Hana Jahangiri

December 3, 2024

Lost Metadata Data Analysis Report

1. Introduction

The given data provides records that have been set in different sports along with their associated details such as the city name and what I believe is the weight of the athlete. The dataset consists of 13 columns, 12 feature columns and one label column, each with varying data types. The label column is numeric while the 12 feature columns consist of information such as weights, sports, and locations, therefore varying from categorical to numerical data types.

The goal of this project is to clean and analyze the given data in order for it to be used for predictive modeling. The information in the feature columns should be used to predict the label of the response variable. The first model that comes to mind as being suitable for this project is a linear model, more specifically a Logistic Regression. Other models that may be suitable are a Decision Tree model and a Linear Support Vector Machine (SVM).

2. Data Cleaning

The first step I took for cleaning the data was to remove any rows with an empty label column. This is because the label column is the response variable and if this is empty then these rows are incomplete and cannot be used for predictive modeling.

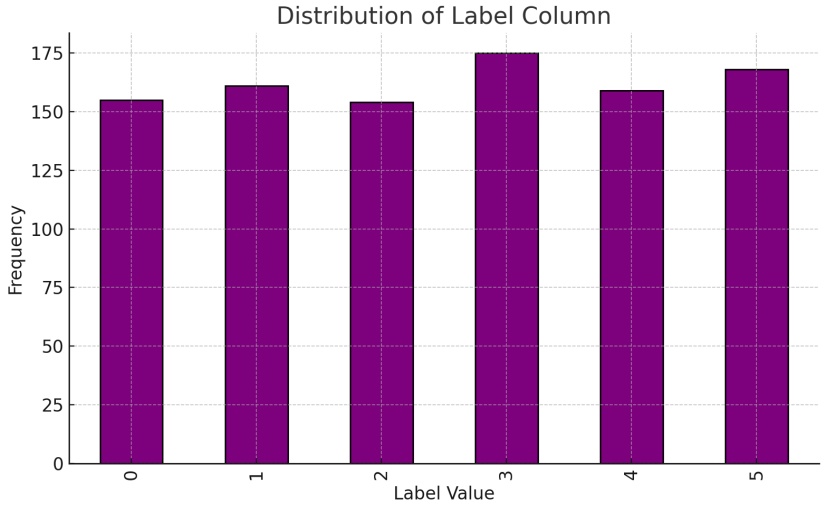
From here, I also addressed missing values in the feature columns. For all the feature columns I replaced the missing values with the placeholder “NA” using pandas.NA in order to make sure that the missing data was all uniform.

Next, I formatted the categorical feature columns by making sure that there were no extra leading or trailing whitespaces, the words began with a capital letter and the rest of the letters were lowercase, and that the values were all being stored as a string.

I then formatted the numerical feature columns, which by far caused me the most difficulty. I made sure that all numeric values that were stored as strings would be converted and stored as their respective numeric data type, whether it was an integer or a float.

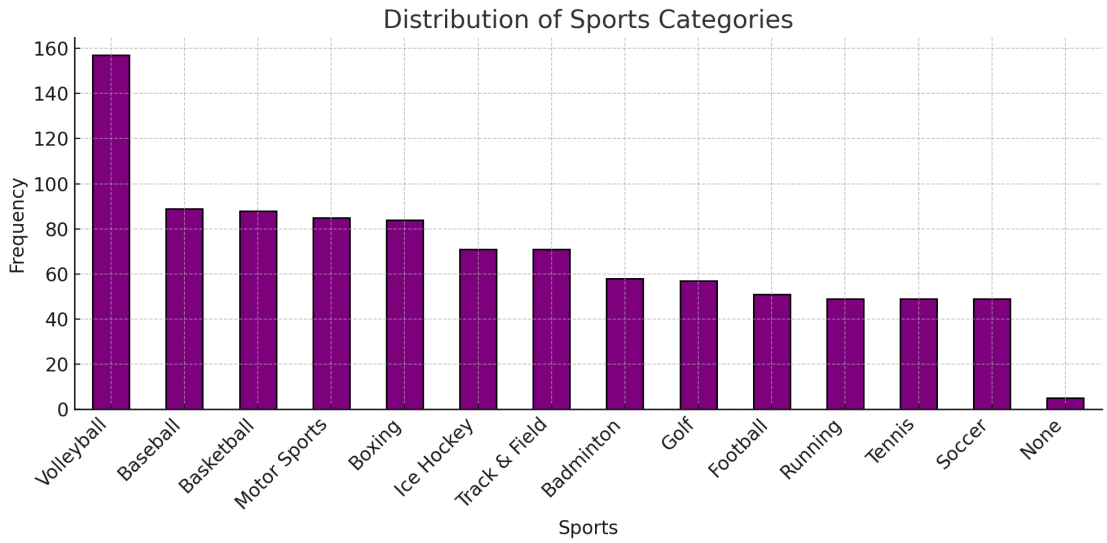
Last, I returned to this block of code and explicitly assigned the correct data types to each column, due to the fact that I was running into errors later in the assignment. I made sure that the string/categorical columns were assigned “object”, the float columns that had integer values were assigned “int64”, and the other float columns were assigned “float64”.

3. Data Visualization



This visualization was created in Python using the Pandas library, to count the frequency of each value in the label column and the Matplotlib library, to create the bar graph. The purpose of this visualization was to group the total occurrences of the six values that were present in the label column to provide their distribution. There was no additional data preparation needed to create

this visualization. This visualization shows us how the labels are distributed across the given dataset and we are able to see that it is relatively balanced, rather than a skewed distribution. This can be important for deciding what model to use in the next step of the assignment.



This visualization was created in Python using the Pandas library, to clean and count the frequency of each sport and the Matplotlib library, to create the bar graph. There was no additional data preparation needed to create this visualization. This visualization shows us the frequency of each sport across the dataset, as well as showing us how many rows did not have a sport provided. By analyzing the bar graph, volleyball is the most frequently occurring sport and the rest of the sports are generally not varying that much. Once again, this may help in deciding what models to use on this dataset.

4. Modeling

The first classifier I chose was the Logistic Regression Classifier. This linear model predicts the target variable by using a linear combination of the feature columns. I did not set any parameters, so the default parameter is max_iter = 100.

The next classifier I chose was the Decision Tree Classifier. This model is non-linear and predicts the target variable by creating a tree structure. I did not set any parameters, so once again the default parameters were used meaning that the tree will grow fully and split until all the leaves are either pure or contain less than min_samples_split = 2 samples.

The last classifier I chose was the Linear Support Vector Classifier. This linear model separates classes by finding a hyperplane with maximum margin [1]. I did not set any parameters, so the default parameters were used where kernel = 'rbf', since no specific kernel was specified.

Model Performance Results (5-Fold Validation)

Model	Mean Accuracy	Standard Deviation of Accuracy
Logistic Regression	0.9599	0.0088
Decision Tree	0.7798	0.0225
Linear SVC	0.9671	0.0089

The model that performed the best with the highest average accuracy of 96.71% with the smallest variation of 0.89% was the Linear Support Vector Classifier. The model that performed the worst with the lowest average accuracy of 77.98% and the largest variation of 2.25% was the Decision Tree. The clear winners in this case were The Linear SVC and the Logistic Regression models.

5. Analysis

Given that the Linear SVC and Logistic Regression Classifiers performed well, this indicates that a linear model is preferred for this dataset. Data that has a linear relationship between features and labels tends to work better with linear models, which allows us to make an

assumption that the given data set has this linear relationship. The Decision Tree model in this case was most likely overfitting for the data which led to its less consistent prediction accuracy.

In the case that the dataset had a skewed distribution of labels, then an F1-score would provide better insights in compassion to vanilla accuracy. However, given the first data visualization provided in this report, we know that the dataset has a balanced distribution of labels.

The data cleaning done in the first task was able to address all major inconsistencies in the given dataset, therefore, there is nothing that I can think of on the top of my head that would further need to be fixed for the sake of this assignment.

The differences in statistical significance between the three classifiers make sense given the distribution of the provided data aligning with linear models, and the two top performing classifiers were both linear classifiers.

Given that I did not adjust any parameters for the classifiers, the parameters can definitely be tweaked to provide better performance. For example, I believe that if the max_depth for the Decision Tree is adjusted, it may allow for it to provide more accurate predictions, regardless of it being a non-linear classifier.

6. Conclusion

The Linear Support Vector Classifier proved to have the best performance with a high accuracy and consistency in predictions. The Logistic Regression also was a reliable model to use on the given dataset.

Some areas for further improvement would be the implementation of the Decision Tree. It did not perform as well, however, if the parameters were adjusted accordingly and with

precision there is a likelihood that this model would be able to provide higher accuracy than it did in this specific instance.

Overall, I believe that this assignment demonstrated the importance of choosing the proper model for a dataset and in my opinion, emphasized the significance of properly analyzing the data before deciding on a model.

7. References

[1] Scikit-learn Developers. (n.d.). Support Vector Machines (SVM) — Scikit-learn Documentation. https://scikit-learn.org/1.5/api/sklearn.svm.html.

[2] Step-by-Step Data Science. (n.d.). Scikit-learn Tutorial: Python Machine Learning. https://www.stepbystepdatascience.com/scikit-learn-tutorial-python-machine-learning.

Extra Credit

1. Introduction

The dataset that I have chosen for this portion of the assignment is the Iris flower dataset provided by the UC Irvine Machine Learning Repository. The Iris dataset has 150 rows, which each represent a different iris flower. It has four numerical feature columns which are the sepal length, sepal width, petal length, and petal width. The target variable in this dataset is the specific species of iris flower, the three species being Iris setosa, Iris versicolor, and Iris virginica.

The goal of this project is to clean and analyze the given data in order for it to be used for predictive modeling. We have to predict which of the three Iris flower species the given iris flower is based on the four feature columns provided.

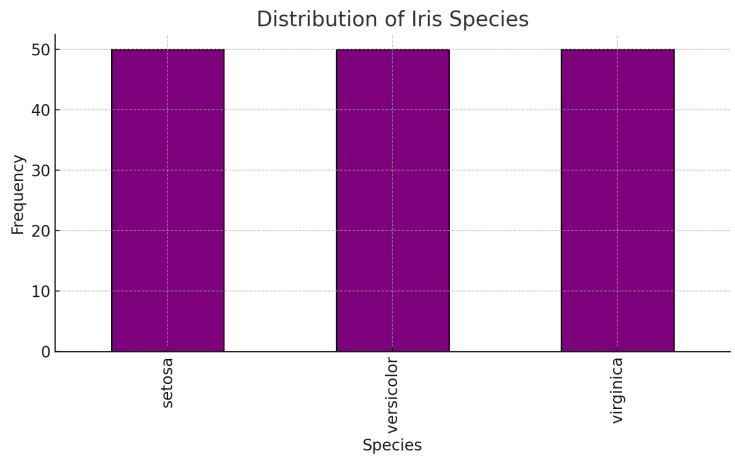
2. Data Cleaning

This dataset is much easier to work with than the previous dataset given that there are no missing values. Therefore, I was able to skip some steps that were previously implemented in the data cleaning process.

The first step I took was to set the numerical feature columns to have a mean of 0 and a standard deviation of 1, in order to make sure that all the features have equal weight.

Next, I adjusted the categorical target variables so that they would be numerical instead. I set Iris setosa to 0, Iris versicolor to 1, and Iris virginica to 2. This is because machine learning models require the target variable to be numerical.

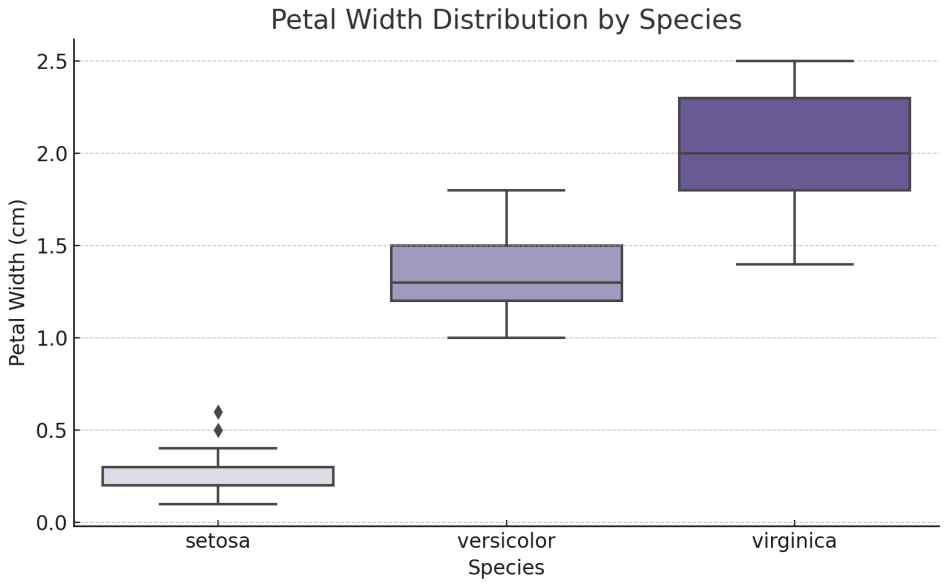
3. Data Visualization



This visualization was created in Python using the Pandas library, to count the frequency of the three species in the dataset and the Matplotlib library, to create the bar graph. The purpose is to see the distribution of the three species across the dataset. There was no additional data preparation needed for this visualization. We are able to see that the dataset provides us with a perfect equal balance of the three species, which is essentially ideal for training our models.

This visualization was created in Python using the Pandas library, to display the distribution of the petal lengths and the Matplotlib library, to create the histogram. There was no additional data preparation needed for this visualization. This visualization shows us that the petal lengths are

not normally distributed or evenly distributed. Therefore, I believe that it would be beneficial to create another graph to visualize the petal lengths or widths in relation to their species of Iris to see if there is a clear relationship or not.



This visualization was created in Python using the Seaborn library, to display a box plot of the petal widths grouped by their specific Iris species. There was no additional data preparation needed for this visualization. This visualization shows how each species of Iris tends to have an associated range of petal width. From the sertosa species generally having a smaller petal width to the virginica species having a generally larger petal width. This is a notable relationship that provides us with useful information regarding the size of an Iris and its predicted Iris species.

4. Modeling

The first classifier I chose was the Logistic Regression Classifier. This linear model predicts the target variable by using a linear combination of the feature columns. I did not set any parameters, so the default parameter is max_iter = 100.

The next classifier I chose was the Decision Tree Classifier. This model is non-linear and predicts the target variable by creating a tree structure. I did not set any parameters, so once again the default parameters were used meaning that the tree will grow fully and split until all the leaves are either pure or contain less than min_samples_split = 2 samples.

The last classifier I chose was the Linear Support Vector Classifier. This linear model separates classes by finding a hyperplane with maximum margin [2]. I did not set any parameters, so the default parameters were used where kernel = 'rbf', since no specific kernel was specified.

Model Performance Results (5-Fold Validation)

Model	Mean Accuracy	Standard Deviation of Accuracy
Logistic Regression	0.973	0.015
Decision Tree	0.960	0.022
Linear SVC	0.980	0.012

All of the models in this case performed well. The model that performed the best with the highest average accuracy of 98% and with the smallest variation of 1.2% was the Linear Support Vector Classifier. There is no significant difference between the performance of these three classifiers, however, the two linear models had a lower standard deviation and were therefore slightly more consistent in their predictions.

5. Analysis

The high accuracy in predictions across all three models indicates to us that the feature columns provided values that were significantly telling of the specific Iris species. The two feature columns that were the most significant were the petal length and the petal width, because

essentially the three Iris species have very distinct sizes. The third visualization I attached earlier displays the indication of this strong relationship as well.

The cleanliness and the simplicity of the chosen dataset is another reason for the straightforward results and relationship between the feature values and the species.

6. Conclusion

Overall, the dataset was ideal for the sake of implementing different classifiers and I was able to achieve a high accuracy in predictions across all three chosen classifiers.

Some possible areas for further improvement could be exploring more complex models given that the data is simple and easy to understand. Playing around with complex models will allow me to further my understanding while not leading me to unnecessary roadblocks due to the nature of the dataset.

7. References

[1] UCI Machine Learning Repository: Iris. https://archive.ics.uci.edu/ml/datasets/iris

[2] Scikit-learn Developers. (n.d.). Support Vector Machines (SVM) — Scikit-learn Documentation. https://scikit-learn.org/1.5/api/sklearn.svm.html.