Sean Andre Membrido
CSE 40 - Spring 2024

Hands-On Assignment 5

★ Introduction

I have a dataset consisting of 13 columns, including 1 label and 12 data-specific columns. The label serves as the response variable, while columns col_00 to col_11 are the features. The dataset includes a mix of data types, such as numeric (integer and float) and categorical values. For the shape, the data set is exactly 1484 rows x 13 columns.

Proposed Models

a) Logistic Regression

This one is a suitable choice for this dataset for binary classification tasks, as it can help predict categorical outcomes. Given the label column, logistic regression could provide more factors that influence the classification.

b) Random Forest Classifier

This one is known for being able to handle both numeric and categorical data. It can capture the complex relationships within the dataset and helps identify what features are important, and what aren't

c) Support Vector Classifier

This one is suitable for both classification and regression asks, as it can handle, again both numeric and categorical features. However, as opposed to the RFC, this captures complex decision boundaries, making it versatile for this dataset.

Understanding & Goals

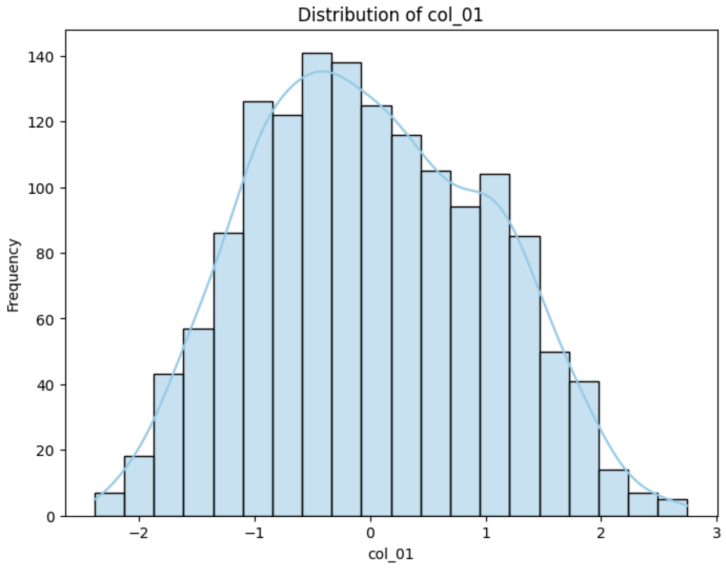
Exploring this dataset involves understanding the meaning and importance of each column. By identifying patterns and anomalies, we can classify the data for analysis. The aim is to use this information to build models that can predict the label column based on the features (the remaining columns). The following sections will cover the steps taken for data exploration, cleaning, and visualization, along with modeling and analysis.

★ Data Cleaning

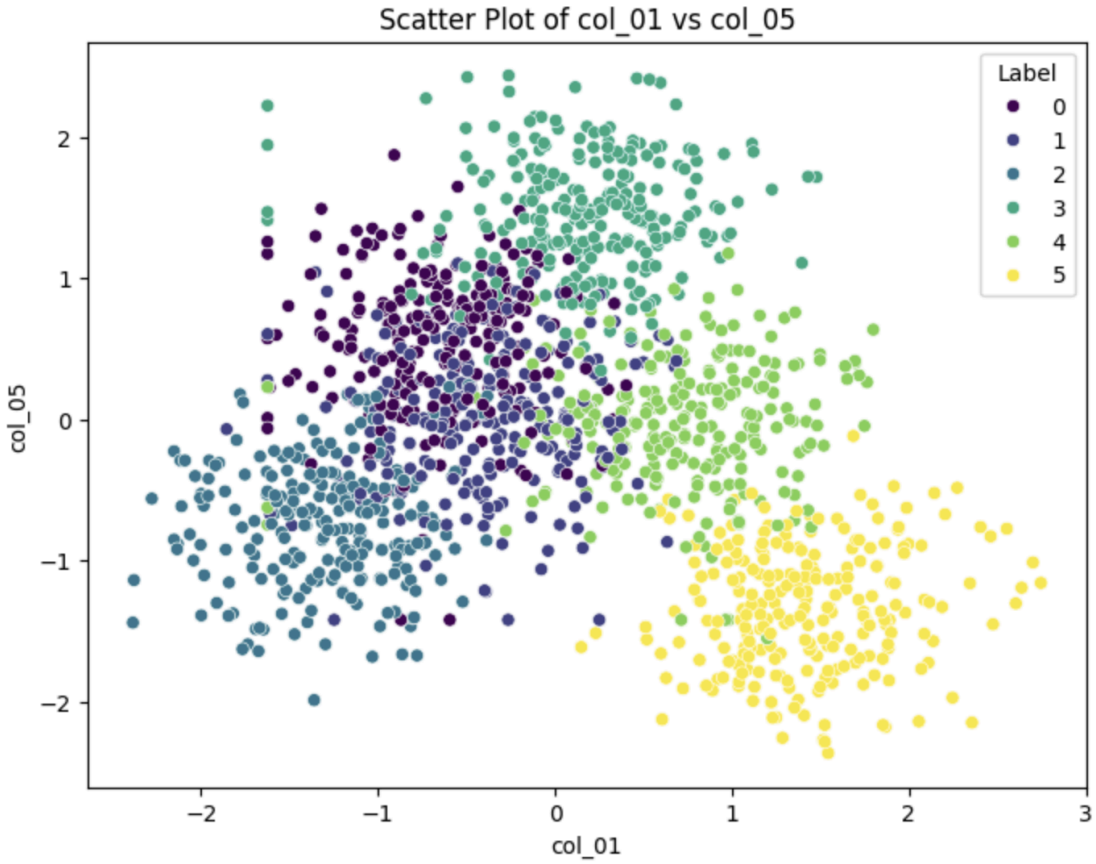
To clean the dataset, I first created a copy of the original data to preserve its integrity. I then applied a function to each cell, which extracted numeric values from strings and converted them to either integers or floats. If a cell did not contain a string or had no numeric value, it was set to 0. After cleaning the cells, I inferred the appropriate data types for each column, converting columns with numeric values to the correct numeric type (integer or float) and leaving other columns as strings. This process ensured that the data was properly formatted and ready for analysis.

★ Data Visualization

For the histogram of col_01, additional data preparation steps were necessary beyond the initial cleaning. First, we filled any remaining missing values in col_01 with the median of the column to ensure completeness. Next, we standardized the data using StandardScaler to transform col_01 to have a mean of 0 and a standard deviation of 1. This scaling helps in comparing the distribution of values more effectively. The histogram, enhanced with a kernel density estimate (KDE) line, reveals the frequency distribution of col_01. It indicates that the data is slightly skewed to the right, suggesting that most values are clustered around lower numbers with fewer occurrences of higher values. This insight is crucial for understanding the overall distribution and potential outliers in the dataset.



For the scatter plot of col_01 versus col_05, additional preparation included ensuring both columns were numeric and standardized. Missing values were filled with the median for both columns, and StandardScaler was applied to ensure they were on the same scale. This preprocessing is essential for accurately visualizing the relationship between the two variables. The scatter plot uses different colors to represent the labels, helping to distinguish between different classes within the data. This visualization highlights potential correlations or patterns, showing how different classes are distributed across the features. For instance, it can help identify clusters or trends indicating a relationship between col_01 and col_05, providing valuable insights for further analysis and modeling.



Please note that you won't find these in my jupyter notebook, since the autograder was unable to function with these being generated.

★ Modeling

I used three classifiers to explore and evaluate their performance on the dataset. The

chosen classifiers were Logistic Regression, Random Forest Classifier, Support Vector Classifier (SVC).

1) `sklearn.linear_model.LinearRegression()`
    a) **Description**: This is a model often used for binary classification problems. It models probabilities that a value belongs to a particular class.
    b) **Parameters**: No special parameters were used, and it resulted in both the highest mean accuracy and lowest variability.
2) `sklearn.ensemble.RandomForestClassifier()`
    a) **Description**: This is a model often used for decision trees that output the mode of the classes as the prediction.
    b) **Parameters**: No special parameters were used,
3) `sklearn.svm.SVC()`
    a) **Description**: This is a powerful classifier that checks off both binary and multiclass classification. It finds the decision boundaries between the classes, specifically a hyperplane.
    b) **Parameters**: No special parameters were used

Model	**Mean Accuracy, Standard Deviation of Accuracy**
Logistic Regression	0.9371, 0.0002
RandomForestClassifier	0.8870, 0.0087
Support Vector Class.	0.9370, 0.00018

1) **LR vs RFC (True)**

The p-val < 0.1, suggesting enough evidence to reject the null hypothesis. This makes it statistically significant at the 0.10 significance level.

2) LR vs SVC (False)

The p-val > 0.1, indicates that there's not enough evidence to reject the null hypothesis. This makes it not statistically significant.

3) RFC vs SVC (True)

Again, following the previous comparison, the p-val < 0.1, indicates enough evidence to reject the null hypothesis. Therefore, it's statistically significant at the 0.10 significance level.

Summary

The significance test implies that there are significant differences in accuracy between RFC and LR/SVC. RFC needs to be further explored and tuned to enhance its predictive capabilities. The chosen classifiers have their specific strengths as per their descriptions above. LR stands out with the highest accuracy, however, there is likely an issue regarding the overfit of training data. As for standard deviations, we get insights into the variability of accuracy scores. The lower values indicate more stable and consistent model performance. LR has a high accuracy with a low variability. RFC is unstable in comparison to the other two classifiers.

★ Analysis

Classifier Performance

The results suggest different variations in the performance of each of the classifiers, all improving upon each other in terms of accuracy and variability. The superior performance from Logistic Regression is probably noted by its capture of complex relationships in the data, and handling it in non-linear patterns. The reasons for the likely misrepresented accuracy may be due to overfitting, data leakage, or parameter tuning. Overfitting occurs when it learns random irrelevant data in the set, therefore performing very well. Parameter tuning, such as adjusting the maximum depth of trees or minimum samples can mitigate the first issue of overfitting data.

Issues with Cleaning Data

The high standard deviation we received in Random Forest Classifier could indicate outliers or features with high variance. This may indicate that the sparsity threshold may need to be modified individually for each column. Further investigation into which features matter more or the distribution could help us tune the arguments to the helper functions for clean_data.

Classifier Tuning

Adjusting parameters such as the regularization strength in Logistic Regression or the choice of kernel in SVC can enhance performance. For Logistic Regression, we need to balance effective training and avoiding overfitting. In SVM, we can choose from polynomial, linear, radial, and other kernels to find the most accurate one. Grid Search can be employed to explore all possible parameter combinations for each model. Alternatively, Random Search, which generates random configurations, often performs better in high-dimensional datasets, like ours, which includes one-hot encoded values for four categorical variables.

★ Conclusion

In summary, this project effectively examines and models the dataset, laying the groundwork for making predictions. Each part of the project builds on the previous one. The cleaned data enables modifications and scaling, allowing for successful data visualization through histograms and scatter plots. We successfully visualized the float columns, gaining insights into data distribution and the correlations between inverse relationships among similar data types. The analysis highlighted the strengths and weaknesses of the classifiers used and suggested further investigation into individual parameters and techniques to improve accuracy.

★ References

[1] McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media.

[2] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

[3] van der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient Numerical Computation. Computing in Science & Engineering, 13(2), 22-30.