The Milestone Report of Capstone Project 1

1. The Aiming Problem

According to the American Association Poison Control Center(AAPCC)'s annual report, there were over 6000 people poisoned by poisonous mushroom per year over the past decade in U.S. The tricky thing about poisonous mushroom is, there is no simple rule to identify them from the non-poisonous ones by their outlook. Hence even experienced mushroom hunters may mistake a poisonous mushroom from the wild.
The objective of the project is to find out if it is possible to identify the poisonous mushroom by the outlook.

2. The Clients

The aiming clients of the project is professional/amateur mushroom hunters, survivalist and anyone who would like to put wild mushroom onto their dining tables but not sure about its edibilty.

3. The Data Set

It will be using the mushroom data set from UCI's machine learning repository.
The dataset has 8124 observations, 22 features. Here is the details:
Attribute Information:
(classes: edible=e, poisonous=p)
cap-shape: bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s
cap-surface: fibrous=f, grooves=g, scaly=y, smooth=s
cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r,pink=p,purple=u,red=e,white=w,yellow=y
bruises: bruises=t, no=f
odor: almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s
gill-attachment: attached=a, descending=d, free=f, notched=n
gill-spacing: close=c, crowded=w, distant=d
gill-size: broad=b, narrow=n
gill-color: black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o, pink=p, purple=u, red=e, white=w, yellow=y
stalk-shape: enlarging=e, tapering=t
stalk-root: bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=?
stalk-surface-above-ring: fibrous=f, scaly=y, silky=k, smooth=s
stalk-surface-below-ring: fibrous=f, scaly=y, silky=k, smooth=s
stalk-color-above-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
stalk-color-below-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
veil-type: partial=p, universal=u

veil-color: brown=n, orange=o, white=w, yellow=y

ring-number: none=n, one=o, two=t

ring-type: cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z

spore-print-color: black=k, brown=n, buff=b, chocolate=h, green=r, orange=o, purple=u, white=w, yellow=y

population: abundant=a, clustered=c, numerous=n, scattered=s, several=v, solitary=y

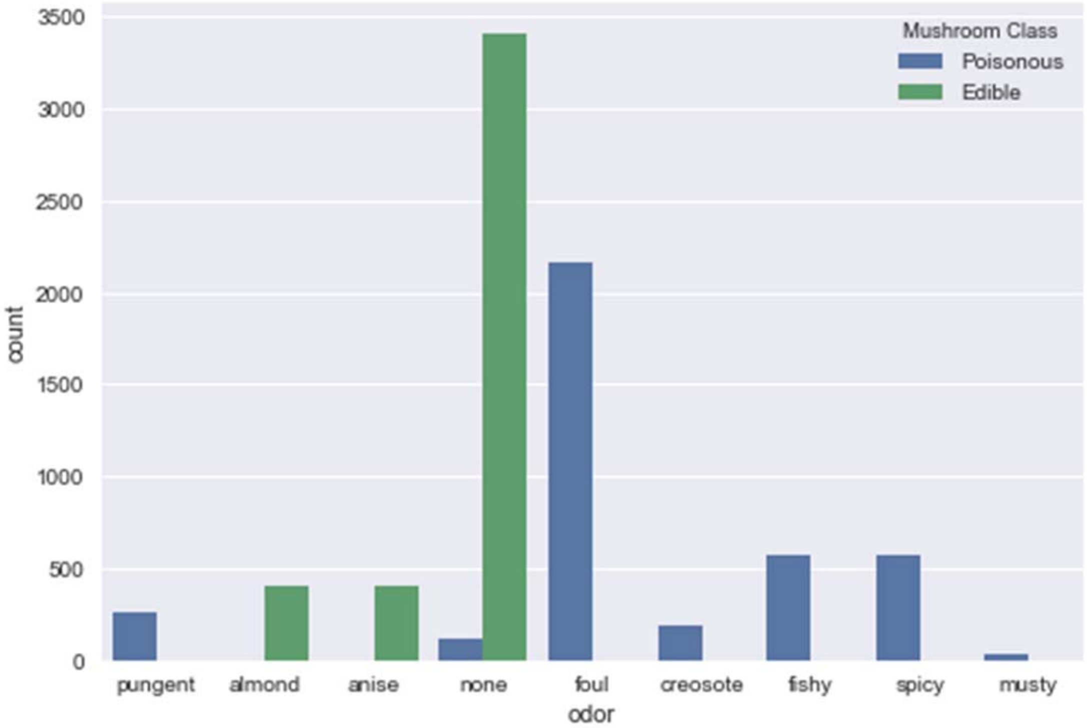
habitat: grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d

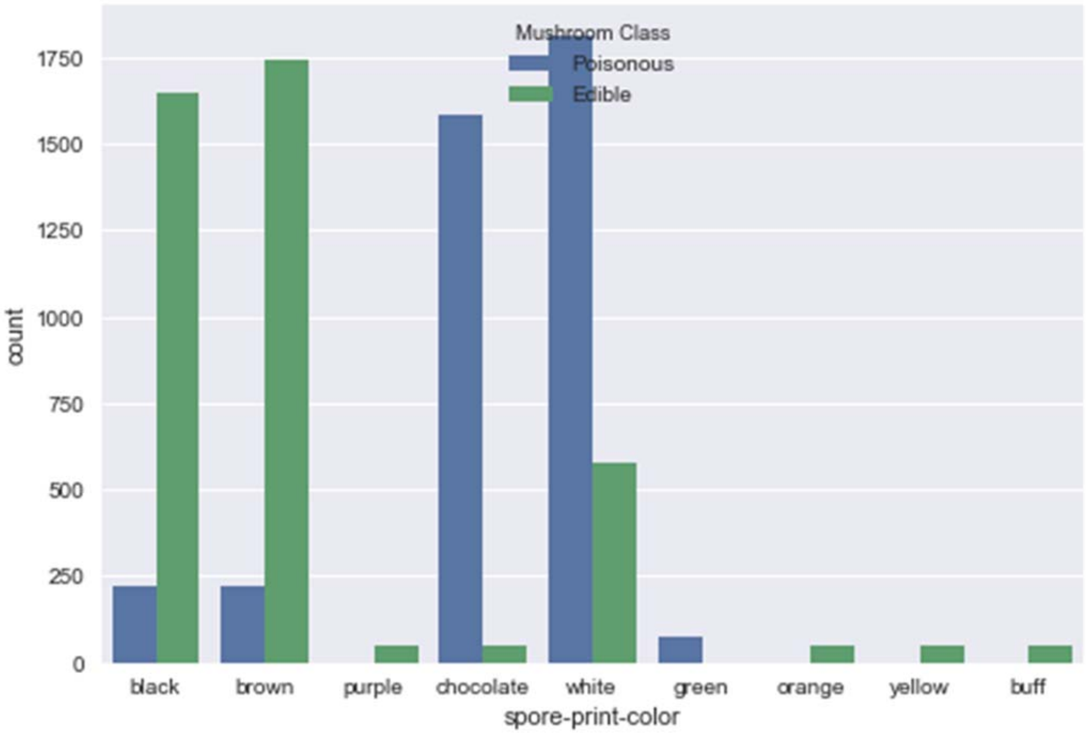
4. Data Wrangling

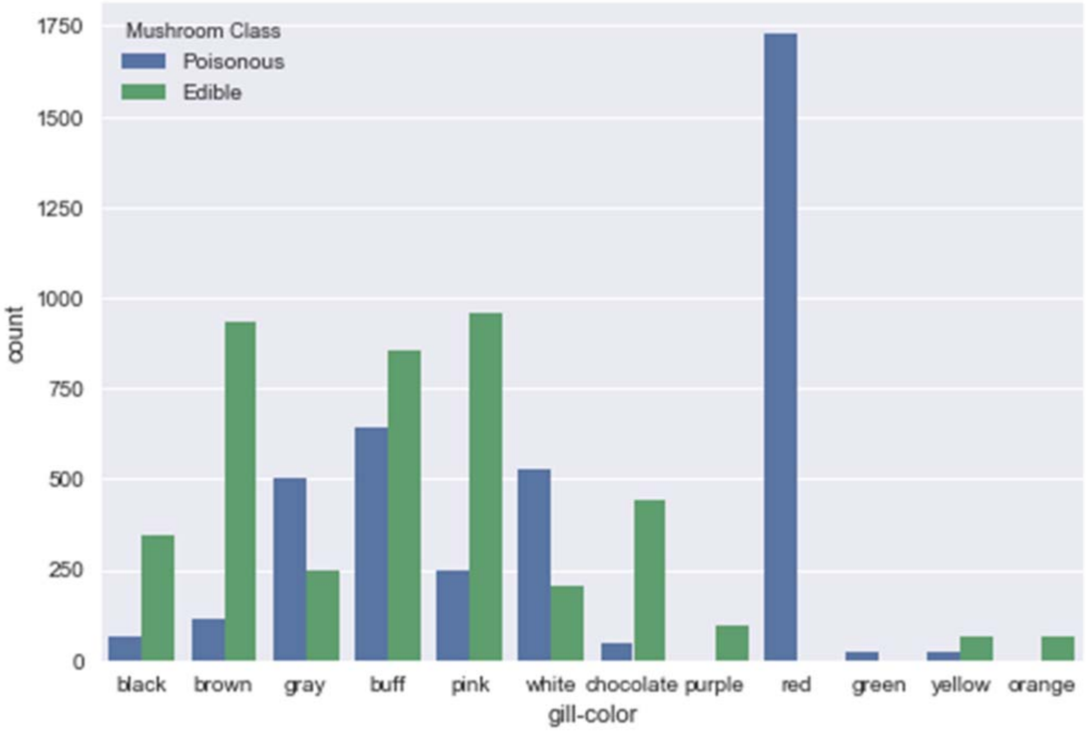
The mushroom data set is quite clean already, thus not much cleaning is required. The only missing data is within the '*stalk-root'* variable, there are 2480 of missing values. The amount of the missing values are too large to just abandon, hence they are replaced with guessing values (the value of the previous row --'ffill' method).

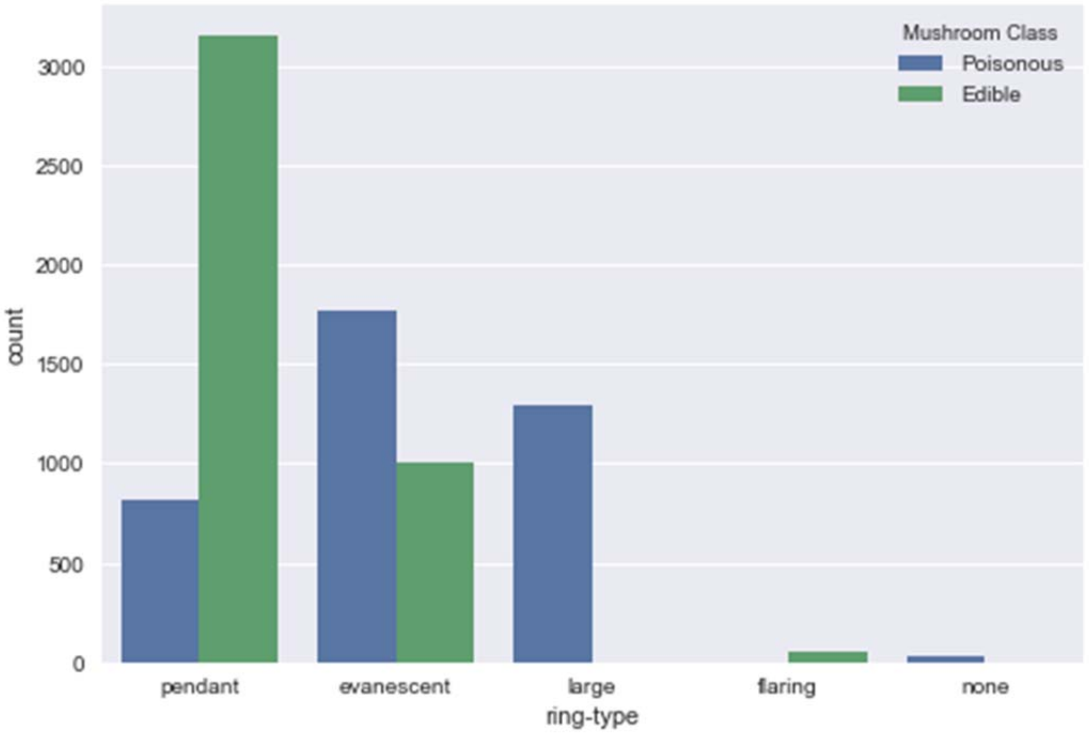
5. Initial Findings

The chi-square test is run between the target variable(*'Class'*) and all the input variables. It turns out that the target variable has stronger association with some of the input variables, namely, the 'odor', 'spore-print-color', 'gill-color', and 'ring-type'.




Mushroom Class
Poisonous
Edible
count
1750
1500
1250
1000
750
500
250
0
black
brown
purple
chocolate
white
green
orange
yellow
buff
spore-print-color


Mushroom Class
Poisonous
Edible
count
1750
1500
1250
1000
750
500
250
0
black
brown
gray
buff
pink
white
chocolate
purple
red
green
yellow
orange
gill-color



And the target variable has zero association with the ‘veil-type’ variable, for all the observations fall into one veil type, partial type. Hence the ‘veil-type’ won’t help distinguish a mushroom’s edibility.

6. Data preprocessing

Two preprocessing steps are taken:
Frist, according to the second part of the initial findings, the whole ‘veil-type’ column is dropped.
Second, due to the categorical nature of the data, they are converted into numerical variables using the *get_dummies()* method from *pandas.*

7. Train data and test data

The data set is splatted into training set and test set, the test set is 20% of the original data set, the train data size is 80%, randomly selected using the train-test-split function. The parameters tuning process will only use the train data, the test data will only be used to test for accuracy after the parameters are tuned.

8. Algorithms

Two separate algorithms are used, the Support Vector Classification (SVC) and the Multinomial Naive Bayes (MultinomialNB). The details are below:

The SVC
Under the SVC model, the *GridsearchCV (CV=5)* function is applied for three of the parameters’

tuning:
Kernel : linear, rbf
Gamma (rbf kernel only): 0.001, 0.0001
C (linear and rbf kernel) : 1, 10, 100, 1000

Both the linear and rbf kernel can reach 100% grid scores, the optimal parameters for rbf kernel is *'C': 1000, 'gamma': 0.001, 'kernel': 'rbf',* and the optimal parameters for linear kernel is *'C': 1, 'kernel': 'linear'.*
While tested on the test data, both the precision rate and the recall rate is 100%, which means that the classification makes zero mistake.

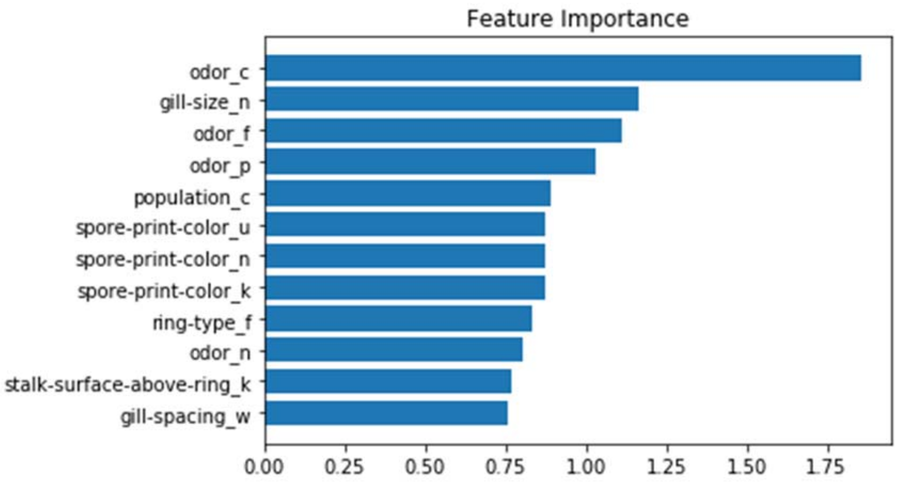
The Multinomial Naive Bayes

The similar GridsearchCV(CV=5) function is used, but only one parameter needs to be tuned under the MultinomialNB model:
Alpha: 0.0001, 0.001, 0.01, 0.1, 1, 10
The grid score is slightly less than the SVM model, the best grid score is 99.7% while alpha=0.0001, but the test result also reaches 100% precision rate and 100% recall rate.

9. Feature Importance

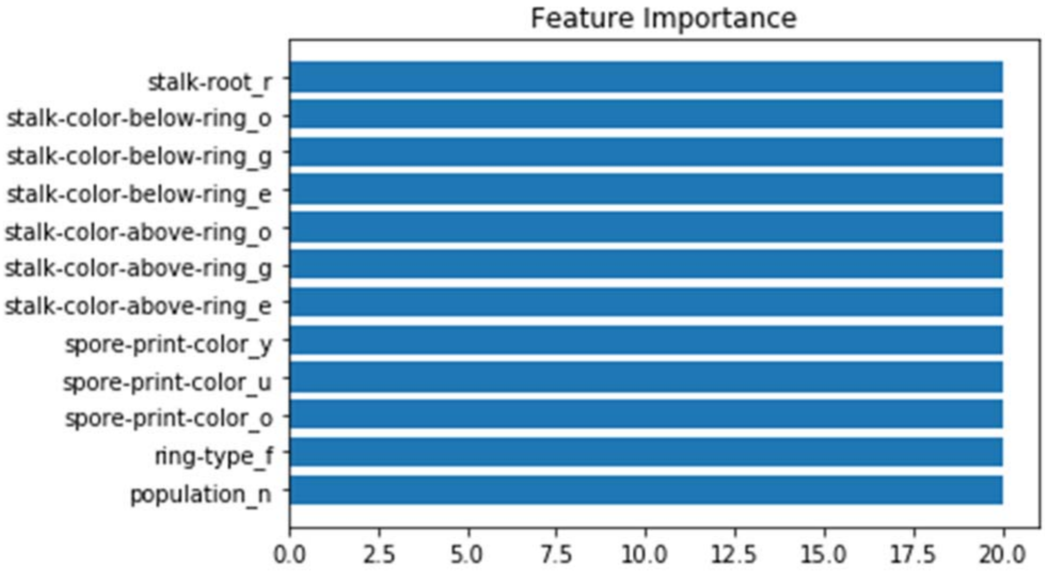
SVM model:

Under the linear kernel, feature importance can be estimated using the *coef_* attribute. A linear SVM creates a hyperplane to maximize the linear distance between the two classes. The *svc.coef_* attribute represents the vector coordinates which are orthogonal to the hyperplane and their direction indicates the predicted class. The absolute size of the coefficients in relation to each other can then be used to determine feature importance for the data separation task.
But with non-linear kernels like the rbf, the hyperplane does not measure the linear distance between the two classes, therefor the hyperplane does not exist in the original space, and the coefficients are not directly related to the input features.
The 12 features with the highest absolute values coefficients are as followed:



The Multinomial NB:

The Multinomial NB is also a linear algorithm, therefor the feature importance can also be estimated with the *MultinomialNB.coef_* attribute, and the results are as followed:



Clearly the two models put on different weights on the features, but the spore-print-color, ring-type, are two of the most import features of both models, which is consistent with the chi square test result in step 5.

10. Conclusions

The machine learning technique can separate poisonous mushrooms from the edible ones, with 100% accuracy. If built into a smart phone app, it can help the mushroom hunters to avoid the poisonous mushrooms in the wild. When the app users come across the mushroom that they are not sure about, as long as they input the correct information such as odor, color etc., the app will give the correct classification.