**Project 4: Detecting Poisonous Mushrooms**

**CS6230 AI Tools**

**Learning Outcomes**

1. Identify, access, load, and prepare (clean) a data set for a given problem.

2. Train, apply, and evaluate supervised machine learning models.

3. Interpret machine learning models to identify predictive features.

4. Communicate findings through generated data visualizations and reports.

**Overview**

Machine learning has a wide variety of applications, many of which are used to provide advanced functionality to products for end users. For example, ML models are used to estimate digital ad click-through probabilities and optimize ad targeting, generate product recommendations, or offer suggested edits and completions when composing messages. In many of these contexts, we are only concerned with the accuracy the predictions from a machine learning (ML) model, not how a specific model may make those predictions.

There are cases where we do need to be able to explain how a machine learning model came to particular decision. For example, when a bank denies a credit card applicant or a request to increase a credit limit, they are legally required to provide the customer with specific reasons that led to that decision. Common reasons include: the customer is using too much of their credit, the customer's amount of debt is too high relative to their income, or they were recently late on one or more payments. Most machine learning algorithms are not designed to allow the rules they learn to be easily translated into human-understandable interpretations, however. These models are designed to be used as "black boxes."

In this project, we are going to explore two models that are amenable to human interpretation: Decision Trees and Logistic Regression. You are going to use a dataset of mushroom (fungi) properties (e.g., color and odor) to predict whether the mushrooms are poisonous or editable. Unlike other plants, there are no simple rules for determining the edibility of a mushroom – each species is different, so the naturalist must often first determine the species and then cross-reference a list of edible species. The two models will learn two different representations of rules that can be used to predict whether a given mushroom is editable or not. You will interpret those rules and try to generate a simple, human-readable description of what to look for when trying to determine if a mushroom is editable or not.

As a side note, you should be aware of similarity and differences between ML and data mining. ML and data mining are both concerned with automated discovery of patterns and rules from data. One of the most common methods in data mining is "association rule mining." For example, association ruling mining might be used to identify which products tend to be purchased together (reflected through "association rules") so that an online store can recommend additional items for a customer to add to their shopping cart. Many companies that sell enterprise software offer software suites for "business process automation," which combines some sort of "process mining" algorithm to infer rules and a "business rules engine" to automate decision making based on user-provided or derived rules. Business rule engines often incorporate a graphical user interface so that users can inspect or modify rules without needing to have programming experience.

In ML, automated discovery of patterns is just a step on the way to building predictive models. ML is concerned primarily with the predictive power of the resulting models and makes a conscious decision to employ techniques that generate very accurate models with no regard for explaining how predictions are made.

**Instructions**

**1. Loading the Data and Initial Assessment**

a. Load the mushrooms.csv file as a DataFrame.

b. Using the output of the initial head(), info(), and describe() methods to get a sense of what variables are in the data. What are the variable types? How many values does each variable have?

c. Which column indicates whether the mushrooms are poison or editable? This will be the output variable that we are predicting.

d. Which columns can be used as the input variables?

e. Convert all of the variables to categorical types.

**2. Label Encoding**

Machine learning training algorithms and models work on numerical data. We need to convert the sets of strings indicating the label and feature values to floating-point numbers. The labels will need to be organized as 1D vector with one entry per sample and the class of each sample as an integer-valued floating-point number. For example, the classes of the first 10 records are given as:

A screenshot of a cell phone

Description automatically generated with medium confidence

We want to encode every "p" (for poisonous) as a 1 and every "e" (for editable) as a 0. The first 10 entries of your label vector should be:

[1 0 0 1 0 0 0 0 1 0]

Create the label vector using the [LabelEncoder](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html) class from scikit-learn to do this.

**3. Feature Encoding**

We also need to represent the input variables as numerical features. The input variables are all categorical, so we'll need to encode them. We'll use a technique called [one-hot encoding](https://scikit-learn.org/stable/modules/preprocessing.html#preprocessing-categorical-features). Each create a column for each value of the categorical variable. When a sample has a given value, a 1 is put the associated column and 0's are put in the other columns. For example, the ring-number column has 3 unique values: n, t, and o. The three associated columns in a feature matrix might look like this:

|  |  |  |
| --- | --- | --- |
| **n** | **t** | **o** |
| 1 | 0 | 0 |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |

To convert the categorical variables: use Panda's [get\_dummies()](https://pandas.pydata.org/docs/reference/api/pandas.get_dummies.html) function to create a new DataFrame of "dummy columns" for each categorical column, append the DataFrames to a list, concatenate all of the resulting DataFrames horizontally into a single DataFrame using the Pandas [concat()](https://pandas.pydata.org/docs/reference/api/pandas.concat.html) function. Review the resulting DataFrame using the head() and info() methods.

Grab the underlying 2D Numpy array of the resulting DataFrame using the values property. Your matrix should have 8124 rows (one per sample) and 117 columns.

**4. Experimental Setup**

In most cases, we want to apply machine learning models to make predictions on data unavailable at training time. To evaluate the model under realistic conditions, we divide the data into training and testing sets. We hold the testing set data aside until we are ready to evaluate the model. Since the testing set is not used or referenced in training, the evaluation on the testing set gives us a realistic estimate of how the model will perform on new, unseen data.

Use scikit-learn's [train\_test\_split()](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html) function to divide the feature matrix and label vector into training (75%) and testing (25%) sets. The function will partition the rows of the matrix and vector consistently so that they continue to match up. It is important to pass the labels to the stratify parameter. The function will use the labels to ensure that the samples are partitioned so that the ratios of edible and poisonous mushrooms in the training and testing sets are match the ratio from the original data set.

**5. Train, Visualize, and Interpret a Decision Tree**

a. Train a Decision Tree model using the [DecisionTreeClassifier](https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html) class using the training set.

b. Predict the labels for the testing set.

c. Calculate the [accuracy score](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html) and a [confusion matrix](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html) from the predicted and true labels. This data set is relatively "easy," so you should achieve 100% accuracy and your confusion matrix should only have non-zero entries on the diagonal.

d. Plot the Decision Tree using the following code snippet:

dot\_data = export\_graphviz(dt,

feature\_names=df\_features.columns,

proportion=True,

precision=2,

rounded=True,

class\_names=encoder.classes\_,

impurity=False,

leaves\_parallel=True)

graph = graphviz.Source(dot\_data)

graph

Note that dt refers to the Decision Tree model, df\_features is the DataFrame of one-hot encoded columns, and encoder is the LabelEncoder object.

Decision Trees are trained using a greedy algorithm. For each split, all of the features are evaluated in terms of how well they separate the samples into their respective classes and the feature that offers the best split is chosen. As a result, features near the top of the tree tend to be the most informative / predictive.

e. The Decision Tree is pretty large and can be difficult to interpret. The DecisionTreeClassifier class takes a parameter called ccp\_alpha that can be used to control the tradeoff between the complexity of the resulting tree and prediction accuracy. In some situations, it may be preferable to accept a reduction in accuracy to achieve a model that you can more easily interpret and explain to non-technical stakeholders. Try values between 0.001 and 0.1. Evaluate the accuracy of and plot the resulting trees.

**6. Train and Interpret a Logistic Regression Model**

a. Train a Logistic Regression model using the [SGDClassifier](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html) class with the parameter loss="log".

b. Evaluate the model using accuracy and confusion matrix.

c. Use the [predict\_proba()](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html#sklearn.linear_model.SGDClassifier.predict_proba) method to get the estimated probabilities. This will be a samples x 2 array – only use the first column. Plot a histogram of the probabilities using Seaborn's [histplot()](https://seaborn.pydata.org/generated/seaborn.histplot.html) function. Observe that the probabilities are very bimodal in this case.

d. The Logistic Regression model is given by the following equation:

where are the feature values for a single sample, is the intercept, and are the weights for the features. When the model is trained, the training algorithm tries to find intercept and weight values that maximize correct predictions for the training set samples.

Inspect the model weights. The SGDClassifier has a 1 x features array coef\_ that contains the learned weights. Plot a histogram of the weights using the first (and only row). The weights with the largest magnitudes indicate features with the most predictive power.

**7. Lab Report**

Write a lab report that answers the following questions:

a. Identify and describe a situation from your career in which being able to explain how a machine learning model makes decisions to business partners and customers.

b. The feature matrix has 117 columns but not all features are used in the Decision Tree. How many features are used in the Decision Tree? What are they? Remember that a single feature can be used in multiple nodes. Why do you think the Decision Tree doesn't use all of the features?

c. Use the tree to identify the 3 most predictive features you identified in (b).

d. Identify the leaf with the largest proportion of samples for each class. Together, these two leaves classify around 85% of the samples. Trace the paths from the leaves to the root to identify the feature values that identify most the samples in each class. You should be able to write these rules as Boolean expressions of the form feature1 <= t1 and feature2 <= t2 … From these rules, write a few sentences explaining to someone who knows nothing about machine learning how to differentiate editable and poisonous mushrooms.

e. Choose a tree from the ccp\_alpha experiment that you believe offers the optimal tradeoff between interpretability and accuracy and write some text justifying your decision.

f. Print the features with weight magnitudes greater than the cutoff you chose. How much overlap is there between the features used in the Logistic Regression and Decision Tree models?

g. Negatively-weighted features that count towards an edible prediction, while positively-weighted features count towards a poisonous prediction. Write some text explaining what characterizes editable and poisonous mushrooms according to the Logistic Regression model.

h. Compare and contrast the features considered most predictive by the Decision Tree and Logistic Regression models. Do they characterize the two groups (poisonous vs. edible) in similar ways?

**Submission Instructions**

Save the Jupyter notebook and your report as PDFs and upload the files through Canvas.

**Rubric**

|  |  |
| --- | --- |
| Plots: appropriate types of plots were chosen for each analysis, axes are properly labeled, used correct axes for variables, points were colored as required, lines were coloring as required, used a legend if appropriate, chose appropriate axes limits to make plot readable and do not cause misleading interpretations, font sizes and figure resolution are legible. | 10% |
| Notebook Formatting and Presentation: Notebook is polished and clean. No unnecessary code. Section headers are used. Plots are described and interpreted using text. The report contains an introduction and conclusion. | 10% |
| Report: Report is written in a professional manner using proper grammar and spelling. Report is a useful standalone document that can be shared with a business partner. | 10% |
| Data Preparation: Labels are properly encoded to numerical values. Categorical features are encoded using dummy variables. Samples are divided into training and testing sets appropriately. Decision tree and logistic regression models and trained and evaluated on the right data sets. The original and regularized trees are plotted. The highest-weighted logistic regression features were identified. | 20% |
| Reflection Questions: Reasonable attempts were made to answer each reflection question. Answers are supported by appropriate plots. Analyses were executed correctly from a technical standpoint. | 50% |