Proposal

Capstone Project-Mushroom Classification

Lin Zhu

Domain Background

Mushroom hunting, also known as mushrooming or mushroom picking, describes the activity of gathering mushrooms in the wild, typically for eating. In common sense, generally mushrooms with bright color, special cap shape or wired smell are suspicious as toxic class, but is it all of the truth?

In this project, I utilized the mushroom dataset from UCI machine learning repository, to train and compared multiple machine learning models. The goal of our endeavors is to determine whether or not a mushroom would kill you. Given death is an extreme consequence for guessing incorrectly, which is hard to accept, the optimal model should have least error. Meanwhile, it is interesting to explore what features matter most in identifying the edibility.

Problem Statement

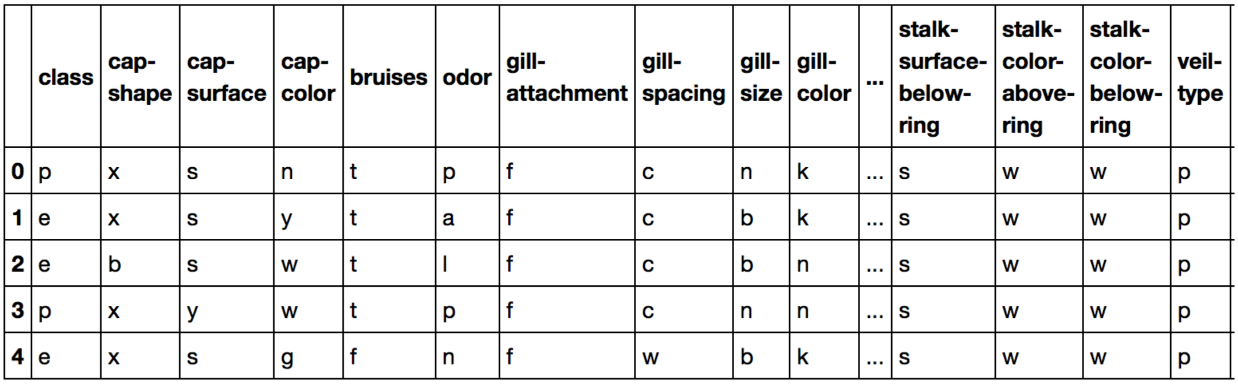
In this study, I am going design and evaluate a machine-learning model that could effectively classify the gill mushrooms as either "edible" or "poisonous":

- Apply machine-learning models to predict the mushroom edibility in testing dataset.
  - Solution step 1: utilizing Multilayer Perceptron Neural Network, train models with the training dataset and make predictions of testing dataset
  - Solution step 2:Refine the model if any unsatisfied performance
- Which features are most indicative of a poisonous mushroom?
  - Explore the correlation between multiple features
  - Identify the top two features

Datasets and Inputs

The UCI Machine Learning repository has a mushroom dataset, consisting of descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family Mushroom drawn from The Audubon Society Field Guide to North American Mushrooms (1981).

The “Mushroom.csv” dataset has 8124 sets of data points. One column “Class” including “e” as edible and “p” as “poisonous”, as label. The other 22 feature columns are shown below:
A sample data:



	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color	...	stalk-surface-below-ring	stalk-color-above-ring	stalk-color-below-ring	veil-type
0	p	x	s	n	t	p	f	c	n	k	...	s	w	w	p
1	e	x	s	y	t	a	f	c	b	k	...	s	w	w	p
2	e	b	s	w	t	l	f	c	b	n	...	s	w	w	p
3	p	x	y	w	t	p	f	c	n	n	...	s	w	w	p
4	e	x	s	g	f	n	f	w	b	k	...	s	w	w	p

The dataset itself is well balanced between two type labels (Figure 3), with an edible-to-poisonous ratio of 1.07.

Solution Statement

According to the problem statement, data preprocessing, MLP neural network and dimension reduction techniques (PCA) will be used to solve this problems. Details regarding each step are given below.

Data preprocessing:

- **Data cleaning**: remove non-related data, missing value or treatments to any other abnormalities
- **Data encoding**: transform categorical data into numerical data for further training and prediciton

Machine Learning Algorithm

- The initial thought is to encode the features utilizing one hot encoder, and expand it into a 116 columns of dummy features.
- Considering the large scale of final feature sets (116 X 8124), I decided to try Multilayer Perceptron (MLP) neural network models with tensorflow.

Dimension Reduction:

- Principle Component Analysis: uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components

Benchmark Model

This dataset was public available on Kaggle, the community already contributed multiple methods for edibility classification. Here a set of averaged value was selected as benchmark.

- For accuracy, my goal is to reach 0.98, optimally above 0.99
- For total running time, my goal is to reach 2s, optimally below 1s.

Evaluation Metrics

According to intimal observation, with an “e” over “p” ratio of 1.07,the dataset is balanced between “edible” and “poisonous”.

- Accuracy score is commonly used in binary classifier evaluation. It considers both true positive prediction and true negative predictions with equal weight.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN\ (the\ whole\ dataset)}$$

- Running time is another consideration. Sometimes there are trade off between the accuracy and running times. The differences in running time would increase along the data size extending. It would also generate high computing cost.

Project Design

The project will be conducted with the following steps

- **Data Cleaning**
  - Check the unique value of each features
  - Remove features with only one unique value
- **Date Encoding**
  - I choose One-Hot-Encoder since it has the advantage that result is binary rather than ordinal and that everything sits in an orthogonal vector space.
- **Machine Learning Model Design**
  - Generate training and testing datasets.
  - Define model
    - Layer shape (inputs, outputs, hidden layer)
    - Define weight and biases
    - Prediction function
  - Train and predict:
    - Loss function
    - Train_step
    - Feed function

- Accuracy
- Number of steps

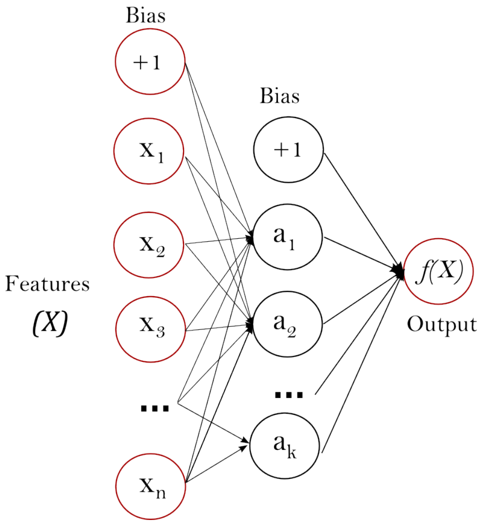


Figure 1. Typical MLP neural network

Feature Importance Study

For the second task, the initial feature importance study, I decide to use PCA to explore and find an answer that, among all 116 expanded features (X_oh, the other two columns are label columns with dummy values, marked as y_oh), which ones really count. The study follows steps below:

- Import PCA
- Fit PCA with feature dataset (116 columns X 8124 rows)
- Get the explained variance ratio
- Plot the explained variance for each feature