In [ ]:
```
import numpy as np
import pandas as pd
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
import sklearn.ensemble as ske
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
df=pd.read_csv("winequality-white.csv",sep=";")
```

In [ ]:
```
reg = ske.RandomForestRegressor()
```

In [ ]:
```
X = df.drop('quality', axis=1)
y = df['quality']
```

In [ ]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [ ]:
```
reg.fit(X_train, y_train)
```

Out[ ]:
```
RandomForestRegressor
RandomForestRegressor()
```

In [ ]:
```
y_pred = reg.predict(X_test)
y_pred
```

Out[ ]:
```
array([6.91, 7.44, 6.72, 5.11, 6.82, 6.17, 5.26, 5.14, 5.84, 5.13, 6.78,
       5.24, 6.83, 5.63, 6.78, 5.1 , 7.32, 5.48, 6.76, 5.64, 5.34, 5.85,
       5.15, 6.13, 6.35, 5.2 , 5.02, 6.04, 6.76, 5.63, 5.05, 5.07, 5.6 ,
       5.87, 5.48, 6.67, 6.37, 5.35, 5.2 , 5.71, 5.26, 6.  , 6.11, 5.67,
       4.73, 5.89, 5.68, 5.17, 5.11, 5.04, 4.99, 6.08, 5.57, 6.15, 6.45,
       5.59, 5.87, 6.11, 5.96, 6.97, 6.03, 6.22, 5.65, 6.88, 5.78, 6.67,
       6.06, 5.42, 6.  , 6.  , 5.25, 5.92, 5.32, 4.49, 5.92, 6.18, 5.89,
       6.1 , 7.09, 6.87, 5.61, 5.59, 5.97, 6.37, 5.18, 6.88, 5.09, 6.55,
       5.29, 5.88, 6.95, 5.08, 5.62, 6.63, 6.33, 6.53, 5.99, 5.89, 5.15,
       6.18, 5.28, 6.58, 6.81, 5.66, 6.  , 6.16, 6.97, 6.46, 5.48, 6.13,
       5.98, 6.65, 6.11, 5.33, 5.31, 6.8 , 6.78, 6.27, 5.3 , 8.  , 5.6 ,
       6.73, 6.12, 5.68, 6.86, 5.74, 7.04, 5.77, 6.42, 5.27, 6.19, 6.17,
       5.68, 6.37, 5.06, 6.12, 6.01, 5.83, 6.59, 5.27, 6.62, 5.99, 6.33,
       5.53, 6.65, 5.09, 5.05, 5.44, 5.4 , 5.25, 6.32, 5.88, 5.7 , 5.7 ,
       6.86, 4.98, 6.48, 5.97, 6.17, 5.51, 5.69, 5.29, 5.63, 5.81, 6.86,
       6.48, 5.  , 6.01, 5.26, 6.23, 5.46, 6.  , 6.29, 5.33, 7.23, 6.1 ,
       5.65, 6.16, 6.45, 5.34, 6.29, 5.76, 5.95, 6.15, 5.5 , 6.64, 6.32,
       5.47, 6.01, 5.41, 5.69, 5.64, 6.66, 6.22, 5.38, 5.47, 7.03, 6.75,
       5.28, 5.73, 6.61, 4.68, 6.34, 5.84, 5.51, 5.88, 6.68, 6.66, 5.79,
       6.45, 5.85, 6.62, 6.21, 7.1 , 5.05, 5.31, 6.02, 5.57, 6.22, 6.3 ,
       5.02, 5.46, 5.77, 6.09, 6.04, 5.23, 6.14, 6.62, 6.61, 5.46, 5.85,
       5.07, 6.25, 5.58, 5.31, 6.91, 6.04, 5.85, 5.68, 6.3 , 6.06, 4.7 ,
       6.56, 5.53, 6.5 , 4.91, 4.79, 5.14, 5.86, 5.73, 5.94, 4.64, 5.83,
       5.07, 5.84, 5.44, 6.14, 5.07, 6.08, 5.85, 5.94, 5.44, 5.53, 4.98,
       6.9 , 6.08, 5.71, 5.82, 6.45, 5.48, 5.19, 5.76, 6.54, 7.21, 5.65,
       6.22, 5.25, 6.06, 6.32, 5.38, 5.87, 5.92, 6.34, 4.86, 6.08, 7.07,
       7.35, 5.32, 5.64, 5.43, 5.9 , 5.83, 6.22, 5.32, 7.77, 5.52, 5.81,
       6.19, 5.5 , 5.18, 4.99, 5.1 , 5.9 , 5.38, 6.53, 5.27, 5.41, 5.84,
       6.37, 4.61, 6.09, 5.91, 5.33, 5.99, 6.33, 7.1 , 5.88, 6.33, 4.88,
       6.88, 6.76, 6.92, 6.36, 6.16, 5.15, 5.21, 6.65, 5.47, 7.34, 5.06,
       5.97, 6.17, 6.01, 6.06, 5.23, 5.6 , 6.87, 6.43, 7.41, 5.5 , 6.26,
       5.6 , 5.14, 6.16, 5.46, 5.27, 6.31, 4.44, 7.  , 6.08, 6.92, 6.69,
       6.  , 5.88, 6.2 , 5.07, 5.58, 6.97, 6.81, 5.14, 5.2 , 5.36, 5.62,
       5.23, 5.53, 5.52, 5.1 , 5.48, 5.55, 6.23, 5.63, 5.78, 6.06, 5.84,
       6.49, 6.  , 5.98, 5.  , 6.99, 5.38, 6.04, 6.61, 5.73, 6.1 , 5.92,
       6.37, 7.08, 5.86, 5.37, 5.03, 5.44, 6.25, 5.66, 5.46, 4.93, 5.41,
       4.72, 6.48, 6.06, 5.78, 5.22, 5.98, 5.83, 6.33, 5.63, 6.68, 6.39,
       5.2 , 6.95, 6.99, 5.32, 5.67, 7.59, 5.82, 5.69, 6.23, 4.86, 5.99,
       4.89, 5.85, 6.69, 5.9 , 6.  , 6.57, 6.51, 5.82, 5.59, 6.28, 5.71,
       6.11, 6.81, 5.8 , 6.65, 6.07, 6.63, 5.16, 7.77, 6.13, 5.07, 6.16,
       5.98, 6.82, 6.18, 5.07, 5.08, 6.15, 7.27, 6.1 , 4.91, 5.98, 5.28,
       5.19, 5.95, 6.4 , 6.69, 5.79, 5.07, 6.57, 4.92, 5.26, 5.61, 6.38,
```

```
5.5 , 5.84, 6.14, 6.93, 5.71, 6.  , 6.43, 5.14, 5.06, 5.74, 6.26,
5.03, 4.75, 5.81, 5.18, 5.47, 6.13, 5.94, 6.87, 4.87, 5.97, 5.69,
4.89, 7.07, 5.98, 5.04, 5.16, 5.64, 5.96, 5.12, 6.15, 5.47, 5.62,
6.81, 6.05, 5.44, 5.65, 4.63, 6.92, 5.87, 5.03, 6.19, 6.02, 6.18,
6.26, 5.42, 6.  , 6.34, 6.54, 4.79, 5.23, 5.59, 6.52, 5.94, 5.33,
5.14, 6.15, 4.68, 5.36, 5.09, 5.5 , 5.33, 6.1 , 5.07, 6.8 , 6.13,
5.33, 5.61, 7.63, 6.02, 5.09, 5.5 , 5.96, 5.95, 6.75, 7.1 , 7.17,
6.25, 5.37, 5.76, 6.26, 5.63, 6.47, 5.69, 6.49, 5.51, 5.92, 6.61,
5.95, 6.81, 6.74, 5.92, 7.23, 7.01, 7.25, 6.97, 5.4 , 5.86, 7.  ,
6.22, 6.  , 5.34, 6.24, 5.66, 5.03, 5.78, 6.34, 5.35, 5.91, 5.95,
6.62, 6.38, 5.99, 6.69, 5.9 , 6.  , 6.31, 6.11, 5.57, 5.47, 5.25,
5.23, 5.81, 5.19, 6.08, 5.52, 5.52, 7.19, 6.99, 6.61, 5.29, 6.71,
5.99, 6.15, 7.77, 7.08, 5.32, 5.11, 5.11, 6.66, 5.93, 4.96, 5.63,
6.83, 5.39, 5.61, 5.84, 6.59, 5.63, 6.32, 5.92, 5.88, 5.57, 5.83,
5.67, 6.98, 5.67, 5.45, 6.76, 6.39, 5.29, 6.03, 5.81, 5.45, 6.92,
4.77, 5.03, 6.99, 5.32, 6.  , 6.74, 5.58, 5.28, 5.48, 6.06, 5.16,
5.67, 6.79, 5.83, 6.22, 6.28, 5.02, 5.5 , 5.2 , 6.15, 5.67, 6.7 ,
5.75, 6.58, 6.91, 6.67, 6.04, 7.01, 6.61, 5.9 , 5.15, 5.84, 5.03,
7.57, 6.47, 6.76, 6.13, 6.33, 6.2 , 5.77, 5.18, 6.19, 5.63, 5.92,
6.63, 5.11, 5.39, 6.61, 6.1 , 5.79, 6.71, 5.42, 6.23, 6.11, 5.9 ,
6.18, 5.88, 5.98, 6.12, 5.79, 5.16, 5.45, 7.67, 5.77, 5.36, 5.04,
6.84, 6.1 , 7.57, 5.73, 6.  , 6.32, 5.79, 5.1 , 5.76, 5.23, 6.3 ,
6.15, 5.92, 6.9 , 7.12, 5.78, 5.46, 5.04, 6.33, 6.6 , 6.27, 5.51,
5.23, 5.82, 6.28, 5.44, 6.96, 5.28, 5.07, 5.16, 6.77, 5.95, 5.89,
6.07, 6.04, 6.1 , 5.23, 5.02, 5.88, 6.78, 6.06, 5.27, 5.72, 6.37,
6.07, 5.21, 6.31, 6.59, 4.99, 6.92, 6.63, 6.93, 6.23, 6.17, 5.33,
6.46, 6.83, 5.76, 5.46, 6.9 , 6.87, 5.18, 4.92, 5.7 , 5.86, 4.43,
5.93, 5.66, 6.31, 6.03, 6.46, 5.98, 6.77, 5.6 , 6.43, 5.87, 5.92,
5.25, 6.49, 5.38, 6.  , 5.56, 7.  , 6.49, 5.25, 6.8 , 5.39, 6.12,
6.83, 6.08, 5.84, 6.74, 6.31, 6.83, 5.31, 5.43, 6.34, 5.04, 6.79,
5.17, 5.75, 5.07, 5.25, 5.32, 6.07, 5.06, 4.9 , 6.36, 6.03, 5.5 ,
5.17, 7.48, 5.68, 6.33, 5.11, 5.94, 5.92, 6.  , 6.06, 5.55, 6.76,
5.89, 5.8 , 6.06, 5.63, 5.61, 6.41, 6.11, 4.97, 6.45, 6.5 , 5.17,
5.55, 7.5 , 5.44, 5.47, 5.54, 6.65, 5.92, 6.07, 6.39, 6.1 , 5.62,
5.89, 5.38, 5.39, 6.11, 5.91, 6.31, 5.48, 5.97, 6.09, 5.78, 6.  ,
5.85, 5.87, 6.88, 5.34, 6.16, 6.03, 4.8 , 6.44, 5.86, 5.79, 6.01,
5.  , 6.22, 5.6 , 5.38, 5.73, 6.27, 5.76, 7.59, 5.66, 5.93, 5.81,
5.27, 5.61, 6.85, 7.18, 5.11, 5.1 , 4.31, 6.88, 6.27, 5.21, 6.11,
5.51, 6.14, 6.45, 6.84, 5.95, 6.3 , 5.9 , 5.65, 5.04, 6.05, 5.13,
6.07, 6.36, 6.06, 5.94, 6.11, 5.49, 5.7 , 5.2 , 5.94, 5.08, 5.25,
6.  , 5.25, 4.96, 5.73, 5.69, 6.79, 6.12, 6.  , 5.82, 4.9 , 6.01,
5.73, 6.54, 6.98, 5.44, 6.41, 5.81, 6.68, 5.65, 5.24, 6.48, 7.23,
```

```
       6.64, 6.01, 7.01, 5.52, 5.28, 5.15, 6.9 , 5.47, 6.65, 5.23, 5.45,
       6.76, 5.57, 5.56, 5.81, 5.94, 5.12, 6.85, 8.  , 5.5 , 5.29, 5.15,
       5.17, 5.96, 5.98, 5.63, 5.12, 5.69, 5.81, 6.19, 6.23, 6.81, 6.68,
       6.18, 5.19, 6.94, 7.02, 5.26, 7.09, 4.89, 5.24, 5.67, 6.02, 5.73,
       6.69, 4.48, 6.23, 5.45, 5.58, 5.8 , 6.04, 6.03, 6.02, 5.32, 5.2 ,
       5.98])
```

In [ ]:
```
print(y_pred.dtype)

y_pred = np.round(y_pred).astype(int)
y_pred
```

```
float64
```

Out[ ]:
```
array([7, 7, 7, 5, 7, 6, 5, 5, 6, 5, 7, 5, 7, 6, 7, 5, 7, 5, 7, 6, 5, 6,
       5, 6, 6, 5, 5, 6, 7, 6, 5, 5, 6, 6, 5, 7, 6, 5, 5, 6, 5, 6, 6, 6,
       5, 6, 6, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 7, 6, 6, 6, 7, 6, 7,
       6, 5, 6, 6, 5, 6, 5, 4, 6, 6, 6, 6, 7, 7, 6, 6, 6, 6, 5, 7, 5, 7,
       5, 6, 7, 5, 6, 7, 6, 7, 6, 6, 5, 6, 5, 7, 7, 6, 6, 6, 7, 6, 5, 6,
       6, 7, 6, 5, 5, 7, 7, 6, 5, 8, 6, 7, 6, 6, 7, 6, 7, 6, 6, 5, 6, 6,
       6, 6, 5, 6, 6, 6, 7, 5, 7, 6, 6, 6, 7, 5, 5, 5, 5, 5, 6, 6, 6, 6,
       7, 5, 6, 6, 6, 6, 6, 5, 6, 6, 7, 6, 5, 6, 5, 6, 5, 6, 6, 5, 7, 6,
       6, 6, 6, 5, 6, 6, 6, 6, 6, 7, 6, 5, 6, 5, 6, 6, 7, 6, 5, 5, 7, 7,
       5, 6, 7, 5, 6, 6, 6, 6, 7, 7, 6, 6, 6, 7, 6, 7, 5, 5, 6, 6, 6, 6,
       5, 5, 6, 6, 6, 5, 6, 7, 7, 5, 6, 5, 6, 6, 5, 7, 6, 6, 6, 6, 6, 5,
       7, 6, 6, 5, 5, 5, 6, 6, 6, 5, 6, 5, 6, 5, 6, 5, 6, 6, 6, 5, 6, 5,
       7, 6, 6, 6, 6, 5, 5, 6, 7, 7, 6, 6, 5, 6, 6, 5, 6, 6, 6, 5, 6, 7,
       7, 5, 6, 5, 6, 6, 6, 5, 8, 6, 6, 6, 6, 5, 5, 5, 6, 5, 7, 5, 5, 6,
       6, 5, 6, 6, 5, 6, 6, 7, 6, 6, 5, 7, 7, 7, 6, 6, 5, 5, 7, 5, 7, 5,
       6, 6, 6, 6, 5, 6, 7, 6, 7, 6, 6, 6, 5, 6, 5, 5, 6, 4, 7, 6, 7, 7,
       6, 6, 6, 5, 6, 7, 7, 5, 5, 5, 6, 5, 6, 6, 5, 5, 6, 6, 6, 6, 6, 6,
       6, 6, 6, 5, 7, 5, 6, 7, 6, 6, 6, 6, 7, 6, 5, 5, 5, 6, 6, 5, 5, 5,
       5, 6, 6, 6, 5, 6, 6, 6, 6, 7, 6, 5, 7, 7, 5, 6, 8, 6, 6, 6, 5, 6,
       5, 6, 7, 6, 6, 7, 7, 6, 6, 6, 6, 6, 7, 6, 7, 6, 7, 5, 8, 6, 5, 6,
       6, 7, 6, 5, 5, 6, 7, 6, 5, 6, 5, 5, 6, 6, 7, 6, 5, 7, 5, 5, 6, 6,
       6, 6, 6, 7, 6, 6, 6, 5, 5, 6, 6, 5, 5, 6, 5, 5, 6, 6, 7, 5, 6, 6,
       5, 7, 6, 5, 5, 6, 6, 5, 6, 5, 6, 7, 6, 5, 6, 5, 7, 6, 5, 6, 6, 6,
       6, 5, 6, 6, 7, 5, 5, 6, 7, 6, 5, 5, 6, 5, 5, 5, 6, 5, 6, 5, 7, 6,
       5, 6, 8, 6, 5, 6, 6, 6, 7, 7, 7, 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 7,
       6, 7, 7, 6, 7, 7, 7, 7, 5, 6, 7, 6, 6, 5, 6, 6, 5, 6, 6, 5, 6, 6,
       7, 6, 6, 7, 6, 6, 6, 6, 6, 5, 5, 5, 6, 5, 6, 6, 6, 7, 7, 7, 5, 7,
       6, 6, 8, 7, 5, 5, 5, 7, 6, 5, 6, 7, 5, 6, 6, 7, 6, 6, 6, 6, 6, 6,
       6, 7, 6, 5, 7, 6, 5, 6, 6, 5, 7, 5, 5, 7, 5, 6, 7, 6, 5, 5, 6, 5,
       6, 7, 6, 6, 6, 5, 6, 5, 6, 6, 7, 6, 7, 7, 7, 6, 7, 7, 6, 5, 6, 5,
       8, 6, 7, 6, 6, 6, 6, 5, 6, 6, 6, 7, 5, 5, 7, 6, 6, 7, 5, 6, 6, 6,
       6, 6, 6, 6, 6, 5, 5, 8, 6, 5, 5, 7, 6, 8, 6, 6, 6, 6, 5, 6, 5, 6,
       6, 6, 7, 7, 6, 5, 5, 6, 7, 6, 6, 5, 6, 6, 5, 7, 5, 5, 5, 7, 6, 6,
       6, 6, 6, 5, 5, 6, 7, 6, 5, 6, 6, 6, 5, 6, 7, 5, 7, 7, 7, 6, 6, 5,
       6, 7, 6, 5, 7, 7, 5, 5, 6, 6, 4, 6, 6, 6, 6, 6, 6, 7, 6, 6, 6, 6,
       5, 6, 5, 6, 6, 7, 6, 5, 7, 5, 6, 7, 6, 6, 7, 6, 7, 5, 5, 6, 5, 7,
       5, 6, 5, 5, 5, 6, 5, 5, 6, 6, 6, 5, 7, 6, 6, 5, 6, 6, 6, 6, 6, 7,
       6, 6, 6, 6, 6, 6, 6, 5, 6, 6, 5, 6, 8, 5, 5, 6, 7, 6, 6, 6, 6, 6,
       6, 5, 5, 6, 6, 6, 5, 6, 6, 6, 6, 6, 6, 7, 5, 6, 6, 5, 6, 6, 6, 6,
       5, 6, 6, 5, 6, 6, 6, 8, 6, 6, 6, 5, 6, 7, 7, 5, 5, 4, 7, 6, 5, 6,
       6, 6, 6, 7, 6, 6, 6, 6, 5, 6, 5, 6, 6, 6, 6, 6, 5, 6, 5, 6, 5, 5,
       6, 5, 5, 6, 6, 7, 6, 6, 6, 5, 6, 6, 7, 7, 5, 6, 6, 7, 6, 5, 6, 7,
```

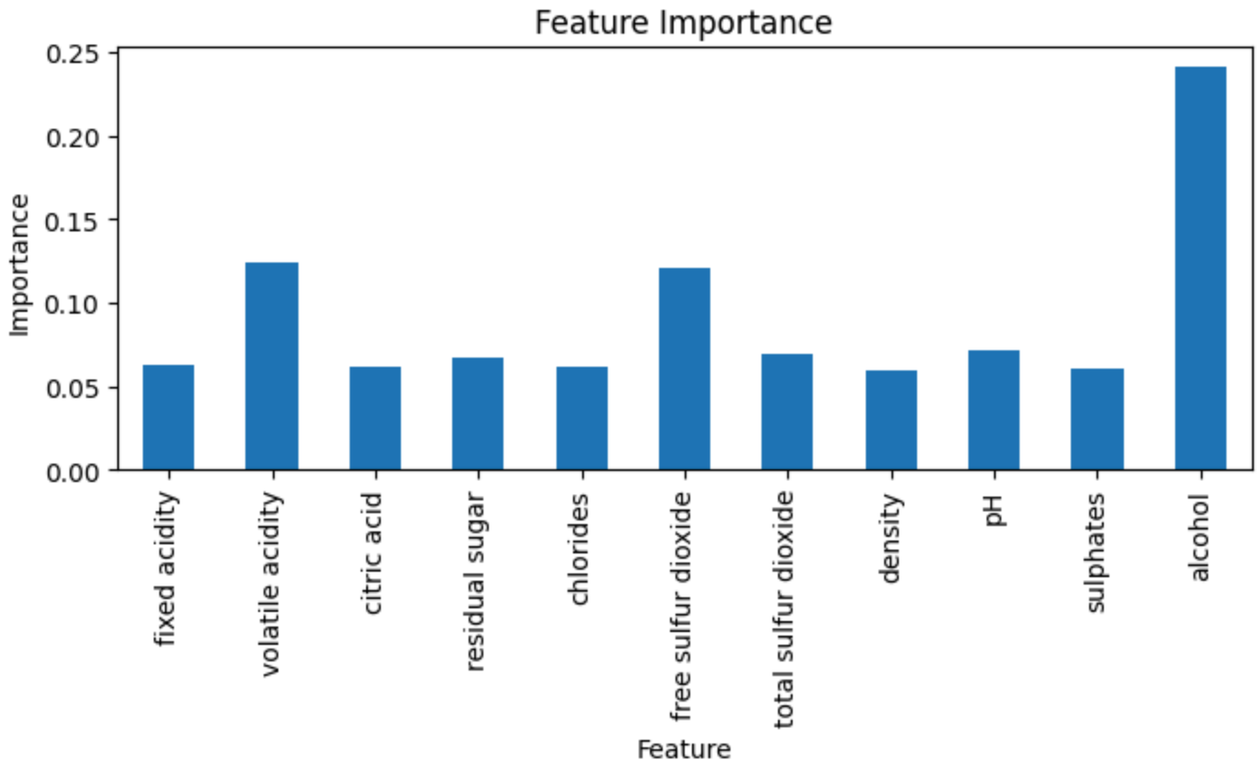
```
       7, 6, 7, 6, 5, 5, 7, 5, 7, 5, 5, 7, 6, 6, 6, 6, 5, 7, 8, 6, 5, 5,
       5, 6, 6, 6, 5, 6, 6, 6, 6, 7, 7, 6, 5, 7, 7, 5, 7, 5, 5, 6, 6, 6,
       7, 4, 6, 5, 6, 6, 6, 6, 6, 5, 5, 6])
```

In [ ]:
```
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
```

```
Accuracy: 0.69
```

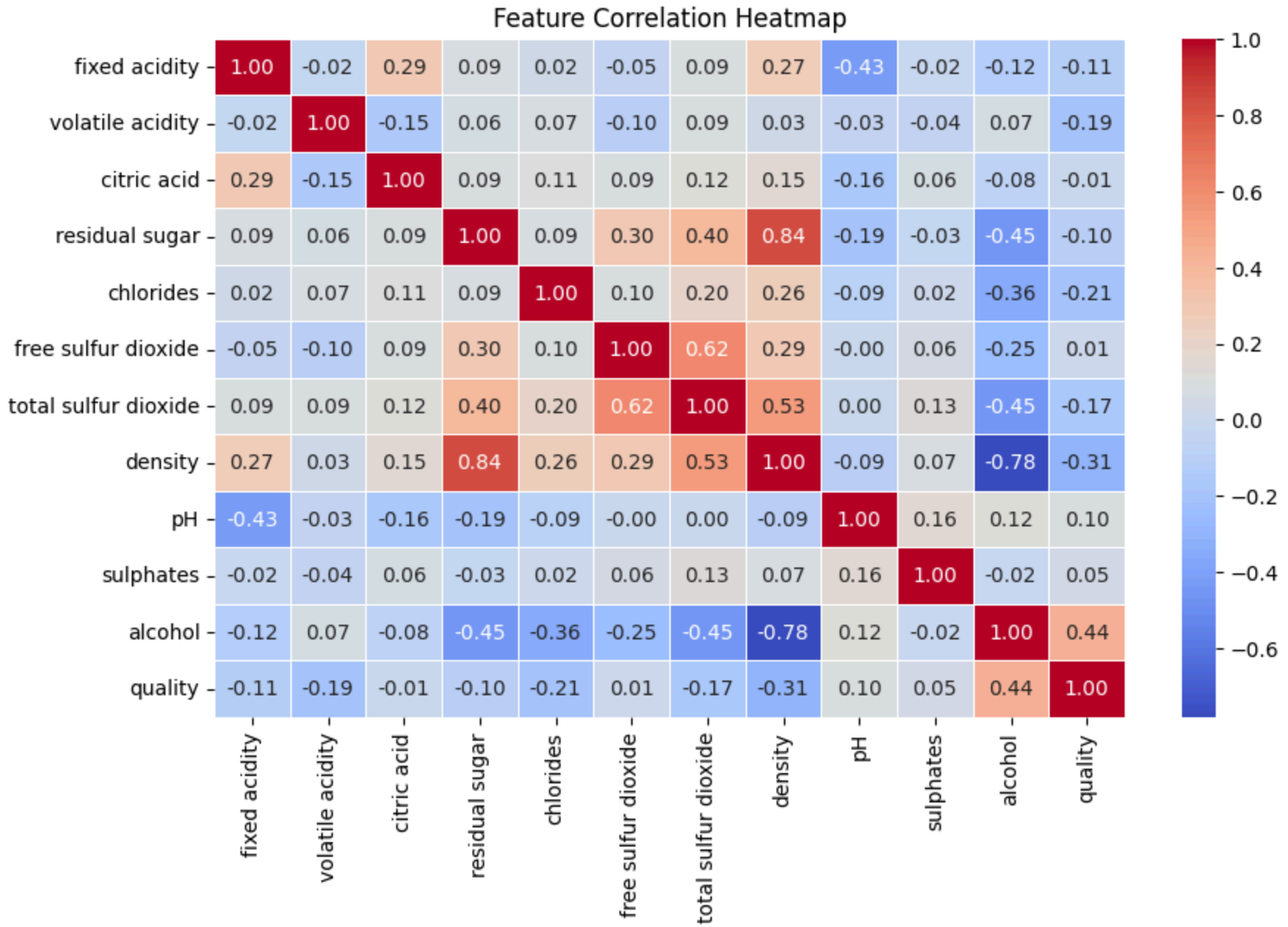
In [ ]:
```
imp_col = reg.feature_importances_
```

In [ ]:
```
fig, ax = plt.subplots(1, 1, figsize=(8, 3))
labels = X.columns
pd.Series(imp_col, index=labels).plot(kind='bar', ax=ax)
ax.set_title('Feature Importance')
ax.set_ylabel('Importance')
ax.set_xlabel('Feature')
plt.xticks(rotation=90)
plt.show()
```



In [ ]:
```
corr_matrix = df.corr()

plt.figure(figsize=(10, 6))
sns.heatmap(corr_matrix, annot=True, fmt=".2f", cmap="coolwarm", linewidths=0.5)
plt.title("Feature Correlation Heatmap")
plt.show()
```


Feature Correlation Heatmap
fixed acidity 1.00 -0.02 0.29 0.09 0.02 -0.05 0.09 0.27 -0.43 -0.02 -0.12 -0.11
volatile acidity -0.02 1.00 -0.15 0.06 0.07 -0.10 0.09 0.03 -0.03 -0.04 0.07 -0.19
citric acid 0.29 -0.15 1.00 0.09 0.11 0.09 0.12 0.15 -0.16 0.06 -0.08 -0.01
residual sugar 0.09 0.06 0.09 1.00 0.09 0.30 0.40 0.84 -0.19 -0.03 -0.45 -0.10
chlorides 0.02 0.07 0.11 0.09 1.00 0.10 0.20 0.26 -0.09 0.02 -0.36 -0.21
free sulfur dioxide -0.05 -0.10 0.09 0.30 0.10 1.00 0.62 0.29 -0.00 0.06 -0.25 0.01
total sulfur dioxide 0.09 0.09 0.12 0.40 0.20 0.62 1.00 0.53 0.00 0.13 -0.45 -0.17
density 0.27 0.03 0.15 0.84 0.26 0.29 0.53 1.00 -0.09 0.07 -0.78 -0.31
pH -0.43 -0.03 -0.16 -0.19 -0.09 -0.00 0.00 -0.09 1.00 0.16 0.12 0.10
sulphates -0.02 -0.04 0.06 -0.03 0.02 0.06 0.13 0.07 0.16 1.00 -0.02 0.05
alcohol -0.12 0.07 -0.08 -0.45 -0.36 -0.25 -0.45 -0.78 0.12 -0.02 1.00 0.44
quality -0.11 -0.19 -0.01 -0.10 -0.21 0.01 -0.17 -0.31 0.10 0.05 0.44 1.00
fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol
quality
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6