

explainable-ai-lime

December 30, 2023

```
[61]: # importing the libraries
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns

from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report
```

```
[62]: # import the dataset
df = pd.read_csv('/kaggle/input/winedataset/wine.csv')
df.head()
```

```
[62]:
```

	Class	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	\
0	1	14.23	1.71	2.43	15.6	127	
1	1	13.20	1.78	2.14	11.2	100	
2	1	13.16	2.36	2.67	18.6	101	
3	1	14.37	1.95	2.50	16.8	113	
4	1	13.24	2.59	2.87	21.0	118	

	Total phenols	Flavanoids	Nonflavanoid phenols	Proanthocyanins	\
0	2.80	3.06	0.28	2.29	
1	2.65	2.76	0.26	1.28	
2	2.80	3.24	0.30	2.81	
3	3.85	3.49	0.24	2.18	
4	2.80	2.69	0.39	1.82	

	Color intensity	Hue	OD280/OD315 of diluted wines	Proline
0	5.64	1.04	3.92	1065
1	4.38	1.05	3.40	1050
2	5.68	1.03	3.17	1185
3	7.80	0.86	3.45	1480
4	4.32	1.04	2.93	735

```
[63]: # checking the dimensions of the dataframe
df.shape
```

```
[63]: (178, 14)
```

```
[64]: # checking for missing values
df.isnull().sum()
```

```
[64]: Class                                0
      Alcohol                             0
      Malic acid                           0
      Ash                                  0
      Alcalinity of ash                     0
      Magnesium                             0
      Total phenols                         0
      Flavanoids                           0
      Nonflavanoid phenols                  0
      Proanthocyanins                       0
      Color intensity                       0
      Hue                                   0
      OD280/OD315 of diluted wines         0
      Proline                              0
      dtype: int64
```

```
[65]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 14 columns):
 #   Column                                  Non-Null Count  Dtype
---  -
 0   Class                                  178 non-null   int64
 1   Alcohol                               178 non-null   float64
 2   Malic acid                            178 non-null   float64
 3   Ash                                   178 non-null   float64
 4   Alcalinity of ash                     178 non-null   float64
 5   Magnesium                             178 non-null   int64
 6   Total phenols                         178 non-null   float64
 7   Flavanoids                           178 non-null   float64
 8   Nonflavanoid phenols                  178 non-null   float64
 9   Proanthocyanins                       178 non-null   float64
10   Color intensity                       178 non-null   float64
11   Hue                                   178 non-null   float64
12   OD280/OD315 of diluted wines         178 non-null   float64
13   Proline                              178 non-null   int64
dtypes: float64(11), int64(3)
memory usage: 19.6 KB
```

```
[66]: df.dtypes
```

```
[66]: Class                                int64
      Alcohol                            float64
      Malic acid                          float64
      Ash                                float64
      Alcalinity of ash                   float64
      Magnesium                           int64
      Total phenols                       float64
      Flavanoids                          float64
      Nonflavanoid phenols                float64
      Proanthocyanins                     float64
      Color intensity                     float64
      Hue                                float64
      OD280/OD315 of diluted wines       float64
      Proline                             int64
      dtype: object
```

1 KDE Plot

```
[67]: sns.set(style="whitegrid")

      # List of variables to plot
      variables_to_plot = ['Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash',
                           ↪ 'Magnesium',
                           'Total phenols', 'Flavanoids', 'Nonflavanoid phenols',
                           ↪ 'Proanthocyanins',
                           'Color intensity', 'Hue', 'OD280/OD315 of diluted wines']
```

```
[68]: # Remove extra space in column names
      df.columns = df.columns.str.strip()
```

```
[69]: # Calculate the number of rows and columns for the subplot grid
      num_variables = len(variables_to_plot)
      num_rows = (num_variables - 1) // 3 + 1
      num_cols = min(num_variables, 3)
```

```
[70]: # Create subplots with a dynamic grid and vertical spacing
      fig, axes = plt.subplots(nrows=num_rows, ncols=num_cols, figsize=(15, 15))

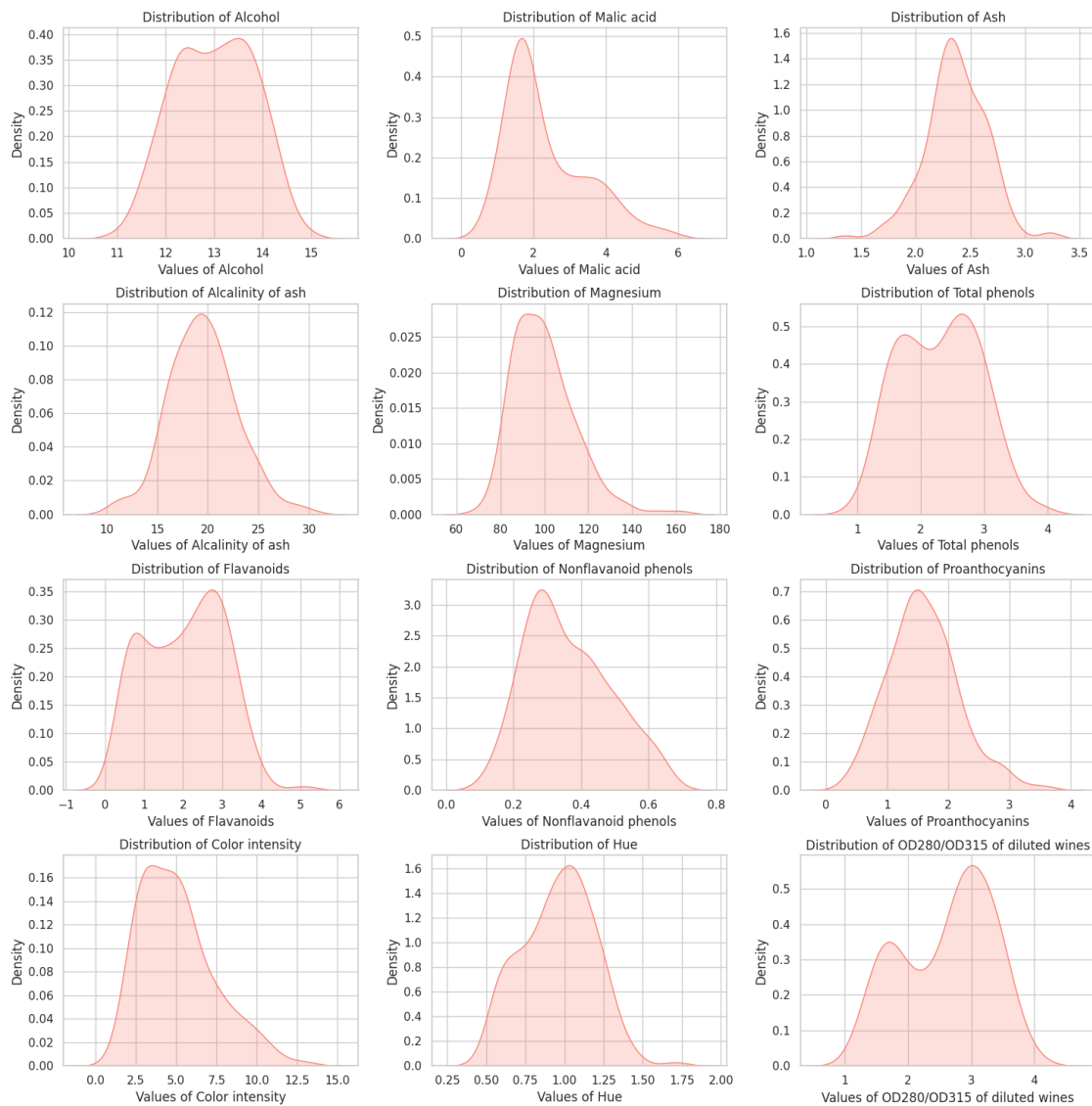
      # Flatten the axes for easy iteration
      axes = axes.flatten()

      # Loop through variables and plot KDE on each subplot
      for i, variable in enumerate(variables_to_plot):
```

```

sns.kdeplot(df[variable], ax=axes[i], color='salmon', fill=True,
↪common_norm=False)
axes[i].set_title(f'Distribution of {variable}')
axes[i].set_xlabel(f'Values of {variable}')
axes[i].set_ylabel('Density')
plt.tight_layout(h_pad=0.8)
plt.show()

```



2 Model Building

```
[71]: # splitting the data into independent and dependent variables
x = df.drop(columns=['Class'])
y = df['Class']
```

```
[72]: # diving the dataset into training and testing sets
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, train_size=0.7,
↳random_state=42)
```

```
[73]: # building the model
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(n_estimators=100, max_depth=5,
↳min_samples_split=2, min_samples_leaf=1, random_state=42)
clf.fit(x_train, y_train)
```

```
[73]: RandomForestClassifier(max_depth=5, random_state=42)
```

```
[74]: # importing the necessary libraries
from sklearn.model_selection import GridSearchCV

# Define the parameter grid
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [3, 5, 7],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
```

```
[75]: # Perform GridSearchCV
grid_search = GridSearchCV(clf, param_grid, cv=5)
grid_search.fit(x_train, y_train)
```

```
[75]: GridSearchCV(cv=5,
                  estimator=RandomForestClassifier(max_depth=5, random_state=42),
                  param_grid={'max_depth': [3, 5, 7], 'min_samples_leaf': [1, 2, 4],
                              'min_samples_split': [2, 5, 10],
                              'n_estimators': [50, 100, 200]})
```

```
[76]: # obtaining the best model
clf = grid_search.best_estimator_
```

```
[77]: # Make predictions on the test set
y_pred = clf.predict(x_test)
```

```
[78]: # obtaining the classification report
from sklearn.metrics import classification_report
report = classification_report(y_test, y_pred)
print(report)
```

	precision	recall	f1-score	support
1	1.00	1.00	1.00	19
2	1.00	1.00	1.00	21
3	1.00	1.00	1.00	14
accuracy			1.00	54
macro avg	1.00	1.00	1.00	54
weighted avg	1.00	1.00	1.00	54

3 Explainability via Lime

```
[79]: # importing lime
import lime
from lime import lime_tabular
```

```
[80]: # creating an instance of the lime tabular explainer
lime_explainer = lime_tabular.LimeTabularExplainer(training_data=np.
↳array(x_train), feature_names=x_train.columns,
class_names=['1', '2', '3'], mode='classification')
```

```
[81]: # obtaining the explanation
explanation = lime_explainer.explain_instance(data_row=x_test.iloc[1],
↳predict_fn=clf.predict_proba, top_labels=6, num_features=13)

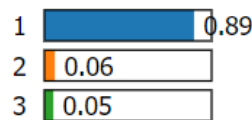
# printing out the explanation
explanation.show_in_notebook()
```

/opt/conda/lib/python3.10/site-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but RandomForestClassifier was fitted with feature names

```
warnings.warn(
```

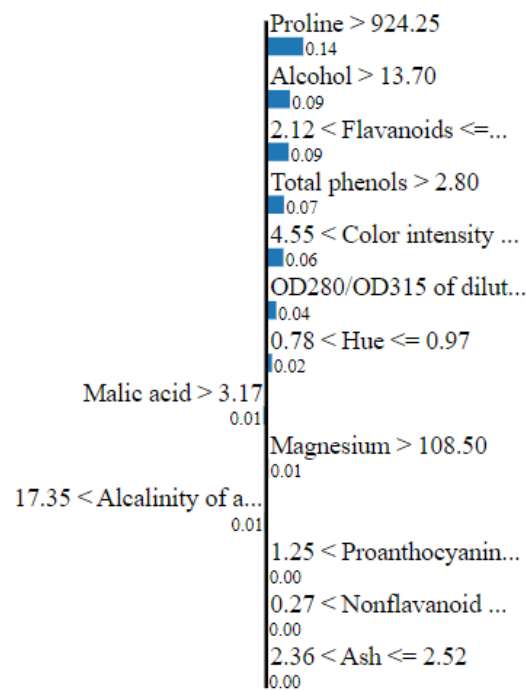
```
<IPython.core.display.HTML object>
```

Prediction probabilities



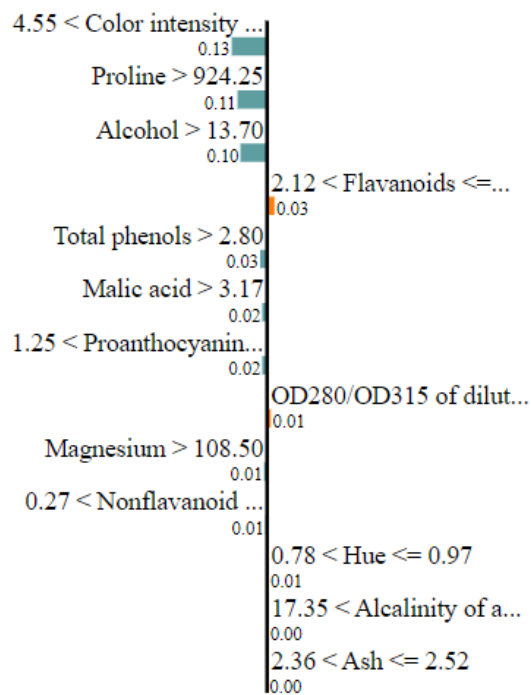
NOT 1

1



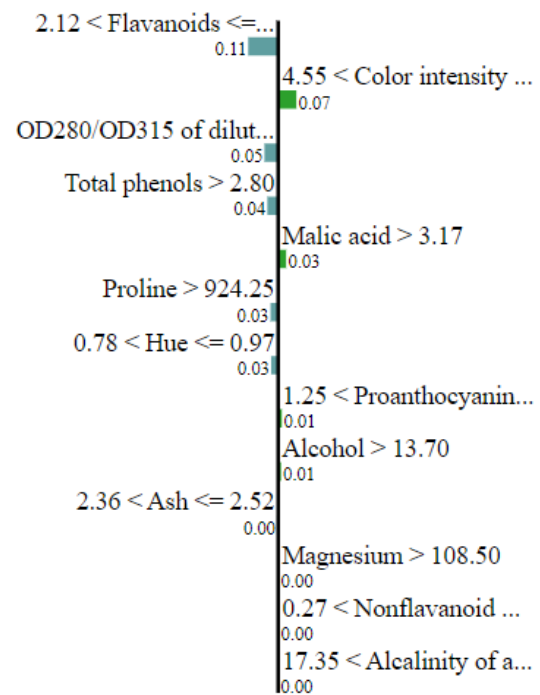
NOT 2

2



NOT 3

3



Feature	Value
Proline	1080.00
Alcohol	14.21
Flavanoids	2.65
Total phenols	2.85
Color intensity	5.24
OD280/OD315 of diluted wines	3.33
Hue	0.87
Malic acid	4.04
Magnesium	111.00
Alcalinity of ash	18.90