DS110 Project Proposal

By: Kayla Wu, Xiang Fu

Template 2: Publicly Available Dataset and Machine Learning

Find a publicly available dataset (give a link in your proposal) that has some data you can use for the final project, and **decide on 2 approaches that you could take to predict some variable with machine learning.** (Approaches include k-nearest neighbors, decision trees, and random forests; you can also experiment with anything else you find in scikit-learn or elsewhere.) You don't need to implement the machine learning algorithm yourself - you can make use of scikit-learn and other libraries. **Identify the columns you will use for prediction.** You should also **plan to vary some parameters in each approach** to achieve the best possible performance - for example, vary k for k-nearest neighbors, or vary maximum depth for decision trees.

Chocolate Bar Ratings

Dataset and inputs (Data preparation)

- Description
    - Believe it or not, chocolate is ranked the most popular candy in the world. For our final project, we are interested in exploring expert ratings of over 1,700 individual chocolate bars from a Chocolate Bar Ratings dataset via Kaggle. In other words, we will be predicting chocolate bar ratings using the following parameters/columns: (1) regional origin, (2) percentage of cocoa, (3) variety of chocolate bean used, and (4) broad bean origin, to train our machine learning model.

Objectives of our dataset exploration

1. Estimate the chocolate bar ratings with the aforementioned parameters in the dataset
2. Determine which parameter(s) will have the greatest impact on chocolate bar ratings

Possible approaches

1. **Decision trees**
    a. We can split the dataset into small segments.
        i. This is where the tree is built and tested using the dataset.
        ii. For instance, Is the cocoa percent 75%? > (Y/N) > Does the bean type originate in Peru? (Y/N) > Was the broad bean grown in Venezuela? (Y/N).
2. **Random forests**
    a. This approach will allow us to determine which parameters are most important. Random forests provide a higher level of accuracy than decision trees.
    b. First, we will select random samples from the dataset, then construct a decision tree for each sample for a prediction result from each decision tree.
    c. Next, we will perform a majority vote for each predicted result.
    d. Finally, we will select the prediction result with the most votes as the final prediction.

List of all variables

- Company (Maker-if known)
- Specific Bean Origin or Bar Name
- Review Date
- **Cocoa Percent** (feature/parameter)
- Company Location
- Rating (outcome)
- **Bean Type** (feature/parameter)
- **Broad Bean Origin** (feature/parameter)

Chosen columns for chocolate bar rating prediction

(1) Cocoa Percent
(2) Bean type
(3) Broad Bean Origin
(4) Specific Bean Origin for Bar Name

Steps of Our Machine Learning Training Process

1. **Data cleaning**
    a. During this process, we will remove inaccurate, corrupted, incorrectly formatted, duplicated, or incomplete data from our dataset.
2. **Train the model**
    a. We will use our data to incrementally improve our model's ability to predict the ratings of chocolate
    b. Split the dataset into two sets: a training set and a testing set.
        i. Training data:
            1. 80% of the dataset
        ii. Testing data (Evaluation):
            1. 20% of the dataset
3. **Evaluate the model**
    a. Train the model on the training set.
    b. Test the model on the testing set and evaluate performance.
    c. For Decision tree:
        i. Run the trained model on the test data and see what it predicts.
            1. test_pred_decision_tree = clf.predict (test_x)
        ii. Visualize the result with confusion matrix (A way to express how many of a classifier's predictions were correct, and when incorrect, where the classifier got 'confused')
            1. confusion_matrix(y_test, y_pred_test)
            2. Can also use a Seaborn heatmap() to visualize the confusion matrix
    d. For Random Forest:
        i. We can use this an accuracy score to measure how many labels the model got right out of the total number of predictions
            1. accuracy_score(y_test, y_pred_test)
        ii. Scikit- Learnis classification_report()
            1. classification_report (y_test, y_pred_test)
4. **Parameter Tuning**
    a. In this step, we will tune model parameters to improve the performance
    b. Model hyperparameters to consider
        i. Decision Tree: random state, maximum depth, minimum samples leaf
        ii. Random Forest: max_features (maximum number of features Random Forest is allowed to try in individual tree), max_depth_, min_ sample_leaf, n_estimators (the number of trees to build before taking the maximum voting or averages of predictions), random_state