**Finding Sick Trees with Photos**

Naat Ambrosino, [nambrosino@bellarmine.edu](mailto:nambrosino@bellarmine.edu)

Jerrin Redmon, [jredmon3@bellarmine.edu](mailto:jredmon3@bellarmine.edu)

**ABSTRACT**

Up to 150 word summary of your project.

This project contains an exploratory analysis and implementation of machine learning algorithms for the Wilt Data Set from the UCI Machine Learning Repository. This data set contain information of image data for use in a program that determines if a tree is diseases or healthy. In this project we will go through the Wilt dataset to run machine learning to test the accuracy of the program. With this we can learn how these machine learning principle play out while also performing an exploratory analysis on the data set as well. The end goal is observing how well the algorithms performed on the data set and bring insight from it.

1. **INTRODUCTION**

Provide a one or two paragraph introduction to your project in which you describe the data set you are working with and the classification target (what are you trying to predict?) you will be pursuing.

We are performing an analysis of the Wilt Data Set from UCI Machine Learning Repository. This dataset contains data of a sensor from a camera used to detect areas of a forest in order to detect and find diseased trees. The goal is to create a model that detects diseased trees from an aerial photograph (Using Quickbird Imagery) . We will also employ several data principles to perform an exploratory data analysis to find any information from the data set before we implement any of the machine learning techniques.

1. **BACKGROUND**

In this section, provide some background for the problem for which the data were collected. For example, if you were using the mushroom data set, you write up some background on what a mushroom is, why the data were originally collected, what question(s) the authors were trying to answer, etc.

The Wilt Data Set from the UCI Machine Learning Repository; contained data from both training and testing of covered land and diseased trees. Trees are very important to this planet and keeping them safe and away from diseased is important in sustaining these plants. The goal was to help detect trees that are sick or healthy from photographs. From this data the problem at large was scientists wanted to be able to tell if a tree was sick from a birds eye view using GLCM for pan images. This extracts texture features from photos, in this case forests/trees.

**EXPLORATORY ANALYSIS**

This section will be similar to your exploratory analysis project. First, provide a summary of the data set similar to your first exploratory analysis: *e.g. this data set contains 398 samples with 7 columns with various data types*. In this summary, provide the data types of your columns (in a table) and then rather than providing tabular statistics and plots for each variable, provide only statistics and plots that seem unusual. For example, if one or two variables have significant missing values or the distribution of the variable is skewed or looks unusual note that. Provide the unusual statistics or plots in this section. Provide any other appropriate plots (e.g. correlation matrix, heatmaps, bar charts, etc.) that you deem necessary.

The author of the Wilt dataset split the data into two separate file and We decided to combine both set of data as the original to give us a data set that contains 4893 rows and 6 columns. The columns are split up into class which distinguishes if the tree was diseased or healthy. GLCM which is the pan brand image texture to distinguish the image data. The Green and Red Mean values which contain the mean of both types of color from the photographs. The Mean NIR is the value of the mean for near infrared images of the forest. The last is the SD\_Pan which is the standard deviation of the pan band data.

**Table 1: Data Types**

|  |  |
| --- | --- |
| *Variable Name* | *Data Type* |
| sick | Integer64 |
| GLCM\_pan | Float64 |
| Mean\_Green | Float64 |
| Mean\_Red | Float64 |
| Mean\_NIR | Float64 |
| SD\_pan | Float64 |

During the cleaning process we found the data to be very clean, but we chose to look through it to find any anomalies. We did several box plots of the data to find any outliers or anomalous data that may skew with the machine learning algorithms. From our plots we found some small outliers, mainly 1482, 1492, 1515, 2813 which can be seen from the plot below(Figure 1). However, we felt that these outliers were so small that they wouldn’t affect the machine learning in a negative way. However, there was one consistent outlier in both Mean\_Red and Mean\_Green at 479. This outlier was very far out and with careful consideration we choose to drop the outlier from the data so that our experiments can flow smoothly(Figure 2).

*Chart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generated*

**Figure 1: Boxplot of all the data without any changes. Figure 2: Boxplot of all the data without 479 outlier with the data having been normalized**

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*

Describe how you prepared the data for your model. For example, you might need to normalize the data, so variables with wider ranges of values don’t overshadow variables with smaller ranges. If you decide to drop variables from the model or create variables from existing columns, explain the process and the reasoning behind those decisions.

The process in preparing that dataset we choose to change several points in order to have the experiments work properly. First was to combine both the training and testing data sets since the author choose to split them. Then we drop the outliers we felt may cause issues(Mainly 479). Finally, was to change the class from a string to an integer of 0 and 1 to represent healthy and diseased as true or false. This was done for problems that were caused by the column names. Any other small adjustment was to run different types of tests for our models.

* 1. *Experimental Design*

You will run your model several times with different parameters to see what different results you get. In a table, describe your experimental parameters. Three or four experiments are sufficient. This is where you will describe how you divided your data into train, validate and test data sets. For example:

Table X: Experiment Parameters

|  |  |
| --- | --- |
| **Experiment Number** | **Parameters** |
| 1 | All Five (5) all raw features, 80/10/10 split for train, validate, and test(yA) |
| 2 | All Five (5) all raw features, 50/25/25 split for train, validate, and test(yA) |
| 3 | All Five (5) all raw features, 80/10/10 split for train, validate, and test(yB) |
| 4 | All Five (5) all raw features, 50/25/25 split for train, validate, and test(yB) |
| 5 | min\_samples\_leaf=2, f1-scores=[.99,.81,.98]. Going to increase until the f1 scores start getting worse. |
| 6 | All Five (5) normalized features with 70/15/15 split for train, validate, and test |
| 7 | All Five (5) normalized features with 50/25/25 split for train, validate, and test |
| 8 | dfA with noralized features, 80/10/10 split for train, validate, and test |

* 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 v3.5.2 running the Anaconda 4.3.22 environment for Apple Macintosh computer was used for all analysis and implementation. In addition to base Python, the following libraries were also used: Pandas 0.18.1, Numpy 1.11.3, Matplotlib 1.5.3, Seaborn 0.7.1, SKLearn 0.18.1, and Patsy 0.41. Provide a brief explanation of why you chose these tools.

The following tools used for this data analysis: Python v3.8.11 running the Anaconda 2021.11 for Window operating System. The following imports and libraries that were implemented: Pandas, Numpy, Scipy.stats, pylab, matplotlib, seaborn, copy, sklearn [Tree, ensemble, metrics, model\_selection], graphviz, mpl\_toolkits.

Pandas and Numpy was used for basic data manipulation and cleaning. Scipy, pylab, mathplotlib, and seaborn were used for the generation of plot and graphs to visualize the data. Copy was needed to make deep copies of data frames. Sklearn and its imports were used for both the decision tree model and random forest model. The model selection import was to split the data into train and test data sets. Mpl was used to graph our 3d plots, and metrics was used to calculate metrics of the model.

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

Provide the classification measures for each experiment. For example, you could provide a contingency table for each model to measure how well it classifies data. You could also do an ROC curve (using SciKit Learn). You need to demonstrate how you are measuring the success/failure of the models.

Decision Tree Contingency Table Results

precision recall f1-score support

all other land cover 0.98 0.99 0.99 453

diseased tree 0.88 0.74 0.81 31

accuracy 0.98 484

macro avg 0.93 0.87 0.90 484

weighted avg 0.98 0.98 0.98 484

precision recall f1-score support

all other land cover 0.99 0.99 0.99 453

diseased tree 0.86 0.81 0.83 31

accuracy 0.98 484

macro avg 0.92 0.90 0.91 484

weighted avg 0.98 0.98 0.98 484

precision recall f1-score support

all other land cover 0.99 1.00 0.99 453

diseased tree 0.93 0.87 0.90 31

accuracy 0.99 484

macro avg 0.96 0.93 0.95 484

weighted avg 0.99 0.99 0.99 484

precision recall f1-score support (Leaf 10)

all other land cover 0.99 1.00 0.99 453

diseased tree 0.93 0.87 0.90 31

accuracy 0.99 484

macro avg 0.96 0.93 0.95 484

weighted avg 0.99 0.99 0.99 484

**(Leaf 10 Provided the Best Results)**

precision recall f1-score support

all other land cover 0.98 0.99 0.99 639

diseased tree 0.86 0.71 0.78 45

accuracy 0.97 684

macro avg 0.92 0.85 0.88 684

weighted avg 0.97 0.97 0.97 684

precision recall f1-score support

all other land cover 0.99 0.98 0.98 642

diseased tree 0.70 0.83 0.76 42

accuracy 0.97 684

macro avg 0.84 0.90 0.87 684

weighted avg 0.97 0.97 0.97 684

Random Forest Contingency Table Results

precision recall f1-score support

all other land cover 0.99 1.00 0.99 689

diseased tree 0.94 0.81 0.87 37

accuracy 0.99 726

macro avg 0.96 0.90 0.93 726

weighted avg 0.99 0.99 0.99 726

precision recall f1-score support

all other land cover 0.99 1.00 0.99 689

diseased tree 0.97 0.81 0.88 37

accuracy 0.99 726

macro avg 0.98 0.90 0.94 726

weighted avg 0.99 0.99 0.99 726

precision recall f1-score support

all other land cover 0.99 1.00 0.99 689

diseased tree 0.91 0.84 0.87 37

accuracy 0.99 726

macro avg 0.95 0.92 0.93 726

weighted avg 0.99 0.99 0.99 726

precision recall f1-score support

all other land cover 0.99 1.00 1.00 463

diseased tree 0.95 0.86 0.90 21

accuracy 0.99 484

macro avg 0.97 0.93 0.95 484

weighted avg 0.99 0.99 0.99 484

* 1. *Discussion of Results*

Discuss which of your models provided the best classification (or some other outcome if not classification). Explain why you think your best model was the best and why your worst model was the worst.

The first of the results above are several results from the Decision Tree. Both the dfA set and the dfB set both gave amazing results, with an average f1-score of .99. The second set of results are from the Random Forest algorithm. We set an estimators to 100 which was default parameter, which gave us very good results. This also gave us results with an average of .99 as well. We also noticed after several tests that the random forest data has an overall lower recall and support score.

* 1. *Comparison of Models*

Since you are running at least two machine learning classification models, compare the models and explicitly discuss which model was the best.

After discussion of the results, we came to an agreement that the decision tree’s classification models performed much greater than the Random Forest classifications models. The precision, recall, f1-score, and support all had better scores and performance overall. This is mostly likely due to the algorithm fitting much better than the random forest which required a bit of data manipulation to work properly.

* 1. *Problems Encountered*

No project goes perfectly smooth. Discuss any problems you had with obtaining the data, preparing the data, implementing the model, or evaluating the model. **It would be highly unusual to indicate that you had no problems.**

One of the column names being ‘class’ caused problems. I also had problems with sns.boxplot for the dfSick data frame, with no such problem for any other data frames. Also, a problem that the confusion matrix function had where the y\_pred and y\_test should be flipped.

* 1. *Limitations of Implementation*

Discuss the limitations of your model(s). Are there reasons your models might not be the best way to predict the target data? What other models might work better?

During the process in choose which machine learning algorithm would be best for the data set, we have to ask which ones were compatible. For example, K-Nearest Neighbor would not work with this data set. We also experience difficulties implementing Naïve Bayes and support vector machine. A main reason this we only want to predict between 2 classes, so for example that would make SoftMax regression not possible. There may be other models that work better and possibly could have worked with different kinds of manipulation, but what we picked gave us the current and best results overall.

* 1. *Improvements/Future Work*

What would you like to do to improve your model in future work? Some items you might consider discussing are performing more experiments, using different models, adding, or removing variables, finding a different data set, etc.

If anything were to be improve it would be to look into more types of machine learning algorithms to see if we can result that not just super good. This will allow us to compare it to our current models so we can check and make sure if the numbers are correct. It never hurts to try to find worse results to check your work. Besides that, the current status of the work is green and all good.

1. **CONCLUSION**

Finish up with a paragraph or two of summarizing your problem, the results, and your conclusions (good model, bad model, needs more work, etc.).

The Wilt data set was made because people wanted to determine if trees were diseased via arial photographs. So, they made a program that helps detect these sick, where the data is different aspects and textures of the images. Then we employed an analysis of the data to observe the accuracy of the photos and the program detecting them. From our results we found that the data had incredible accuracy which high precision and accuracy. We wanted to make sure so we performed several tests and they all came out around the same with high numbers and great results. We conclude that the analysis went very well, but more types of test can be done to improve what is already in place. The models left us with mostly good data that assures that the scientists work in detecting diseased trees via arial photographs was mostly a success.

**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.

<https://ieeexplore.ieee.org/document/6599565>

*Division of Labor*

In this section after you references, provide a paragraph or two outlining what each team member did on this project. Please don’t tell you both did everything! Be honest about who did what.

The division of labor ended up as follows:

Naat made the notebook (data exploration, cleaning, selecting, and running tests) and the decisions regarding it and the data, made markdown comments analyzing results, provided explanation of concepts for the documentation, clarified directions, researched, did some proofreading/editing, and scheduled meetings.

Jerrin was in charge of this write up . Jerrin also made some edits to the notebook, mostly bolding some of the markdown comments for emphasis for the presentation.