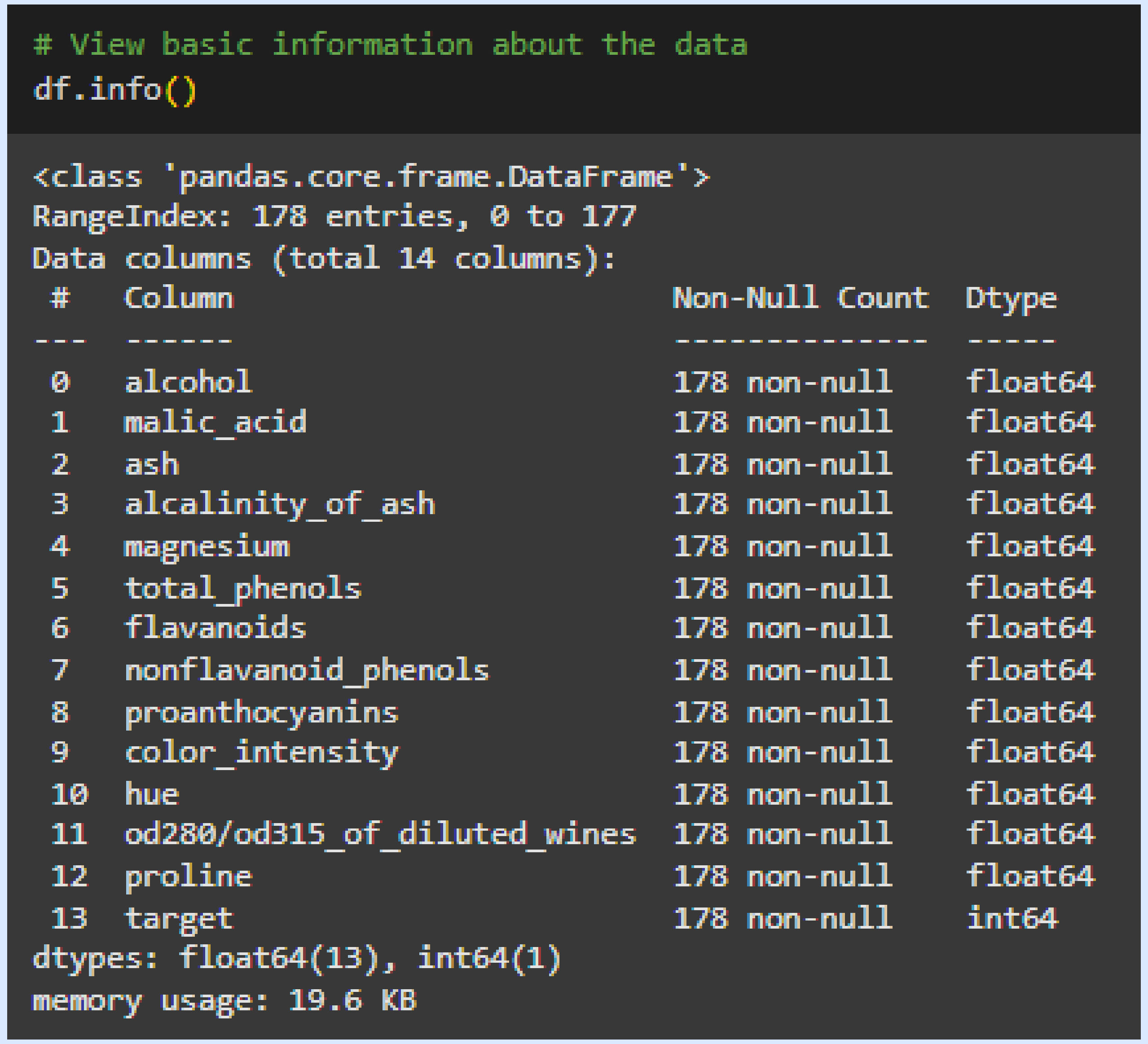
Input data



```python
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)
```

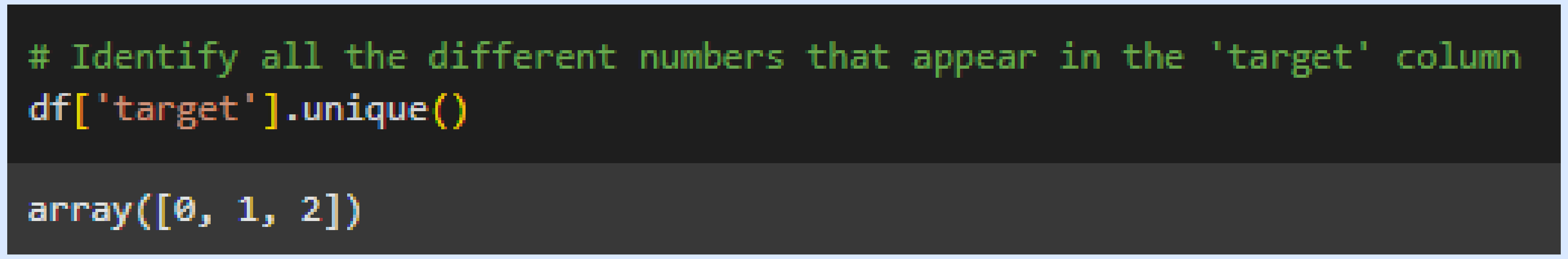
	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	target
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065.0	0
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38	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

Explotatory Data Analysis (EDA)



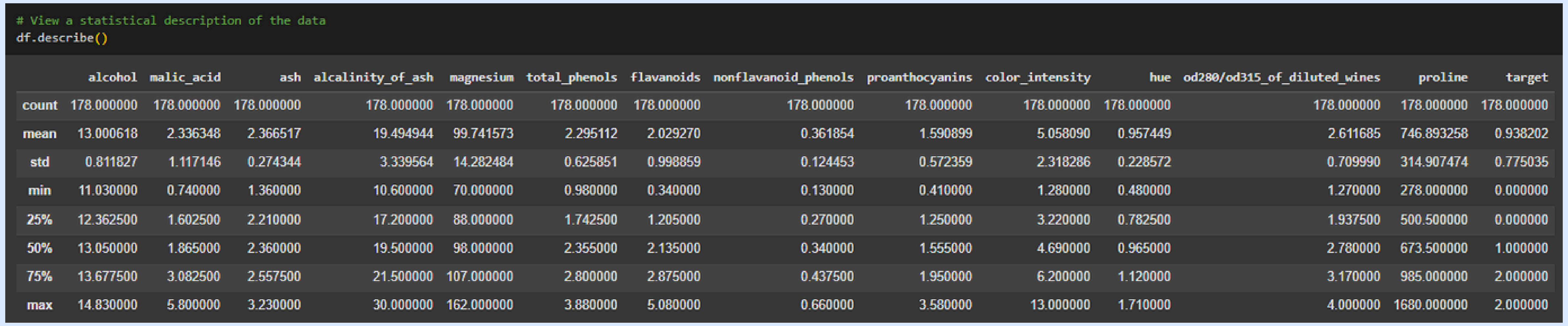
```
# View basic information about the data
df.info()
```

```
<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
```



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

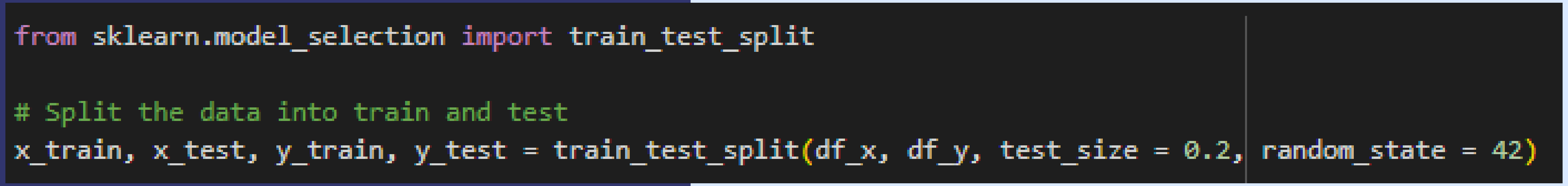
```
array([0, 1, 2])
```



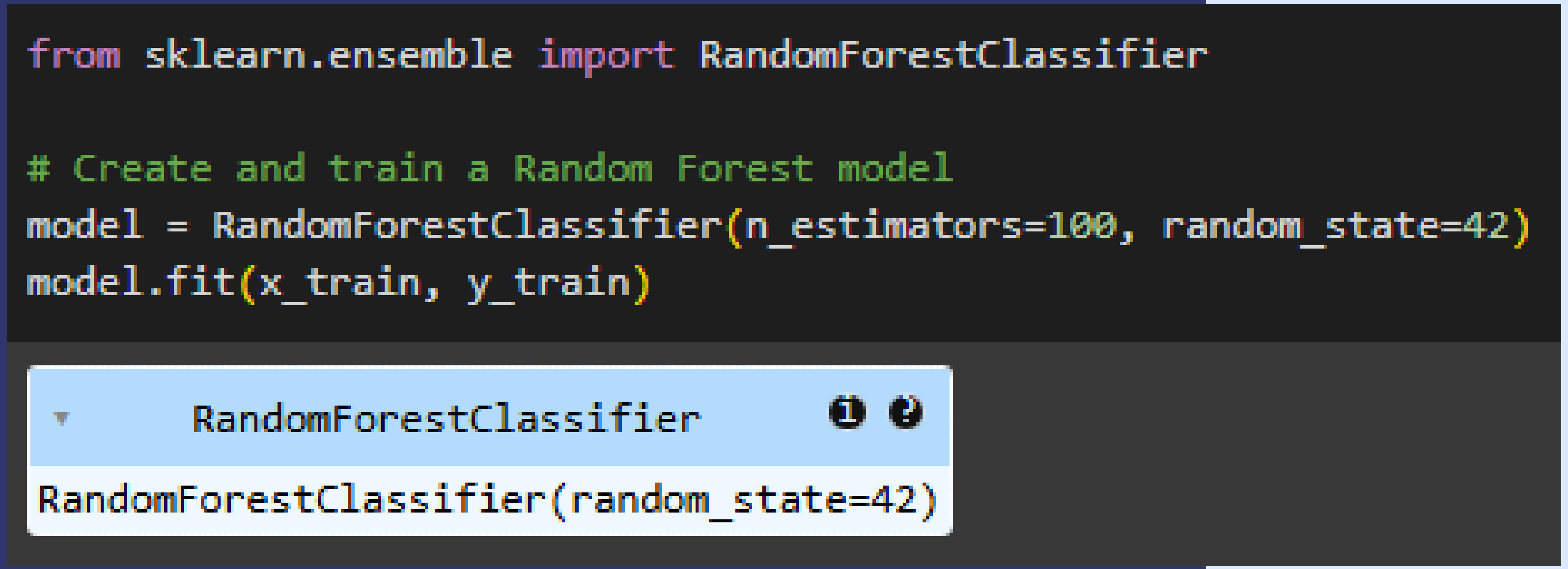
```
# View a statistical description of the data
df.describe()
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	target
count	178.000000	178.000000	178.000000	178.000000	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	5.058090	0.957449	2.611685	746.893258	0.938202
std	0.811827	1.117146	0.274344	3.339564	14.282484	0.625851	0.998859	0.124453	0.572359	2.318286	0.228572	0.709990	314.907474	0.775035
min	11.030000	0.740000	1.360000	10.600000	70.000000	0.980000	0.340000	0.130000	0.410000	1.280000	0.480000	1.270000	278.000000	0.000000
25%	12.362500	1.602500	2.210000	17.200000	88.000000	1.742500	1.205000	0.270000	1.250000	3.220000	0.782500	1.937500	500.500000	0.000000
50%	13.050000	1.865000	2.360000	19.500000	98.000000	2.355000	2.135000	0.340000	1.555000	4.690000	0.965000	2.780000	673.500000	1.000000
75%	13.677500	3.082500	2.557500	21.500000	107.000000	2.800000	2.875000	0.437500	1.950000	6.200000	1.120000	3.170000	985.000000	2.000000
max	14.830000	5.800000	3.230000	30.000000	162.000000	3.880000	5.080000	0.660000	3.580000	13.000000	1.710000	4.000000	1680.000000	2.000000

Data Modeling

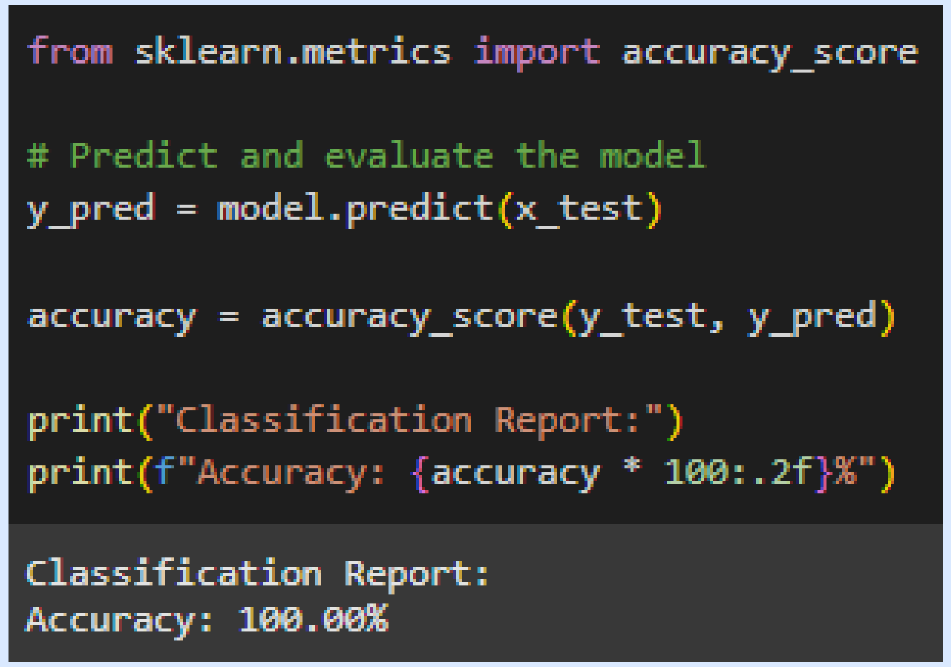


```python
from sklearn.model_selection import train_test_split

# 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)
```



```python
from sklearn.ensemble import RandomForestClassifier

# Create and train a Random Forest model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(x_train, y_train)
```

```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

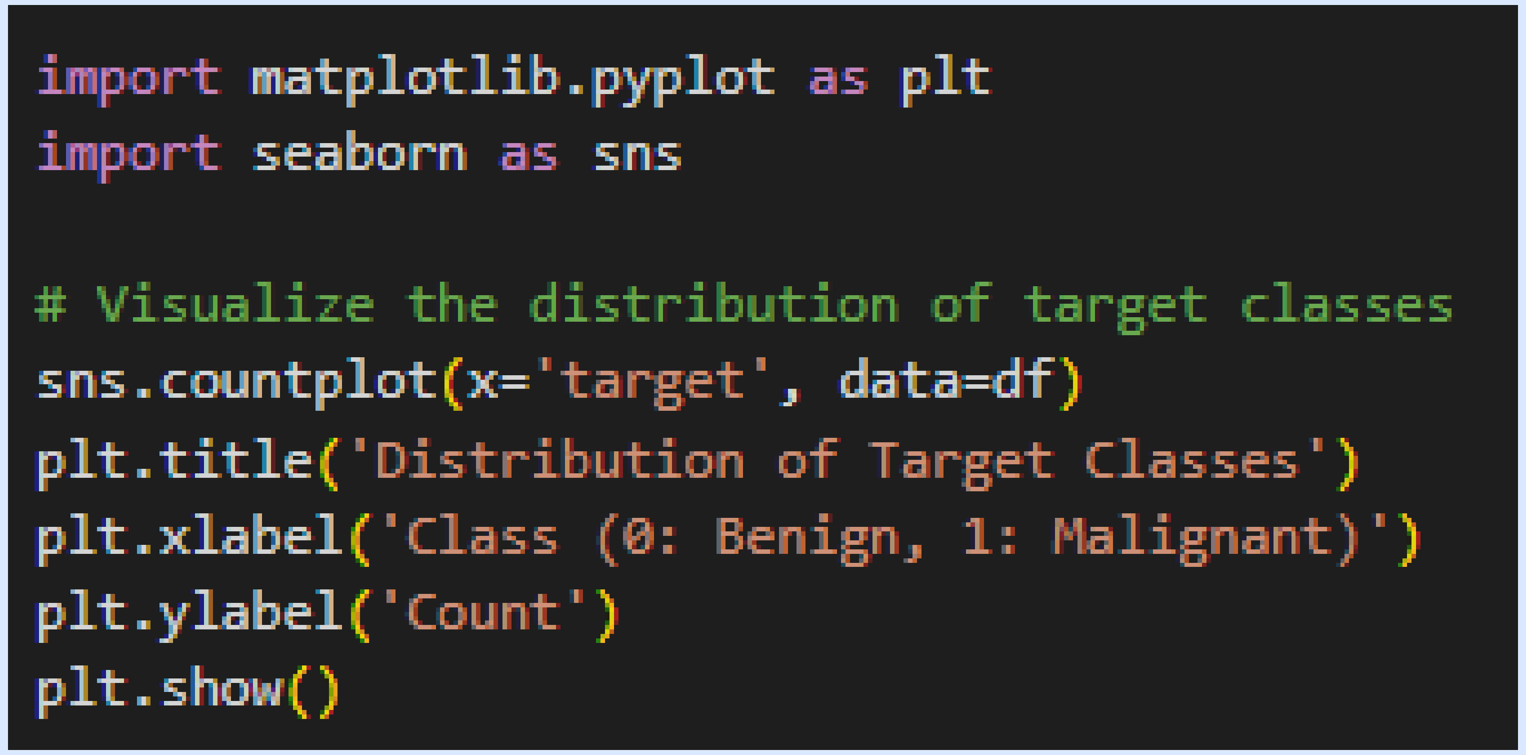


```python
from sklearn.metrics import accuracy_score

# 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}%")
```

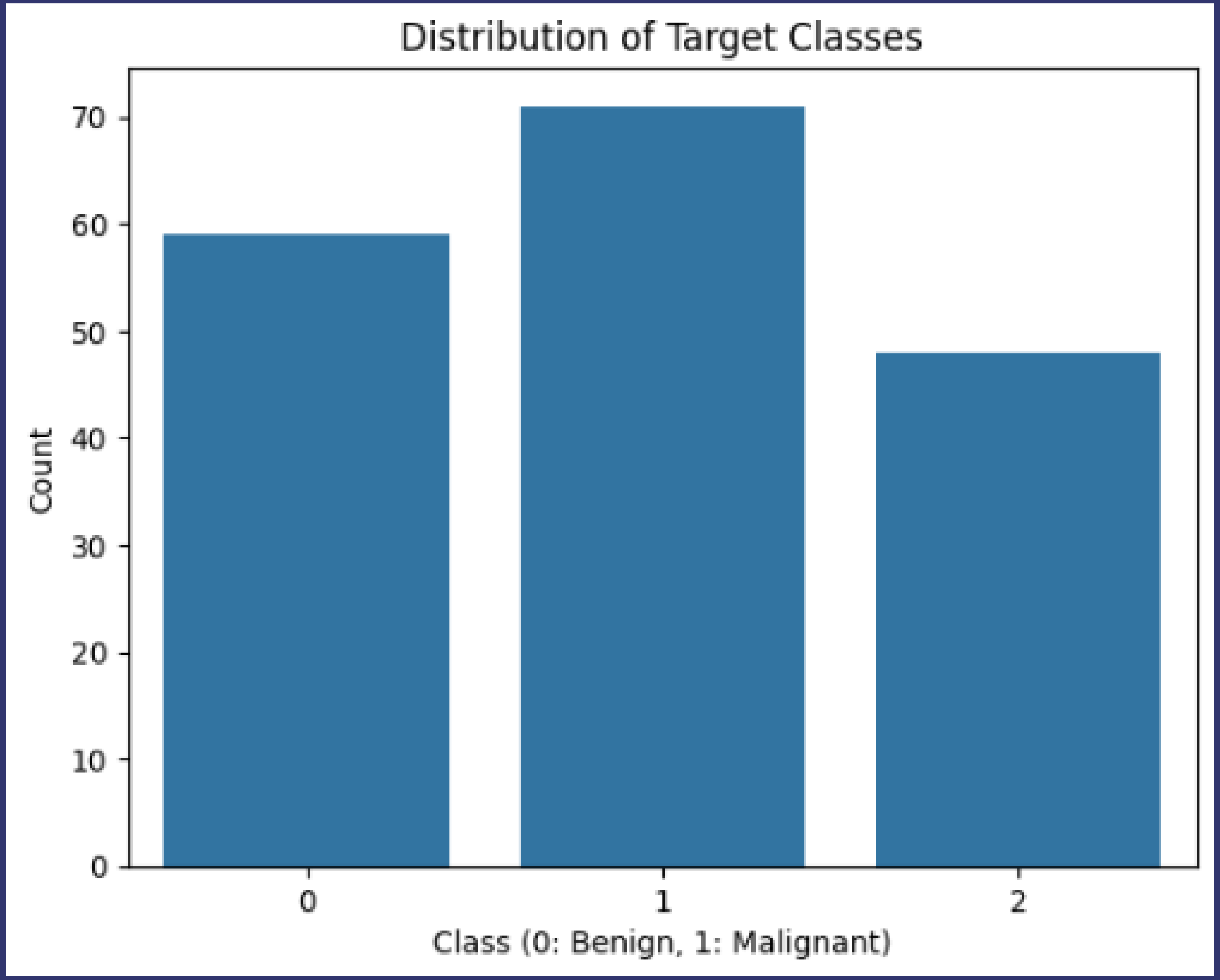
```
Classification Report:
Accuracy: 100.00%
```

Data Visualization

Distribution of target classes

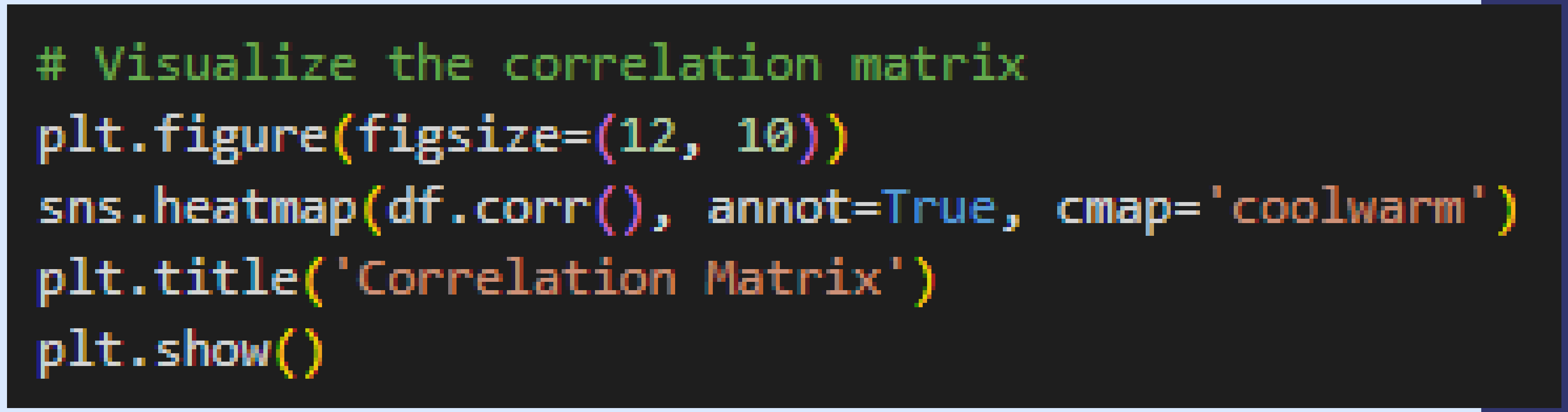


```
import matplotlib.pyplot as plt
import seaborn as sns

# Visualize the distribution of target classes
sns.countplot(x='target', data=df)
plt.title('Distribution of Target Classes')
plt.xlabel('Class (0: Benign, 1: Malignant)')
plt.ylabel('Count')
plt.show()
```

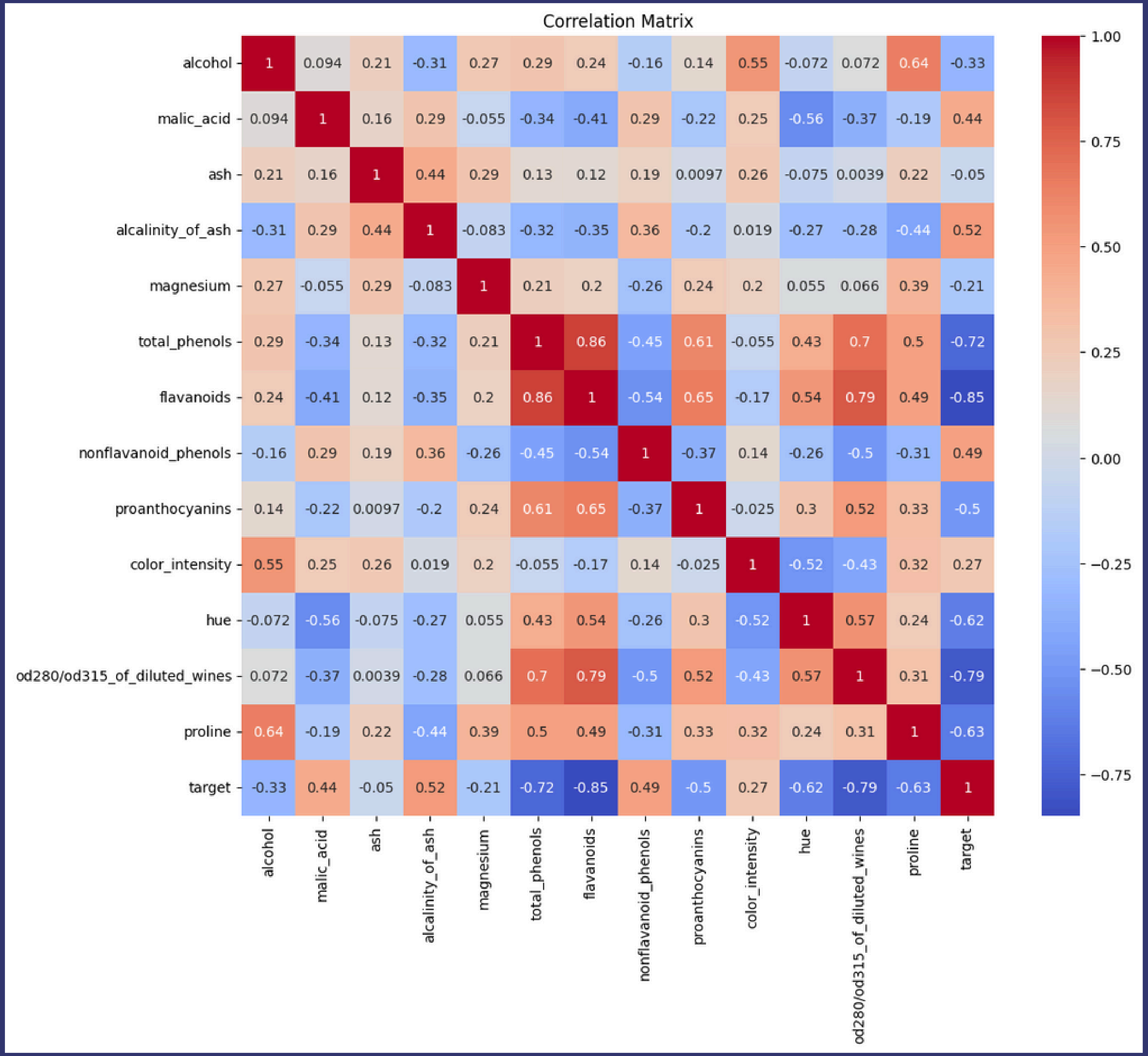


Data Visualization

Colerrelation Matrix



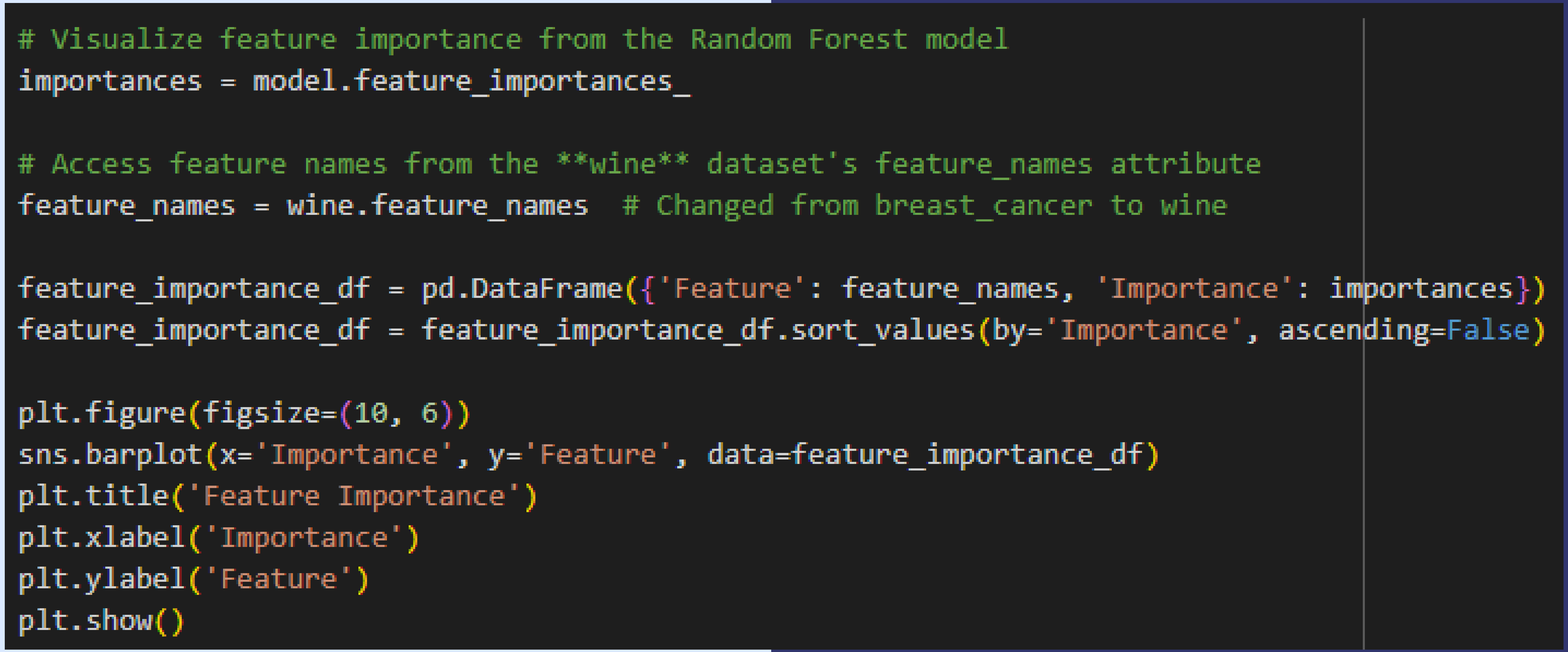
```
# Visualize the correlation matrix
plt.figure(figsize=(12, 10))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```



	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	target
alcohol	1	0.094	0.21	-0.31	0.27	0.29	0.24	-0.16	0.14	0.55	-0.072	0.072	0.64	-0.33
malic_acid	0.094	1	0.16	0.29	-0.055	-0.34	-0.41	0.29	-0.22	0.25	-0.56	-0.37	-0.19	0.44
ash	0.21	0.16	1	0.44	0.29	0.13	0.12	0.19	0.0097	0.26	-0.075	0.0039	0.22	-0.05
alcalinity_of_ash	-0.31	0.29	0.44	1	-0.083	-0.32	-0.35	0.36	-0.2	0.019	-0.27	-0.28	-0.44	0.52
magnesium	0.27	-0.055	0.29	-0.083	1	0.21	0.2	-0.26	0.24	0.2	0.055	0.066	0.39	-0.21
total_phenols	0.29	-0.34	0.13	-0.32	0.21	1	0.86	-0.45	0.61	-0.055	0.43	0.7	0.5	-0.72
flavanoids	0.24	-0.41	0.12	-0.35	0.2	0.86	1	-0.54	0.65	-0.17	0.54	0.79	0.49	-0.85
nonflavanoid_phenols	-0.16	0.29	0.19	0.36	-0.26	-0.45	-0.54	1	-0.37	0.14	-0.26	-0.5	-0.31	0.49
proanthocyanins	0.14	-0.22	0.0097	-0.2	0.24	0.61	0.65	-0.37	1	-0.025	0.3	0.52	0.33	-0.5
color_intensity	0.55	0.25	0.26	0.019	0.2	-0.055	-0.17	0.14	-0.025	1	-0.52	-0.43	0.32	0.27
hue	-0.072	-0.56	-0.075	-0.27	0.055	0.43	0.54	-0.26	0.3	-0.52	1	0.57	0.24	-0.62
od280/od315_of_diluted_wines	0.072	-0.37	0.0039	-0.28	0.066	0.7	0.79	-0.5	0.52	-0.43	0.57	1	0.31	-0.79
proline	0.64	-0.19	0.22	-0.44	0.39	0.5	0.49	-0.31	0.33	0.32	0.24	0.31	1	-0.63
target	-0.33	0.44	-0.05	0.52	-0.21	-0.72	-0.85	0.49	-0.5	0.27	-0.62	-0.79	-0.63	1

Data Visualization

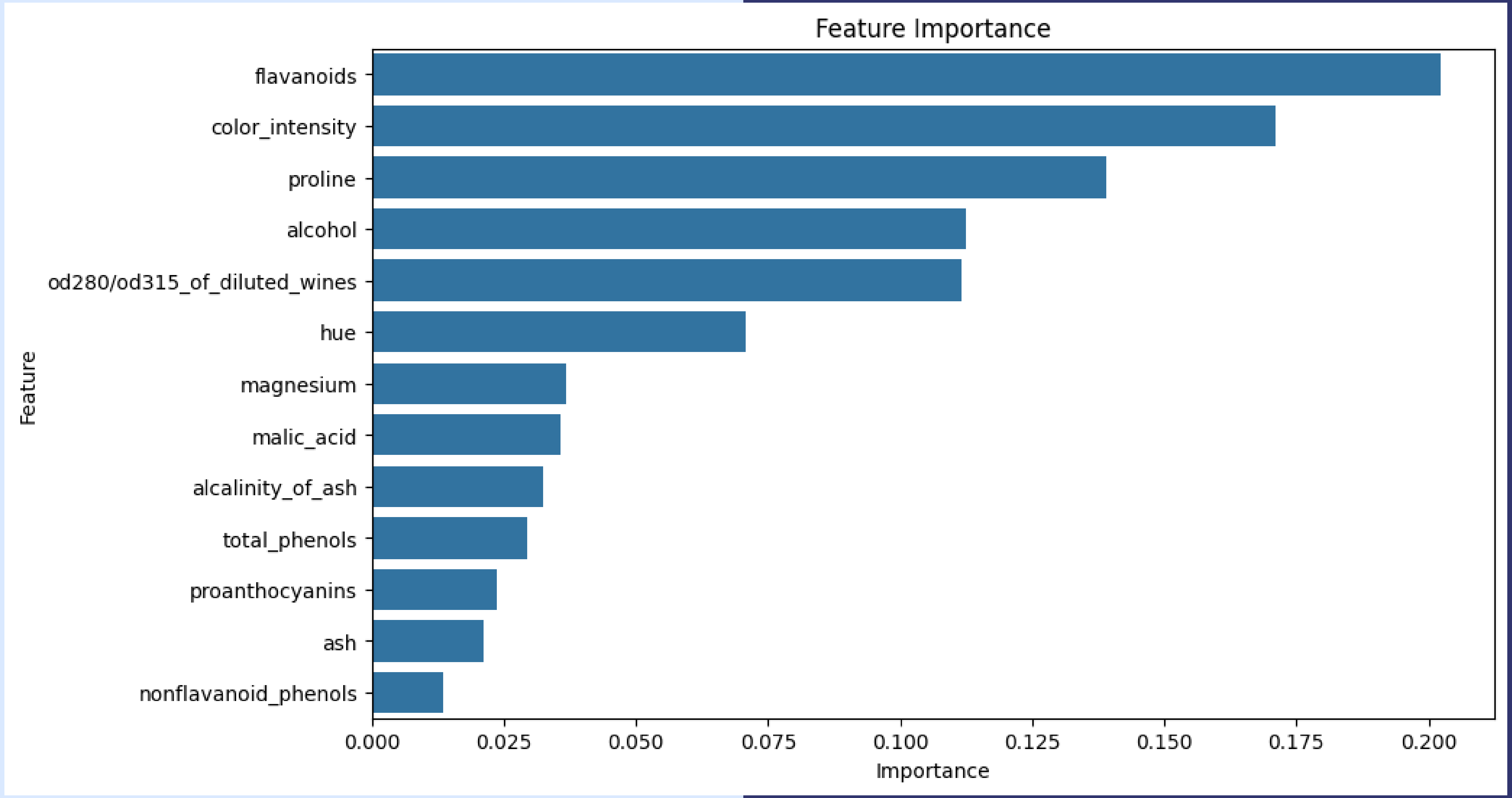
Feature Importance



```python
# Visualize feature importance from the Random Forest model
importances = model.feature_importances_

# Access feature names from the **wine** dataset's feature_names attribute
feature_names = wine.feature_names  # Changed from breast_cancer to wine

feature_importance_df = pd.DataFrame({'Feature': feature_names, 'Importance': importances})
feature_importance_df = feature_importance_df.sort_values(by='Importance', ascending=False)

plt.figure(figsize=(10, 6))
sns.barplot(x='Importance', y='Feature', data=feature_importance_df)
plt.title('Feature Importance')
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.show()
```

Data Visualization

Feature Importance



THANK YOU FOR

Your Attention

muhammadrasyas.123@gmail.com

StickyHoong

Muhammad Rasya Syarifuddin