**Avila Bible**

Jacklyn Tellez, [jtellez2@bellarmine.edu](mailto:jtellez2@bellarmine.edu)

Hunter Waldridge, [hwaldridge@bellarmine.edu](mailto:hwaldridge@bellarmine.edu)

**ABSTRACT**

For this project, the Avila data set was used. The data from this set was drawn from 800 images of the Avila Bible. An exploratory analysis was performed to clean and better understand the data set. The two machine learning models used in this project to predict the classes were K-Nearest Neighbor and Decision Tree. Each model was running four times under four different experiment parameters. In this data set, the classes were twelve copyist who wrote pages of the Avila Bible.

1. **INTRODUCTION**

The classification target of the data set is the copyists. Before printers, individuals would have to copy pages of books. Our data set is for trying to predict the copyists of the Avila Bible. Since each copyist had their own way of writing, our data set looks at ten features to help indicate which copyist is which. The copyists are not given names, but are identified with the letters A, B, C, D, E, F, G, H, I, W, X, and Y. The ten features used to make the predictions are intercolumn distance, upper margin, lower margin, exploitation, row number, modular ratio, interlinear spacing, weight, peak number, and modular ratio divided by interlinear spacing.

1. **BACKGROUND**

The Avila Bible was created before the time of printers, so its creation required a group of copyist to work together to write the book (C. De Stefano). A pattern recognition system was used to determine the patterns of the images from the Avila Bible to determine the different copyist. It distinguished between the twelve copyist who worked on one book together by using different features that when analyzed are unique to a certain copyist. The program was used on 800 images of the Avila bible that was created in the XII century in Italy and Spain.

1. **EXPLORATORY ANALYSIS**

This data set had both a training set and a test set. The training test had 10430 samples with 11 columns. The test set had 10437 samples with 11 columns. The 11 columns in both sets are the same and the types for each can be found in Table 1: Data Types. The standard deviation for upper margin was unusually high for a data set that should have been normalized. There was one data point causing this high standard deviation and we removed the entire row. A couple of the statistics for Intercolumn distance were negative. We found this odd because we did not know how column distance could be negative. When assessing our data for missing data, not a single row had any missing data. We found this unusual, especially since it was such a big data set. We looked at the data specifically for a class. When we went to make box plot for class B, we noticed that its data for the first five columns are exactly the same in each row.

**Table 1: Data Types**

|  |  |
| --- | --- |
| *Variable Name* | *Data Type* |
| Intercolumnar distance | Float 64 |
| Upper Margin | Float 64 |
| Lower Margin | Float 64 |
| Exploitation | Float 64 |
| Row number | Float 64 |
| Modular ratio | Float 64 |
| Interlinear spacing | Float 64 |
| Weight | Float 64 |
| Peak number | Float 64 |
| modular ratio/ interlinear spacing | Float 64 |
| Class | Object |

1. **METHODS**

In this section, describe how you prepared the data for your model and performed multiple experiments using different parameters for the model(s).

* 1. *Data Preparation*

Our data set came already normalized. It was normalized using the z-normalization method. This method is the process of normalizing the data set so that all elements of the data set have a mean of 0 and a standard deviation of zero. We removed one row from the data set because it contained a major outlier for the upper margin column. The majority of the data points for the column upper margin ranged from just below zero to under fifty, but the outlier was just below 400. This can be seen in figure 1. Between adding the two sets together and removing a row, this caused the means and standard deviations to be slightly off from 0 and 1, respectively.

* 1. *Experimental Design*

Since our data was already normalized, we were unable to perform any experiments with the raw data. We had four experiment parameters that were each ran for the models K-nearest neighbor and decision tree.

Table 2: Experiment Parameters

|  |  |
| --- | --- |
| **Experiment Number** | **Parameters** |
| 1 | All ten (10) normalized features but only two of the classes with a 70/15/15 split for train, test, and validate |
| 2 | All ten (10) normalized features but only two of the classes with an 80/10/10 split for train, test, and validate |
| 3 | All ten (10) normalized features with an 80/10/10 split for train, test, validate |
| 4 | Five (5) normalized features with an 70/15/15 split for train, test, validate |

* 1. *Tools Used*

Describe all of the software tools you used to perform your data preparation and model implementation. For example:

The following tools were used for this analysis: Python v3.8.8 using Anaconda version 2.1.1 for Apple Macintosh computer was used for all analysis and implementation. In addition to base Python, the following libraries were also used: Pandas, Numpy, Matplotlib, Seaborn, SKLearn, Random. Pandas was used for graphing purposes during the exploratory analysis as well as bringing in the datasets. Numpy was used to check for missing data and for the correlation matrix visualization. Seaborn was used for the heatmap and confusion matrices. SKLearn was used for the machine learning models. Random was used to help create a new dataset using 5 of the original dataset’s columns.

1. **RESULTS**
   1. *Classification Measures*

**K-Nearest Neighbors**

**Table 3: Classification Report of Experiment 1**

Graphical user interface, application

Description automatically generated

**Table 4: Classification Report Experiment 2**Graphical user interface, text, application

Description automatically generated

**Table 5: Classification Report Experiment 3**

Graphical user interface, application, calendar

Description automatically generated

**Table 6: Classification Report Experiment 4**

Graphical user interface, application

Description automatically generated

**Decision Trees**

**Table 7: Classification Report Experiment 1**

Graphical user interface, application

Description automatically generated

**Table 8:** **Classification Report Experiment 2**

Graphical user interface, application

Description automatically generated

**Table 9: Classification Report Experiment 3**

Graphical user interface, application

Description automatically generated

**Table 10: Classification Report Experiment 4**

Graphical user interface, application, table

Description automatically generated with medium confidence

To determine the success and failure of models we looked at the accuracy from each classification. The Accuracy Scores in the nineties (.999 to .900) we considered successful. The scores in the eighties we considered somewhat successful. Anything below eighty we considered failure.

* 1. *Discussion of Results*

The Decision tree model provided the best classification. The accuracy levels were always higher, and it contained the highest accuracy levels overall. Decision Tree experiments 1, 2, and 3 were successful. K-Nearest Neighbor experiments 1 and 2 were somewhat successful. K-Nearest Neighbor experiments 3 and 4 and Decision Tree experiment 4 were failures. For both KNN and Decision Tree, experiments 1 and 2 produced better results than 3 and 4. Experiments 4 had the overall worst results. Experiments 1 and 2 most likely produced better results because only two classes with similar counts were used. Experiments 1 and 2 differed because of the split values used. When an 80/10/10 split was used, better results were produced. Experiment 4 most likely failed because the number of features used was limited.

* 1. *Comparison of Models*

Overall Decision Tree was the better model. Comparing the accuracy of each experiment between the two models, decision tree always had the higher accuracy. It was also the easier of the two to code. K-nearest neighbor would probably run better with a smaller data set and a data set with a more even spread of classes.

* 1. *Problems Encountered*

In preforming K-Nearest Neighbors, we ran into a problem when coming up with the k value. Multiple sites and videos suggested using the square root of the number of rows in our data set. Since the number of rows was over twenty thousand, that left k being around 149, but we had three classes with a count of less than that. We ended up using a much smaller k value in order to produce better results. When k of 149 was used the three classes were never predicted. This is why we ran a test with only two of the classes. We chose two classes with higher counts.

Our data set came with both a training and test set, we got confused when it came time to split the data. We did not know if we used all of the data or just one of the sets.

* 1. *Limitations of Implementation*

Our data set was very large. In our research of the different Machine Learning Models, a few of them were most accurate when using small data sets. The counts of the classes were not evenly distributed, which lead to some inaccuracies.

* 1. *Improvements/Future Work*

In the future, finding a different data set to work with would be beneficial. A data set with a more even distribution of classes would be beneficial to the K-Nearest Neighbor model. We chose models that we thought would work best with our data set, but a few of the other models seemed interesting, so finding data sets that would work better with those models would be constructive in the future. Finding a data set that has the raw data in it would be educational. It would allow a comparison of normalized and raw data results.

1. **CONCLUSION**

We used the machine learning models of K-Nearest Neighbor and Decision Tree to run test to predict the copyists of the Avila Bible. For each model, we ran four experiments that differed in split sizes, amount of classes, and amount of features. The Decision Tree Model produced the best accuracy score for each of the four experiments and was the easiest to code. It was the better and easier model of the two. Experiment four was the worst experiment ran in each model because we limited the feature size. In the future, a data set with a more evenly dispersed class counts would show better accuracy scores.

**REFERENCES**

List any websites, books, articles, etc. that you found useful while you worked on this project. It is not necessary to cite the references in the paper unless you specifically mention it in the text.

Starmer, Josh. “StatQuest: K-nearest neighbors, Clearly Explained.” *YouTube,* uploaded by StatQuest, 26 June 2017. <https://www.youtube.com/watch?v=HVXime0nQeI>

“KNN Algorithm In Machine Learning | KNN Algorithm Using Python | K Nearest Neighbor | Simplilearn.” *YouTube,* uploaded by Simplilearn, 6 June 2018. <https://www.youtube.com/watch?v=4HKqjENq9OU&t=408s>

“Decision Tree In Machine Learning | Decision Tree Algorithm In Python |Machine Learning |Simplilearn.” *YouTube,* uploaded by Simplilearn, 20 March 2018. <https://www.youtube.com/watch?v=RmajweUFKvM&t=182s>

“Decision Tree Classification Clearly Explained!” *YouTube,* uploaded by Normalized Nerd, 13 Jan. 2021. <https://www.youtube.com/watch?v=ZVR2Way4nwQ>

“Machine Learning Tutorial Python - 9 Decision Tree.” *YouTube,* uploaded by codebasics, 16 Nov. 2018. <https://youtu.be/PHxYNGo8NcI>

C. De Stefano, M. Maniaci, F. Fontanella, A. Scotto di Freca,

Reliable writer identification in medieval manuscripts through page layout features: The "Avila" Bible case, Engineering Applications of Artificial Intelligence, Volume 72, 2018, pp. 99-110.

*Division of Labor*

Both of us met on Teams to work on the project together at the same time. We also did some independent research on the topics to help us better understand them. Hunter did most of the coding, sharing his screen so Jacklyn could see and make any suggestions for what to do with the data throughout the exploratory analysis and machine learning process. Jacklyn also worked on the report and helped Hunter with some of the coding he struggled with. While Hunter coded, he provided the results and outputs of the code through Teams that needed to be included in the report. We both worked on the presentation together by brainstorming how we can organize the slides. We also both reviewed each other's work and provided suggestions when needed.

**Other directions:**

1. 10-pt, Times New Roman, 1” margins all around (if you use this template you are already set).
2. **Ensure all tables and figures are numbered appropriately and referenced in the text.** See examples above and below.

**APPENDIX:**

**Tables:**

Graphical user interface, application

Description automatically generated

Figure 1: Boxplot of the column Upper Margin with the Outlier

Graphical user interface, application

Description automatically generated

Figure 2: Bar Graph of the counts of each class

<https://nbviewer.org/github/m-pana/avila/blob/master/A%20digital%20approach%20to%20palaeography.ipynb>