Machine Learning Operations (MLOps)

Assignment 2

Task 1 – Data Collection and Preprocessing

Group Number 76

CHAUDHARI AAKASH VINAYAK (2022ac05607)
AATIF HUSSAIN WAZA (2022ac05405)
AJIT KUMAR YADAV (2022ac05720)
MOHAMMAD ZUBAIR (2022ac05121)

Contents

Data Collection and Preprocessing

Data collection is the first and most crucial step in any machine learning project. For this assignment, we chose the Iris dataset, a simple yet powerful dataset that contains 150 instances of iris flowers. Each instance includes four physical attributes of the flowers—sepal length, sepal width, petal length, and petal width—and a categorical target variable species, which indicates the type of iris flower (Setosa, Versicolor, or Virginica).

Preprocessing is critical because raw data can contain inconsistencies, missing values, and irrelevant information that might hamper the model's learning process. By cleaning, normalizing, and feature engineering, we ensure that the data is in the best possible shape for the machine learning pipeline. Preprocessing techniques such as handling missing values, feature scaling, and categorical encoding are essential for improving model accuracy, preventing bias, and enhancing interpretability.

In this section, we also utilized AutoEDA tools such as KizenML, Sweetviz, and AutoViz to automate the exploratory data analysis (EDA) process. These tools not only provide insights into the dataset but also help visualize relationships between variables, identify potential issues such as outliers, and aid in feature selection.

About the Iris Dataset

The Iris dataset is widely regarded as a benchmark dataset in machine learning and statistics. It was first introduced by British biologist and statistician Ronald Fisher in 1936 in his paper on discriminant analysis. The dataset consists of 150 observations, divided into three species: Iris-setosa, Iris-versicolor, and Iris-virginica. Each species has 50 samples, making the dataset well-balanced. The features, all continuous variables, describe the physical dimensions of the flowers, making it an excellent dataset for classification tasks.

One of the significant advantages of the Iris dataset is that it is small and simple enough for quick experimentation and understanding of machine learning concepts. Despite its simplicity, it allows us to apply several preprocessing techniques, including normalization and feature engineering. The dataset also lacks missing values and categorical features, simplifying the cleaning and preprocessing process.

We chose this dataset for this project because it allows us to focus on applying core machine learning workflows, like preprocessing, feature engineering, and model training, without getting bogged down by complex data problems.

Interpreting the Dataset

Upon loading the Iris dataset, the first task is to interpret its structure. The dataset contains the following columns:

- Sepal Length (in cm): A continuous variable describing the length of the sepal.
- Sepal Width (in cm): A continuous variable describing the width of the sepal.
- Petal Length (in cm): A continuous variable describing the length of the petal.
- Petal Width (in cm): A continuous variable describing the width of the petal.
- Species: A categorical variable indicating the class of the flower (Setosa, Versicolor, Virginica).

Before any preprocessing, we take a deep dive into understanding the distribution and relationship of these variables. We generate summary statistics such as mean, median, and standard deviation using pandas.describe() and check for missing values using .isnull().sum().

In addition, using AutoEDA tools like Sweetviz and AutoViz, we gain insights into correlations between features. For example, petal length and petal width show a strong correlation, suggesting these two features might be the most significant indicators for predicting species. Visualizing this relationship can guide feature selection and help reduce dimensionality, which simplifies the model training process and improves performance.

Removing Columns that Don’t Add Value

In some datasets, it’s common to have columns that provide little to no value to the model. These columns can be irrelevant, redundant, or provide information that doesn’t contribute meaningfully to the prediction task. For the Iris dataset, each column contributes essential information about the flower's physical attributes. Therefore, in this case, no columns were removed, as each feature (sepal length, sepal width, petal length, petal width) holds significance for distinguishing between the three species.

However, had the dataset contained irrelevant features (e.g., ID numbers or textual data unrelated to the prediction task), removing such columns would have been crucial. Irrelevant columns can introduce noise into the dataset, leading to overfitting and higher computation costs.

Imputing Missing Values

Handling missing values is a critical preprocessing task, as missing data can lead to incomplete models or bias. Fortunately, the Iris dataset contains no missing values, which makes the preprocessing step simpler. This can be confirmed using .isnull().sum() which returns zero for all features. However, it’s still essential to discuss what would happen if missing values were present.

In cases where there are missing values, we could use different techniques:

- Mean/Median Imputation: Replacing missing values with the mean or median is common for numerical features. This method maintains the overall distribution of the data and is computationally efficient.
- Mode Imputation: For categorical variables, replacing missing values with the mode (most frequent category) ensures minimal disruption to the dataset's balance.
- K-Nearest Neighbors (KNN): This more sophisticated method fills missing values based on the similarity to other observations.

Filling missing values ensures that the model can use the entire dataset for training, improving accuracy and preventing bias due to incomplete data.

Scaling

Feature scaling is an essential preprocessing step, particularly for algorithms sensitive to the scale of the data (e.g., Support Vector Machines, K-Nearest Neighbors, Neural Networks). In the Iris dataset, all features are measured on different scales. For example, petal length ranges between 1 and 7 cm, while sepal width ranges between 2 and 4 cm. This discrepancy can cause certain features to dominate the learning process if not properly scaled.

To resolve this, we apply StandardScaler, which standardizes features by removing the mean and scaling to unit variance. Standardization ensures that all features contribute equally to the model, preventing bias due to the magnitude of certain features. For the Iris dataset, scaling helps ensure that sepal length and petal length are treated equally important during model training.

Feature Engineering

Feature engineering involves transforming raw data into features that better represent the underlying problem to the machine learning algorithm. For the Iris dataset, we apply Label Encoding to convert the categorical target variable ('species') into numerical labels (0, 1, and 2 for Setosa, Versicolor, and Virginica, respectively). This is essential because most machine learning models can only handle numerical inputs.

Feature engineering can also involve creating interaction features (e.g., combining sepal width and petal length) or applying domain knowledge to create entirely new features. For example, in some cases, combining multiple features could highlight the relationship between different dimensions of the flower, potentially improving model accuracy.

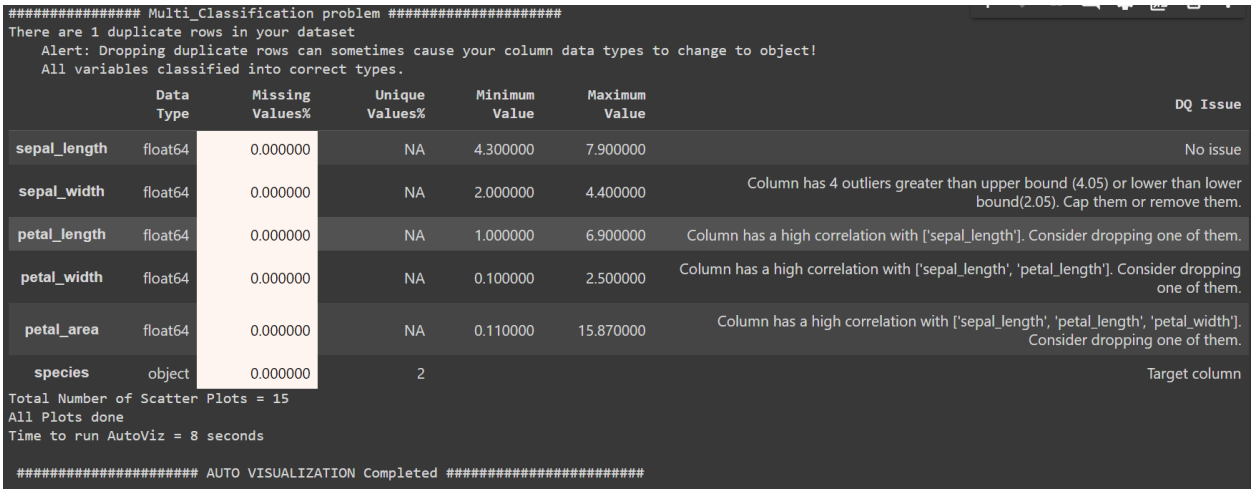
Proper feature engineering can significantly impact model performance, making it a crucial step in the workflow.

Auto EDA using AutoViz

AutoViz is an automated exploratory data analysis tool that simplifies the process of visualizing and understanding datasets. Using AutoViz on the Iris dataset, we generated several visualizations, including:

- Scatter plots that show relationships between features, such as petal length and petal width.
- Histograms that display the distribution of each feature across different species.
- Correlation Heatmaps to highlight relationships between numerical variables.

These visualizations give insights into the most important features for classifying species and reveal correlations that might be useful for feature selection.



```
################ Multi_Classification problem ####################
There are 1 duplicate rows in your dataset
    Alert: Dropping duplicate rows can sometimes cause your column data types to change to object!
    All variables classified into correct types.
```

	Data Type	Missing Values%	Unique Values%	Minimum Value	Maximum Value	DQ Issue
sepal_length	float64	0.000000	NA	4.300000	7.900000	No issue
sepal_width	float64	0.000000	NA	2.000000	4.400000	Column has 4 outliers greater than upper bound (4.05) or lower than lower bound(2.05). Cap them or remove them.
petal_length	float64	0.000000	NA	1.000000	6.900000	Column has a high correlation with ['sepal_length']. Consider dropping one of them.
petal_width	float64	0.000000	NA	0.100000	2.500000	Column has a high correlation with ['sepal_length', 'petal_length']. Consider dropping one of them.
petal_area	float64	0.000000	NA	0.110000	15.870000	Column has a high correlation with ['sepal_length', 'petal_length', 'petal_width']. Consider dropping one of them.
species	object	0.000000	2			Target column

```
Total Number of Scatter Plots = 15
All Plots done
Time to run AutoViz = 8 seconds

 ###################### AUTO VISUALIZATION Completed ######################
```

Figure 1 Auto EDA using AutoViz Summary

Summary of the Dataframe

Using AutoEDA tools like AutoViz and Sweetviz, we summarized the dataset. Key takeaways include:

- Shape: The dataset contains 150 samples and 4 features.
- No Missing Data: All samples are complete, so no imputation was necessary.
- Distributions: Features such as petal length and petal width have clear differences between species, making them highly valuable for classification.

This summary helps guide model selection and feature engineering.

Auto EDA using Sweetviz

Sweetviz is another automated tool for exploratory data analysis, generating detailed reports with visual summaries of each feature. For the Iris dataset, Sweetviz generated insights such as:

- Distributions of Sepal Length/Width and Petal Length/Width: The graphs clearly show how petal-related features differentiate the species.
- Target Variable Distribution: The report highlighted the balanced nature of the target variable, with 50 samples for each species.

This comprehensive report helps us quickly assess the data and make informed preprocessing decisions.

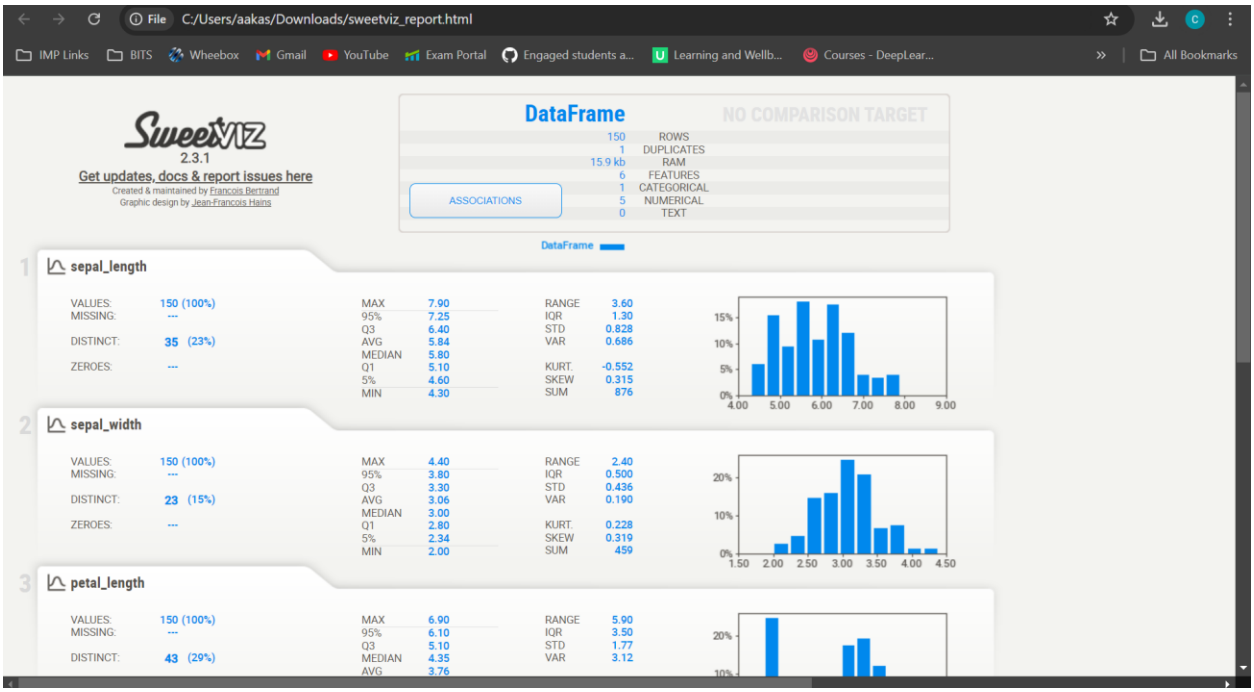


Figure 2 Auto EDA using SweetViz Summary

Summary of the Dataframe

The Sweetviz report provided detailed statistics for each feature, including:

- Sepal Length: Normally distributed, with a range of 4.3 to 7.9 cm.
- Sepal Width: Slightly skewed, ranging from 2.0 to 4.4 cm.
- Petal Length and Width: Both features exhibit significant differences across species, confirming their importance in classification tasks.

GitHub Link: https://github.com/AakashChaudhari03/MLOPS_ASSIGNMENT_2_GRP_NO_76