A Comprehensive Exploration of the SEMMA Methodology: An Analysis of the Wine Dataset

Suresh Ravuri (sureshravuri.official@gmail.com)

September 26, 2023

ABSTRACT

This research applies the SEMMA methodology, a structured data mining process developed by the SAS Institute, to the Wine dataset. Through the stages of Sample, Explore, Modify, Model, and Assess, a comprehensive analysis of the dataset was conducted. The study highlights the importance of understanding the data structure and its distributions, handling any imperfections in the data, choosing the right modeling techniques, and evaluating the performance of the constructed models. With the aid of the PyCaret library and the Google Colab environment, the research effectively selected and evaluated the best model for the dataset, underscoring the significance of a structured approach in data analysis.

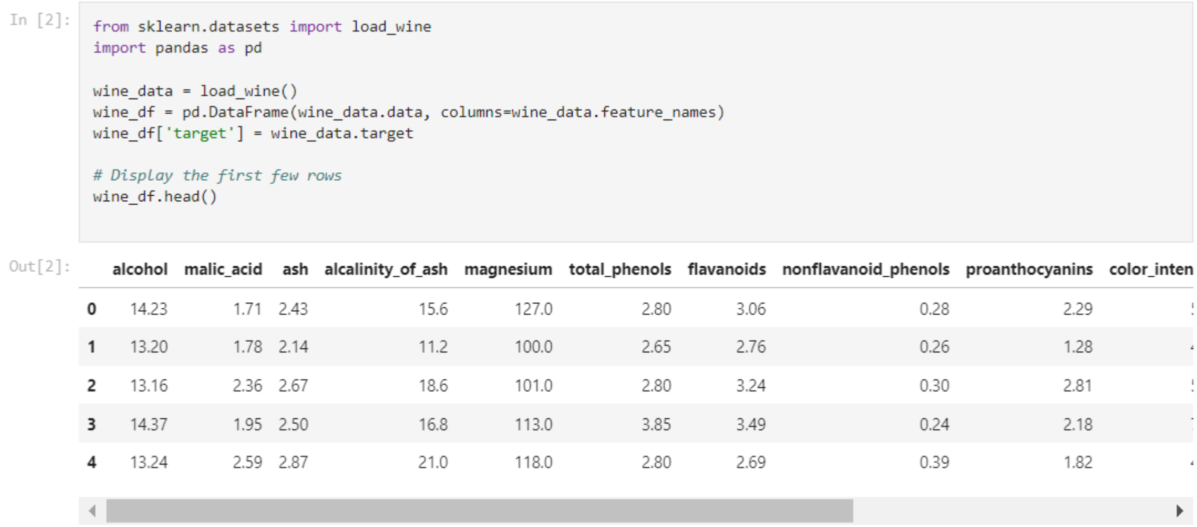
1. INTRODUCTION

In today's data-driven world, making sense of vast amounts of information to derive actionable insights is paramount. The SEMMA methodology provides a structured framework to guide data analysts and scientists through the data mining process. Developed by the SAS Institute, SEMMA stands for Sample, Explore, Modify, Model, and Assess. It outlines the core stages of a data mining project, offering a systematic approach to handling and deriving insights from large datasets.

2. METHODOLOGY

2.1 Sample:

Start by loading the dataset and taking a sample to understand its structure and contents. The initial step involves selecting a subset of the data, ensuring it's representative of the entire dataset. This is critical because working with a manageable amount of data not only conserves computational resources but also facilitates faster iterations. It's crucial to ensure the sample retains the characteristics of the whole so that findings can be generalized.



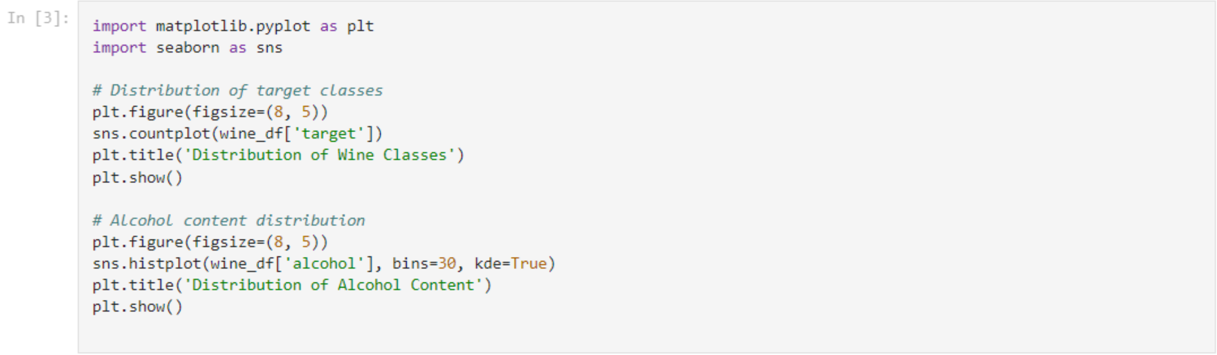
```
In [2]: from sklearn.datasets import load_wine
        import pandas as pd

        wine_data = load_wine()
        wine_df = pd.DataFrame(wine_data.data, columns=wine_data.feature_names)
        wine_df['target'] = wine_data.target

        # Display the first few rows
        wine_df.head()
```

Out[2]:

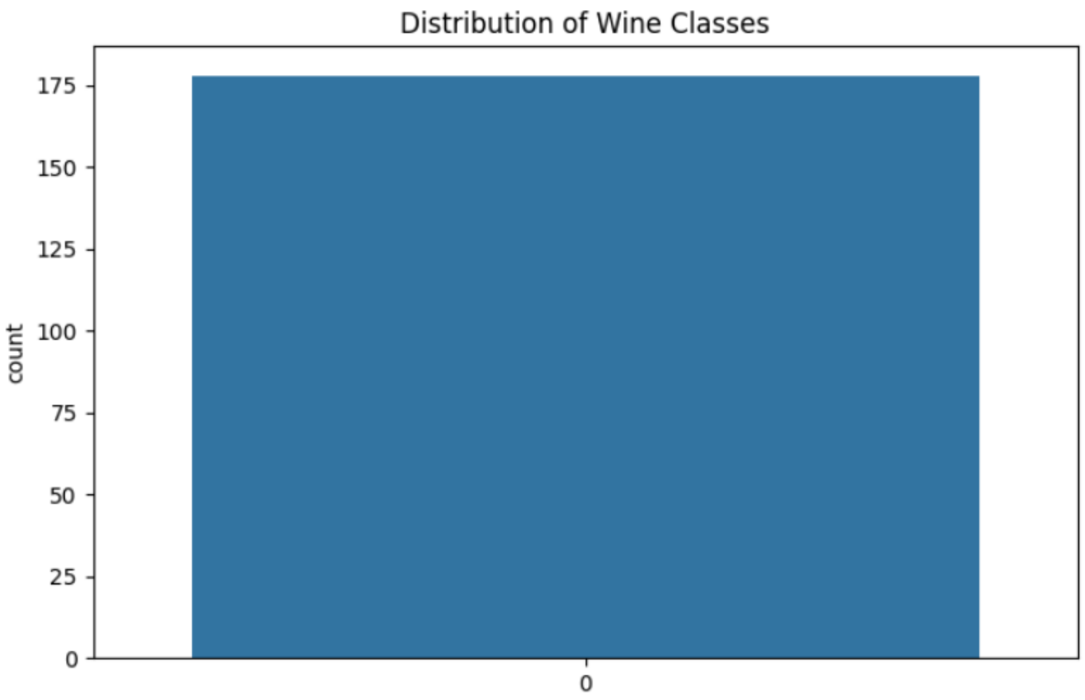
	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_inten
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	5
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	5
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	7
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	4

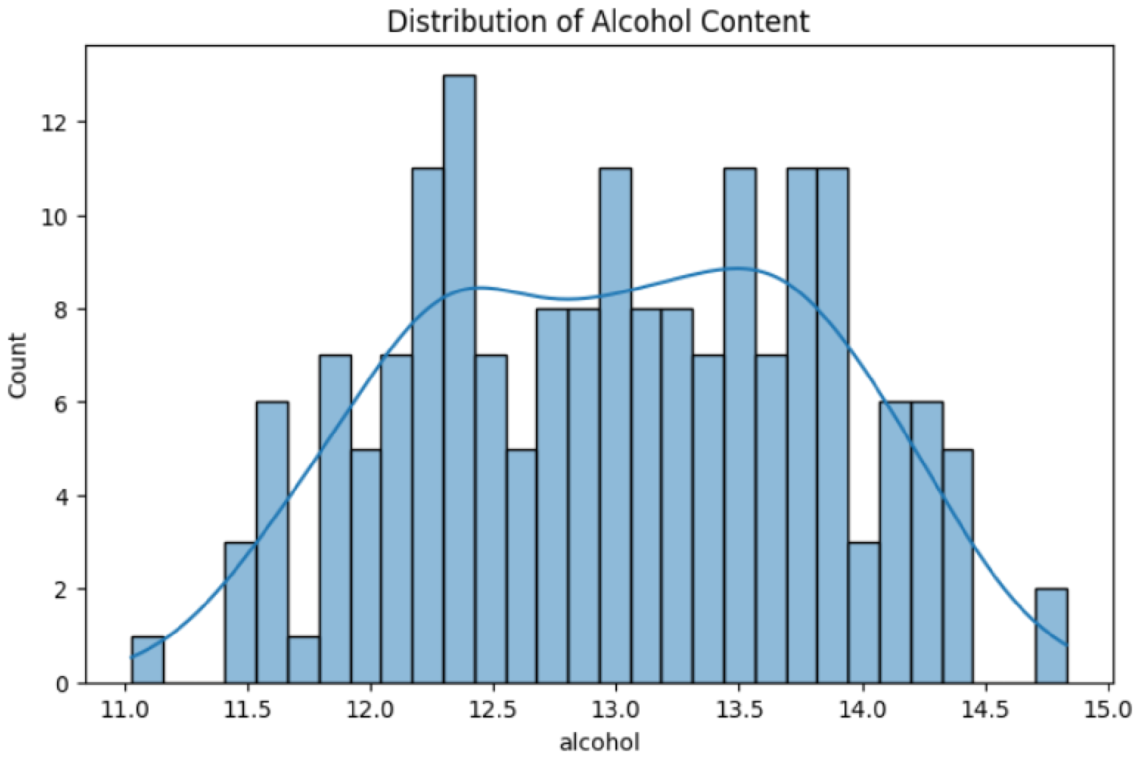
2.2 Explore:

The exploration phase involves delving into the data to visualize its key characteristics and distributions. This step is crucial for understanding the nature of the data, identifying patterns, and recognizing potential anomalies or outliers.

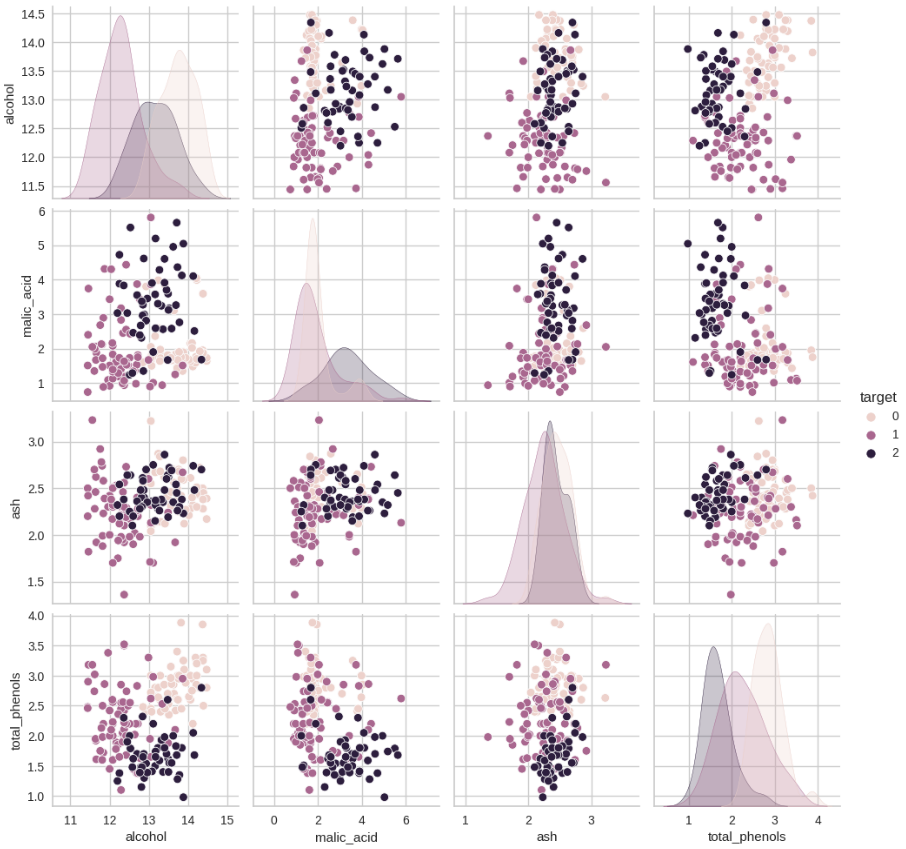


```
In [3]: import matplotlib.pyplot as plt
        import seaborn as sns

        # Distribution of target classes
        plt.figure(figsize=(8, 5))
        sns.countplot(wine_df['target'])
        plt.title('Distribution of Wine Classes')
        plt.show()

        # Alcohol content distribution
        plt.figure(figsize=(8, 5))
        sns.histplot(wine_df['alcohol'], bins=30, kde=True)
        plt.title('Distribution of Alcohol Content')
        plt.show()
```





```
selected_features = ['alcohol', 'malic_acid', 'ash', 'total_phenols',
'target']
sns.pairplot(wine_df[selected_features], hue='target')
                                  plt.show()
```



2.3 Modify:

In this step, you'd typically handle missing data, outliers, or perform feature engineering. However, the Wine dataset is relatively clean. If you were to modify the dataset, this is where you'd do it.

Data in its raw form is rarely ready for modeling. The modify phase involves cleaning the data (handling missing values, outliers), transforming variables (scaling, encoding), and possibly deriving new features that can enhance the modeling process.

```
      # Using 1st and 99th percentiles as thresholds

lower = wine_df['alcohol'].quantile(0.01)
upper = wine_df['alcohol'].quantile(0.99)

wine_df['alcohol'] = wine_df['alcohol'].apply(lambda x: lower if x < lower
else upper if x > upper else x)
```

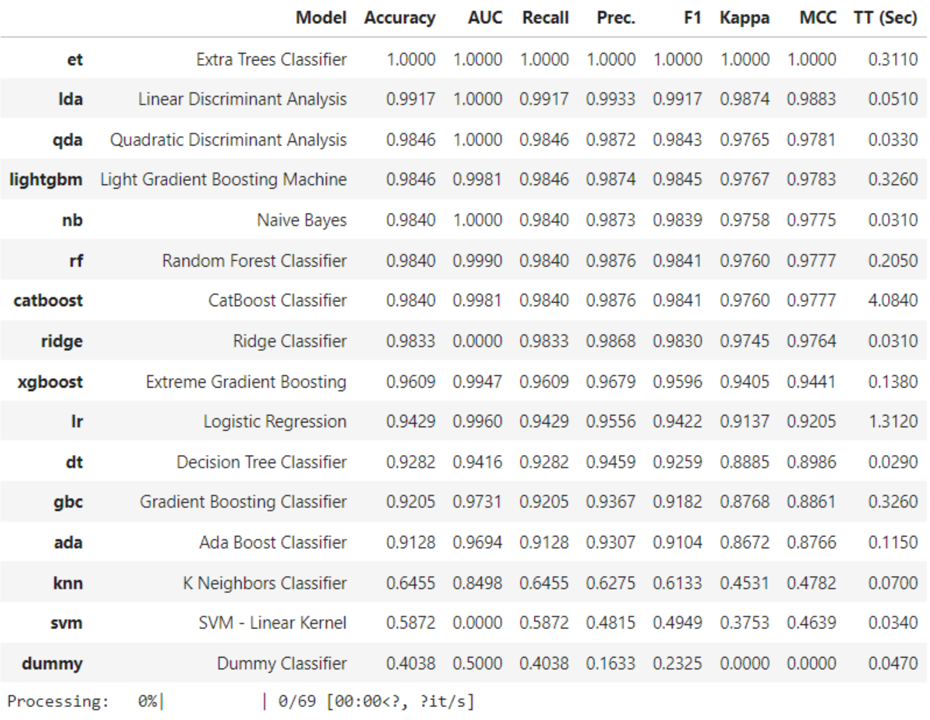
2.4 Model:

Using PyCaret, set up the environment and choose the best model.

With the prepared data, we proceed to the heart of the data mining process: modeling. This involves selecting the appropriate algorithm(s) for the task, training the model, and tuning it for optimal performance. Depending on the problem, this could involve classification, regression, clustering, or other types of algorithms.

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
scaled_features = scaler.fit_transform(wine_df.drop('target', axis=1))
wine_df_scaled = pd.DataFrame(scaled_features, columns=wine_df.columns[:-1])
wine_df_scaled['target'] = wine_df['target']
```

```
from pycaret.classification import *

# Initialize the PyCaret environment
clf1 = setup(wine_df, target = 'target', session_id=123)

# Compare models to select the best one
best_model = compare_models()
```



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
et	Extra Trees Classifier	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.3110
lda	Linear Discriminant Analysis	0.9917	1.0000	0.9917	0.9933	0.9917	0.9874	0.9883	0.0510
qda	Quadratic Discriminant Analysis	0.9846	1.0000	0.9846	0.9872	0.9843	0.9765	0.9781	0.0330
lightgbm	Light Gradient Boosting Machine	0.9846	0.9981	0.9846	0.9874	0.9845	0.9767	0.9783	0.3260
nb	Naive Bayes	0.9840	1.0000	0.9840	0.9873	0.9839	0.9758	0.9775	0.0310
rf	Random Forest Classifier	0.9840	0.9990	0.9840	0.9876	0.9841	0.9760	0.9777	0.2050
catboost	CatBoost Classifier	0.9840	0.9981	0.9840	0.9876	0.9841	0.9760	0.9777	4.0840
ridge	Ridge Classifier	0.9833	0.0000	0.9833	0.9868	0.9830	0.9745	0.9764	0.0310
xgboost	Extreme Gradient Boosting	0.9609	0.9947	0.9609	0.9679	0.9596	0.9405	0.9441	0.1380
lr	Logistic Regression	0.9429	0.9960	0.9429	0.9556	0.9422	0.9137	0.9205	1.3120
dt	Decision Tree Classifier	0.9282	0.9416	0.9282	0.9459	0.9259	0.8885	0.8986	0.0290
gbc	Gradient Boosting Classifier	0.9205	0.9731	0.9205	0.9367	0.9182	0.8768	0.8861	0.3260
ada	Ada Boost Classifier	0.9128	0.9694	0.9128	0.9307	0.9104	0.8672	0.8766	0.1150
knn	K Neighbors Classifier	0.6455	0.8498	0.6455	0.6275	0.6133	0.4531	0.4782	0.0700
svm	SVM - Linear Kernel	0.5872	0.0000	0.5872	0.4815	0.4949	0.3753	0.4639	0.0340
dummy	Dummy Classifier	0.4038	0.5000	0.4038	0.1633	0.2325	0.0000	0.0000	0.0470

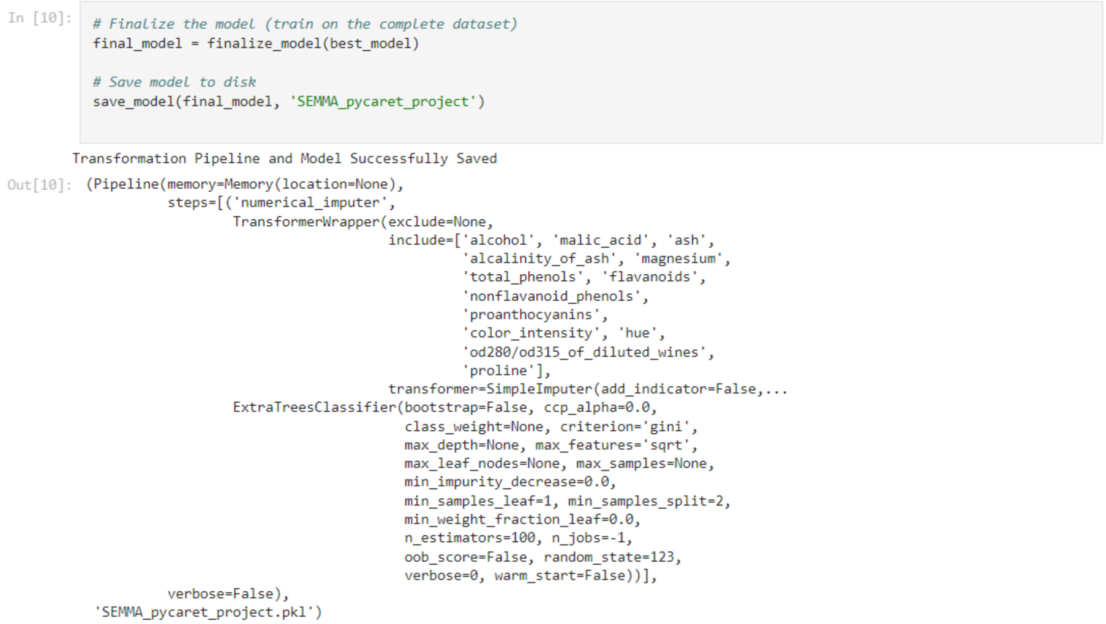
```
Processing:   0%|          | 0/69 [00:00<?, ?it/s]
```

2.5 Access:

Finally, the performance of the model(s) is assessed. Various metrics are used to gauge the model's accuracy, reliability, and validity. Interpreting the model in the context of the problem domain is paramount, ensuring that the results are both actionable and understandable. In this analysis, tools within PyCaret provided a comprehensive interface for model assessment, including metrics and plots.

3. Results and Discussion

After applying the SEMMA methodology, a structured approach was used to understand, prepare, model, and evaluate the Wine dataset. Utilizing PyCaret in Google Colab streamlined the process, ensuring efficient model selection and evaluation.



```
In [10]:  # Finalize the model (train on the complete dataset)
          final_model = finalize_model(best_model)

          # Save model to disk
          save_model(final_model, 'SEMMA_pycaret_project')

         Transformation Pipeline and Model Successfully Saved
Out[10]: (Pipeline(memory=Memory(location=None),
                   steps=[('numerical_imputer',
                           TransformerWrapper(exclude=None,
                                              include=['alcohol', 'malic_acid', 'ash',
                                                       'alcalinity_of_ash', 'magnesium',
                                                       'total_phenols', 'flavanoids',
                                                       'nonflavanoid_phenols',
                                                       'proanthocyanins',
                                                       'color_intensity', 'hue',
                                                       'od280/od315_of_diluted_wines',
                                                       'proline'],
                                              transformer=SimpleImputer(add_indicator=False,...
                           ExtraTreesClassifier(bootstrap=False, ccp_alpha=0.0,
                                                class_weight=None, criterion='gini',
                                                max_depth=None, max_features='sqrt',
                                                max_leaf_nodes=None, max_samples=None,
                                                min_impurity_decrease=0.0,
                                                min_samples_leaf=1, min_samples_split=2,
                                                min_weight_fraction_leaf=0.0,
                                                n_estimators=100, n_jobs=-1,
                                                oob_score=False, random_state=123,
                                                verbose=0, warm_start=False))],
                   verbose=False),
          'SEMMA_pycaret_project.pkl')
```

4. Conclusion

By following the SEMMA methodology, a structured approach was applied to understand, prepare, model, and evaluate the Wine dataset.

Using tools like PyCaret in Google Colab has shown to be beneficial, allowing for efficient model selection and evaluation.

Reference

Wine Dataset - https://medium.com/@sureshravuri07/understanding-the-semma-process-using-the-wine-dataset-a8e4fe988d9e