Mushroom Classification Analysis

Computer Science Capstone

Jason Xiong

[redacted]

Western Governors University

Table of Contents

Project Proposal

Problem Statement

[redacted] Inc. is a company that provides guided mushroom foraging tours. There has been an increase in foragers and their tours are almost always booked. People seek their tours to learn more about how to classify a mushroom and which mushrooms they should forage. [redacted] Inc. has also seen an increase in emails regarding mushroom classification and their response team cannot keep up with the load. With mushroom foragers on the rise, [redacted] Inc. is in need of a web application that can be integrated onto their existing website that will allow users to classify mushrooms easier without the need of sending an email and waiting for a response.

Application Benefits

[redacted] Inc.'s response team's workflow and load will be minimized. Classification of mushrooms will become much easier for foragers and will minimize deaths. Even the most advanced foragers can have difficulty classifying a mushroom; the application will benefit everyone.

Application Description

The application was built with Python and utilized Python's matplotlib and seaborn libraries that generated visualizations with the compiled dataset. The dataset is in the form of a CSV file that is read with Python's pandas library and is displayed on its own page that can be

filtered and searched to allow data exploration. Scikit-Learn is the main library that built the machine learning algorithm that used the random forest classifier on the mushroom dataset.

The machine learning model will be deployed onto the client's website with the use of an API using Flask as the main container of deploying the website. Being a web application, users and the client will easily be able to interact with it. The machine learning algorithm has a 99% success rate of classifying whether the mushroom is poisonous or edible. A log file will be generated to ensure proper navigation, loading of web pages, and successful API calls to monitor the health of the system.

Data

The data required for this project to properly build the model will be from all mushroom forager's experiences that can be uploaded to the website directly. Users may email directly their own dataset that will then be aggregated to existing data to better build the machine learning model. It is expected that all features of the mushroom be provided to expand the existing dataset. The table below are the expected features along with possible values.

Features	**Values**
Cap Shape	Bell, Conical, Convex, Flat, Knobbed, Sunken
Cap Surface	Fibrous, Grooves, Scaly, Smooth
Cap Color	Brown, Buff, Cinnamon, Gray, Green, Pink, Purple, Red, White, Yellow
Bruises	Yes, No
Odor	Almond, Anise, Creosote, Fishy, Foul, Musty, Pungent, Spicy, None
Gill Attachment	Attached, Descending, Free, Notched
Gill Spacing	Close, Crowded, Distant
Gill Size	Broad, Narrow
Gill Color	Black, Brown, Buff, Chocolate, Gray, Green, Orange, Pink, Purple, Red, White, Yellow

Stalk Shape	Enlarging, Tapering
Stalk Root	Bulbous, Club, Cup, Equal, Rhizomorphs, Rooted, Missing
Stalk Surface Above Ring	Fibrous, Scaly, Silky Smooth
Stalk Surface Below Ring	Fibrous, Scaly, Silky Smooth
Stalk Color Above Ring	Brown, Buff, Cinnamon, Gray, Orange, Pink, Red, White, Yellow
Stalk Color Below Ring	Brown, Buff, Cinnamon, Gray, Orange, Pink, Red, White, Yellow
Veil Type	Partial, Universal
Veil Color	Brown, Orange, White, Yellow
Ring Number	None, One, Two
Ring Type	Cobwebby, Evanescent, Flaring, Large, None, Pendant, Sheathing, Zone
Spore Print Color	Black, Brown, Buff, Chocolate, Green, Orange, Purple, White, Yellow
Population	Abundant, Clustered, Numerous, Scattered, Several, Solitary
Habitat	Grasses, Leaves, Meadows, Paths, Urban, Waste, Woods

Objectives

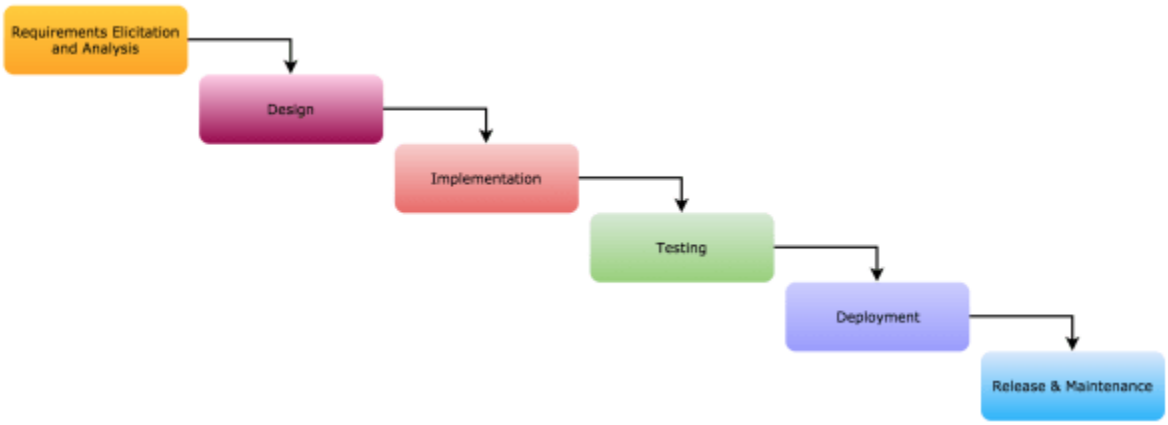
The project's primary objectives are to reduce the risk of deaths and unhoped for visits to the hospital due to eating a poisonous mushroom. Foragers from everywhere will be able to use the API and website to classify their mushrooms. The main objective for [redacted] Inc. is to help manage their workload and provide accurate information within seconds.

Hypothesis

A mushroom can be classified as poisonous or edible based on its identifiable features. The machine learning model is trained to classify the mushroom based on these features.

Project Methodology

The product will be developed and supported using the waterfall methodology. Our team has opted to use the waterfall methodology because the requirements are well-defined, and the application is relatively small. There are six phases to the waterfall methodology:



1. Requirements Elicitation and Analysis – Gather the requirements and scope of the project to ensure no changes will happen during any other phases.
2. Design – Mockup a design to ensure the system meets all requirements.
3. Implementation – Develop the application and ensure each component is functional.
4. Testing – Integrate and merge components to test functionality.
5. Deployment – Install the product.
6. Maintenance – Evaluate the product and ensure it continues to function as needed.

Funding Requirements

Funding for the project comes out to $10,000 for development, $300 a year for API and cloud hardware/services, and $200/hour for maintenance.

Stakeholder Impact

This project solution provides real-time mushroom classification to help people forage mushrooms safely. Traffic to [redacted] Inc.'s website and location for foraging tours will help generate more revenue. The response team at [redacted] Inc. will have a better workflow. Everyone will be benefit from the solution.

Ethical and Legal Considerations

The project does not use any personally identifiable information. No humans or animal subjects are involved. The project also does not track any health or education information.

The system's machine learning algorithm must be used at the customer's own risk. [redacted] Inc. and [redacted] Inc. will not be held accountable for deaths or doctor visits due to foraging poisonous mushrooms.

Expertise and Experience

The developer has 10 years of experience in software development and machine learning applications. The developer also has deployed multiple web applications and machine learning models to the web for public use.

Executive Summary

Problem Statement

More than 6,000 calls regarding consumption of poisonous mushrooms are answered by poison control centers. Over 500 of the cases result in serious illness, while about 2,500 are

treated in healthcare facilities (Verbanas, 2013). Even nibbles of poisonous mushrooms can cause severe illness or even death. The most frequent symptom after consuming a poisonous mushroom can include nausea, vomiting, cramps, and diarrhea. Not only humans are in danger of poisonous mushrooms, but animals as well. Sometimes poisonous mushrooms do not show any symptoms for a while and prevents people from thinking they need treatment (Voa, 2018).

Mushroom foraging is growing in popularity, and it should be done safely to avoid unwanted visits to the doctor.

Customers

The primary customers of this product will be for all mushroom foragers and those like [redacted] Inc. that offer foraging tours and wish to spread proper knowledge about mushrooms. All mushroom foragers must be able to classify whether the mushroom is poisonous or edible. There are only a few features needed to classify the mushroom that anyone can be able to identify and input into the system.

Existing System

There currently is not an existing system that is accessible to everyone to classify a mushroom. Prior knowledge and experience with foraging mushrooms would be needed to ensure proper foraging techniques. The only way to have a better idea of proper foraging techniques would be to attend a foraging tour.

[redacted] Inc.'s response team would answer multiple emails containing questions about classifying mushrooms. The new system will help lower their workflow and should draw in more website traffic flow along with more foraging tours to increase revenue.

Data

Initial data was collected from the [redacted] and used to train the current machine learning model. Data will continually be collected from all foragers who wish to enter data or send in data to improve the machine learning model. This data will contain all necessary features needed to classify the mushroom as poisonous or edible. Any data that is missing information will be discarded to ensure a properly trained model. Even though the current machine learning model has a 99% accuracy rate, more compiled data will only improve the system.

Methodology

The waterfall methodology will be used to complete this project. There are six phases to the waterfall methodology:

1. Requirements Elicitation and Analysis.
    - During this phase, stakeholders will be gathered to go over the requirements and scope of the project. Requirements and scope must be completely established before moving onto the next stage, as no changes should happen during any other phases.
2. Design – Mockup a design to ensure the system meets all requirements.

- The team assigned to this project will make a design of how the project should be implemented. Necessary tools and products will be gathered to ensure smooth completion of the project.

3. Implementation – Develop the application and ensure each component is functional.
    - The team will begin making the software and will be developed to match the requirements. The product will contain a dashboard, machine learning model, and web application for user interaction.
4. Testing – Integrate and merge components to test functionality.
    - Proper testing of the product will be done to ensure it has met the requirements. Dashboard should be intuitive to all users. Machine learning model should return accurate results.
5. Deployment – Install the product.
    - Properly deploy the product to the customer and ensure the product is working after deployment.
6. Maintenance – Evaluate the product and ensure it continues to function as needed.
    - Continue to monitor and maintain the product after deployment. Any bugs, defects, or upgrades can be done in this phase.

Deliverables

A requirements file will be included to ensure all requirements are met to be installed on the machine. A simple command will run the requirements file and install necessary packages to ensure the product is working properly. Executables will be delivered to provide a seamless installation and launch. API keys and necessary logins will be provided as well. A

database shall be built to easily collect data and maintain the machine learning model. This allows for easy upgrades and improvements to the model.

Implementation Plan

1. Gather a balanced raw dataset containing information about both edible and poisonous mushrooms.
2. Implement and utilize descriptive techniques to reduce variables to prevent overfitting of data.
3. Develop a predictive machine learning model with a non-descriptive technique.
4. Test and tune the accuracy of the machine learning model.
5. Develop a dashboard containing data visualizations, the predictive model, and data insight.
6. Dock the product onto the cloud.
7. Deploy the environment to production.
8. Perform acceptance testing.

Requirements Validation

The software is validated by running tests in beta and passing in gathered test data. Logs of the model will be documented to ensure the model is correctly predicting the classification of the mushrooms. A confusion matrix will be used to determine the accuracy, precision, recall, and F1 score of the model to ensure get the best predictions. Accuracy of the model is expected to be above 95%.

The product will be validated by deploying to a beta server to ensure it is fully functional. Checking of logs will be done to ensure capabilities of monitoring the system's health.

Environment and Funding

Cloud and API Server	AWS	$25/month, $300/year
Development Cost	50 hours @ $200/hour	$10,000
Maintenance		$200/hour
Total		$10,000 upfront cost + $25/month. Maintenance for $200/hour

Timeline

Phase	**Begin**	**End**	**Duration**
Requirements	9/01/2019	9/05/2019	5 days
Design	9/06/2019	9/11/2019	6 days
Implementation	9/12/2019	10/01/2019	20 days
Testing	10/02/2019	10/16/2019	15 days
Deployment	10/17/2019	10/18/2019	1 day

Product Attributes

Descriptive Method

Multiple descriptive methods were used. Recursive Feature Elimination, K-Means clustering, logistic regression all played a factor to build the machine learning model. All were used to finding the most important features for the predictive method.

Non-Descriptive (Predictive) Method

Random Forest Classification was used to predict whether the mushroom was edible or poisonous.

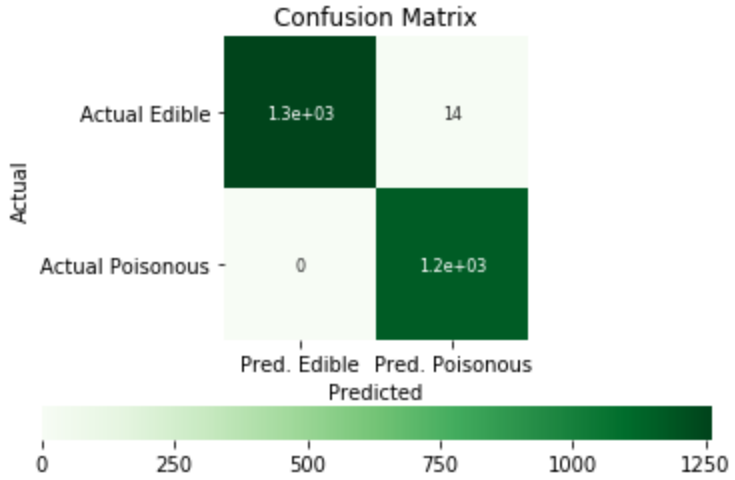
Decision Support

The product allows users to input the features of the mushroom to determine whether the mushroom is poisonous or edible. The confusion matrix below indicates false positives vs. false negatives based on the test dataset. As you can see from the test dataset, we are reaching almost 100% accuracy.

The way the random forest classifier works is “a large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models” (Yui, 2019). In other words, the random forest classifier breaks up the dataset into multiple trees and each tree “votes” on what they believe their test dataset will classify. Aggregate the trees together and they will produce an excellent prediction model.

Accuracy 0.9942575881870386

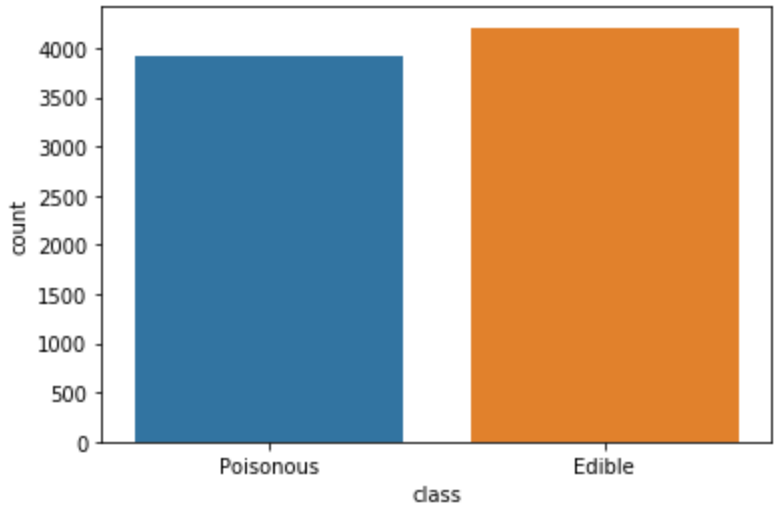
	precision	recall	f1-score	support
0	1.00	0.99	0.99	1274
1	0.99	1.00	0.99	1164
accuracy			0.99	2438
macro avg	0.99	0.99	0.99	2438
weighted avg	0.99	0.99	0.99	2438



Data Management and Preparation

The dataset used to train and test the machine learning model was provided by the [redacted] Mushroom foraging has become incredibly popular and the data helps foragers understand which mushrooms are safe.

I looked at the data and ensured there was enough information on both edible and poisonous mushrooms; having a balanced dataset is important. I also checked for any missing data containing NaN or blanks; luckily there wasn't any data missing for all rows. The confusion matrix (displayed in the Data Visualization section) also shows that the category "veil-type" is useless because there is only one unique value for every row.




```
data.isnull().sum()  # Display the counts of null values

class                     0
cap-shape                 0
cap-surface               0
cap-color                 0
bruises                   0
odor                      0
gill-attachment           0
gill-spacing              0
gill-size                 0
gill-color                0
stalk-shape               0
stalk-root                0
stalk-surface-above-ring  0
stalk-surface-below-ring  0
stalk-color-above-ring    0
stalk-color-below-ring    0
veil-type                 0
veil-color                0
ring-number               0
ring-type                 0
spore-print-color         0
population                0
habitat                   0
```

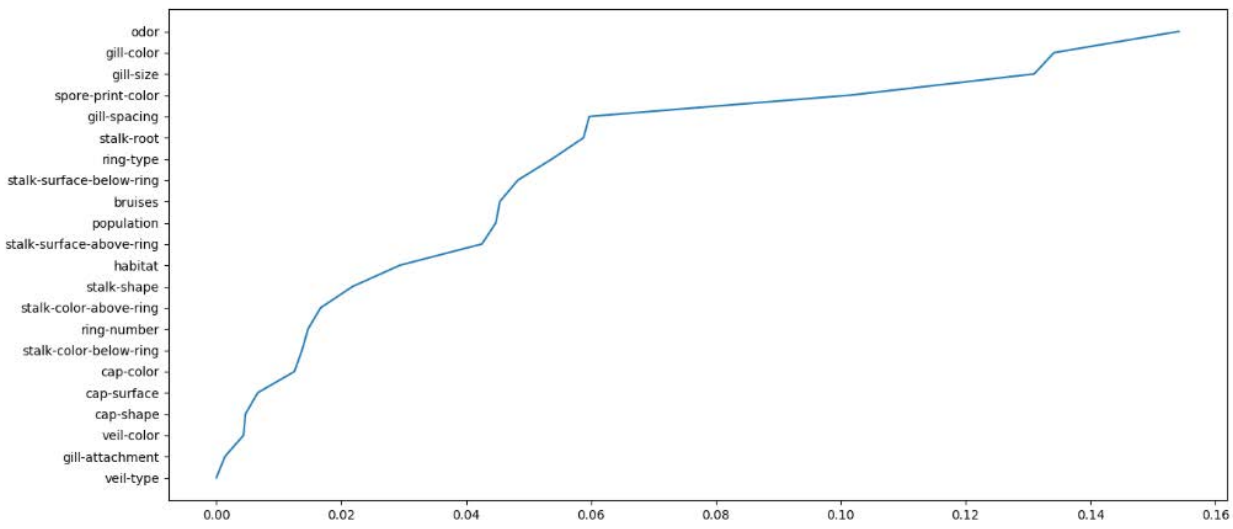
Data Exploration

The data can be explored by the filterable table on the “Data Table” webpage. There is also a bar chart (shown in the Data Visualization section) that shows several categories of how many mushrooms were classified as poisonous or edible based on the specific category.

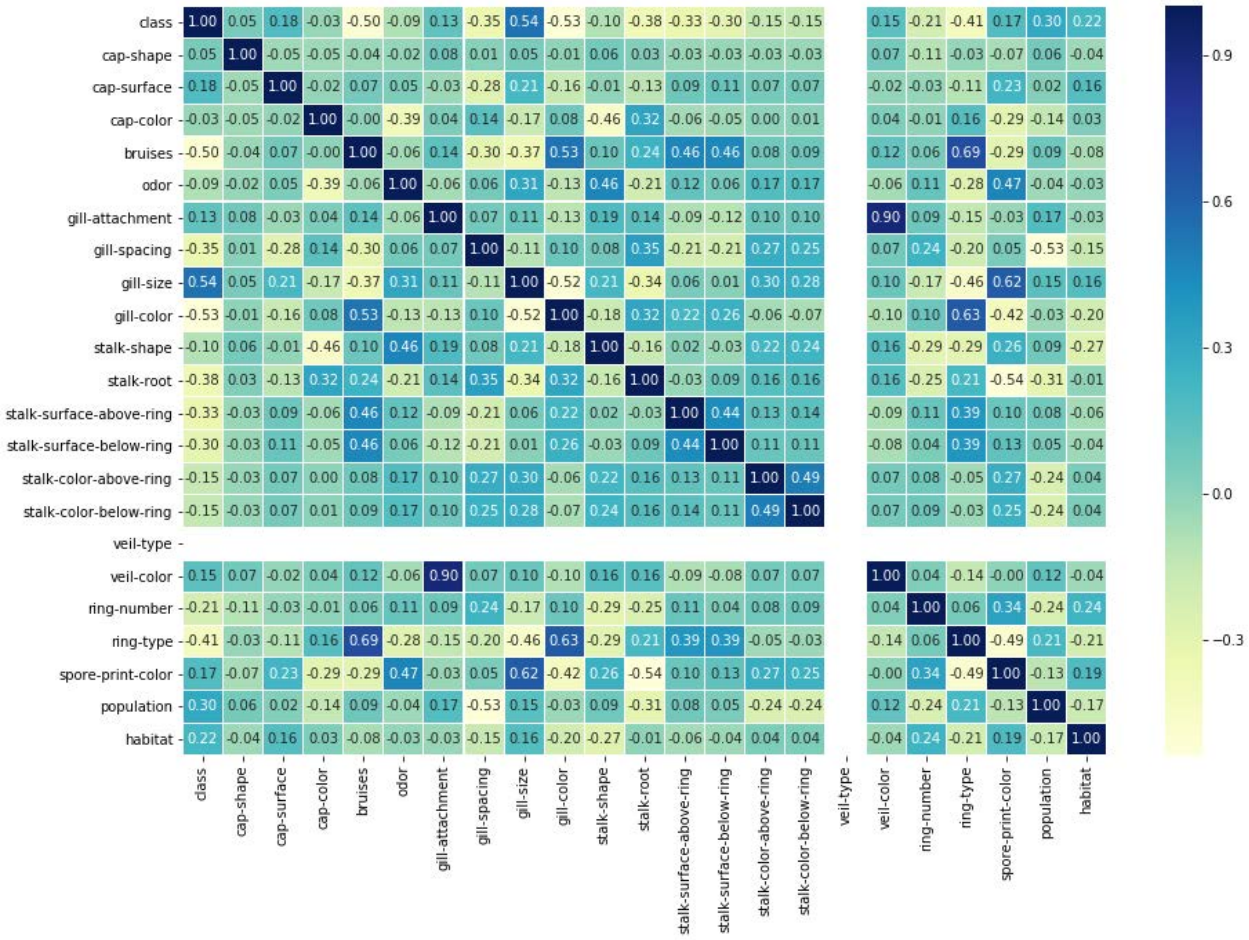
	class	cap-shape	cap-surface	cap-color	
Search	Search class	Search cap-shape	Search cap-surface	Search cap-color	Search bru
0	Poisonous	convex	smooth	brown	bruises
1	Edible	convex	smooth	yellow	bruises
2	Edible	bell	smooth	white	bruises
3	Poisonous	convex	scaly	white	bruises
4	Edible	convex	smooth	gray	no
5	Edible	convex	scaly	yellow	bruises
6	Edible	bell	smooth	white	bruises
7	Edible	bell	scaly	white	bruises
8	Poisonous	convex	scaly	white	bruises
9	Edible	bell	smooth	yellow	bruises

Data Visualization

Data is visualized through the Visualizations page. The line graph shows the most important features to classify the mushroom which is used in the predictive model.



Notice how the top four features help classify over 10% of the mushroom.



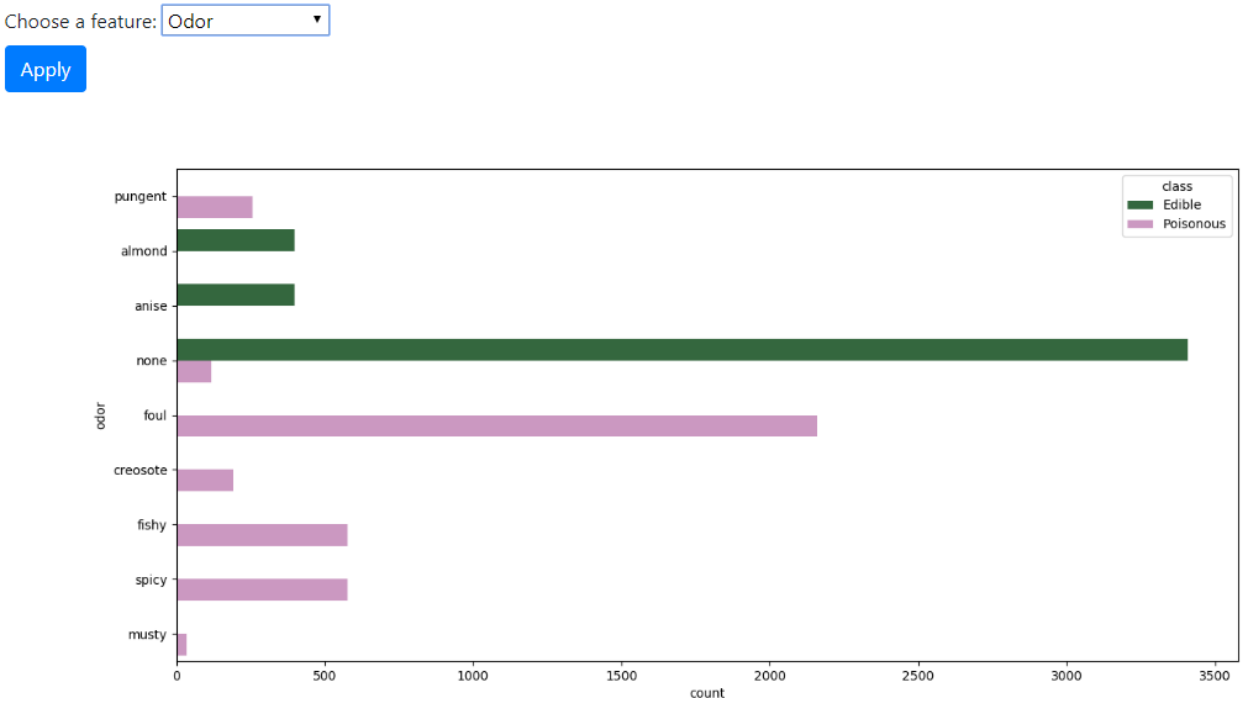
The correlation matrix shows how strong each category meshes well with other categories. We can see from veil-type that it is completely whited out because it doesn't help classify the mushroom at all.

Interactive Queries

Users can look through the data table containing the dataset and is able to filter based on whichever feature they wish. There are 23 features they can filter through.

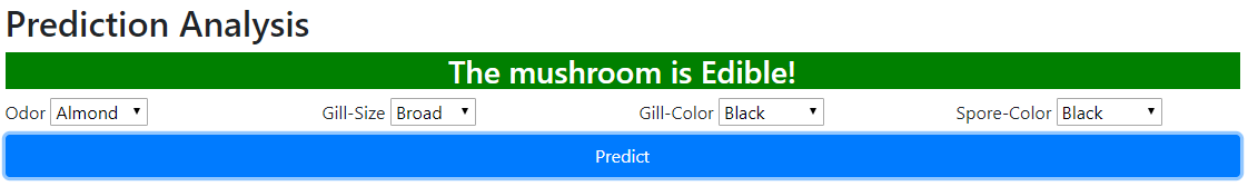
	class	cap-shape	cap-surface	cap-color	
Search	Search class	Search cap-shape	Search cap-surface	Search cap-color	Search bru
0	Poisonous	convex	smooth	brown	bruises
1	Edible	convex	smooth	yellow	bruises
2	Edible	bell	smooth	white	bruises
3	Poisonous	convex	scaly	white	bruises
4	Edible	convex	smooth	gray	no
5	Edible	convex	scaly	yellow	bruises
6	Edible	bell	smooth	white	bruises
7	Edible	bell	scaly	white	bruises
8	Poisonous	convex	scaly	white	bruises
9	Edible	bell	smooth	yellow	bruises

The bar chart updates in real time when the user clicks “Apply” to see specific categories of edible and poisonous mushrooms. This gives a good understand and idea of the data provided and can eliminate whether the mushroom is poisonous or edible from a certain feature.



Machine Learning

The random forest classifier was used for the predictive machine learning model. The model can be retrained in models.py. There are dropdowns for the most important features that will be used for the machine learning model to predict whether the mushroom is poisonous or edible. There will be a response when the button is clicked of the classification.

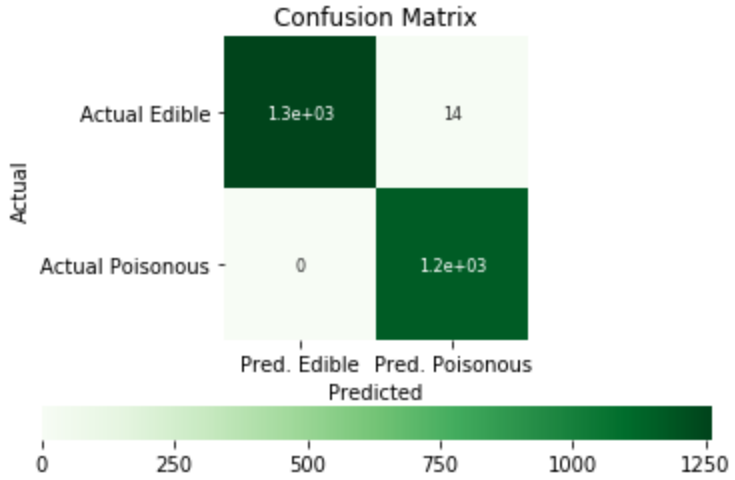


Accuracy Evaluation

The accuracy of the machine learning model was evaluated via a confusion matrix.

Accuracy 0.9942575881870386

	precision	recall	f1-score	support
0	1.00	0.99	0.99	1274
1	0.99	1.00	0.99	1164
accuracy			0.99	2438
macro avg	0.99	0.99	0.99	2438
weighted avg	0.99	0.99	0.99	2438



Security

A logging system was implemented to track the navigation of web pages when the program is in use. POST requests are tracked for the prediction model and the real time bar chart. A login page was also included for administrative purposes but can be extended to other users as well.

As a web application, the program will be uploaded onto a cloud server and can track the application better.

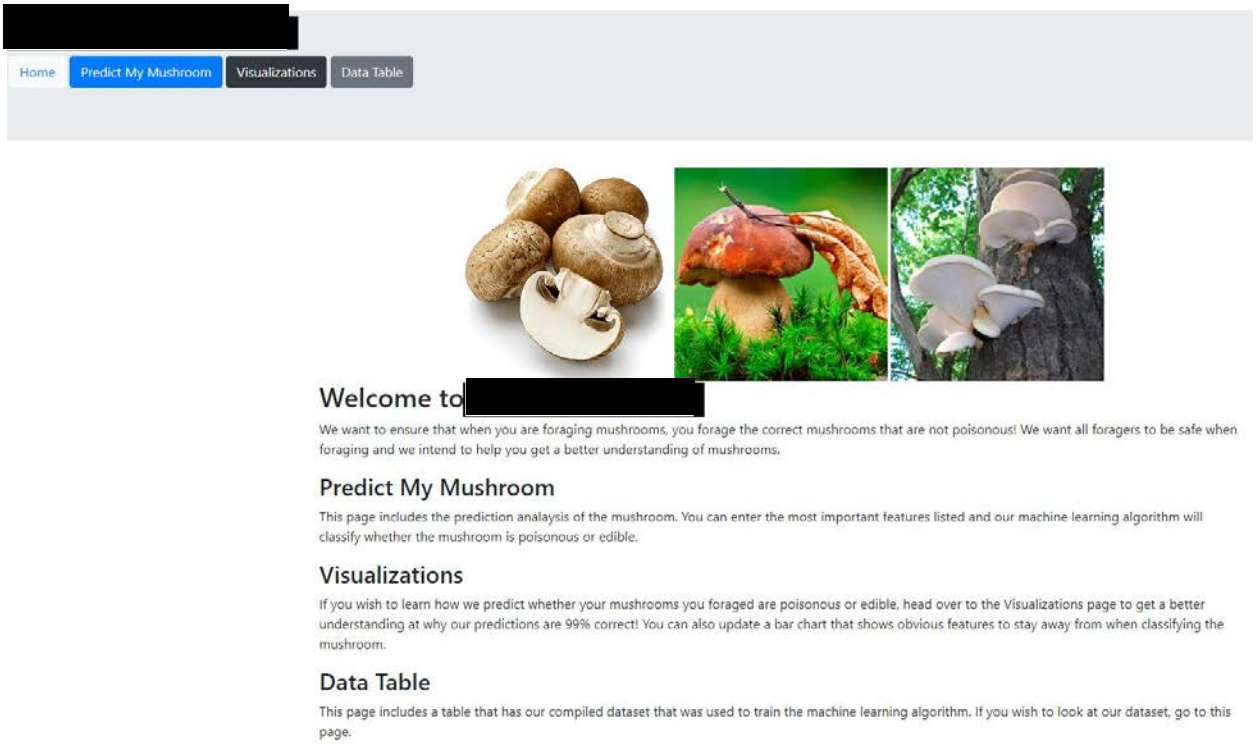
Monitor and Maintenance

As mentioned in the security section, a log file was implemented to ensure the health of the product. When the application is docked onto the cloud, there will be more monitoring tools such as amount of POST requests and traffic to the website.

The machine learning model can be retrained as many times as the client wishes. When more data is provided, the machine learning model can be retrained and tested for accuracy using a confusion matrix or other means.

Dashboard

The dashboard/home page includes the navigation of the website and what each webpage entails.



Documentation

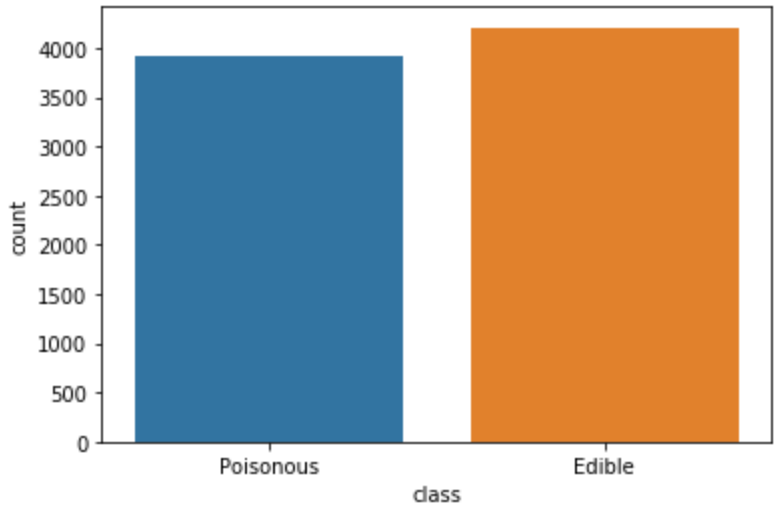
Business Requirements

1. The product shall help the workflow of the response team at [redacted] Inc.
2. The product shall predict the classification of whether the mushroom is poisonous or edible with high accuracy.
3. The product shall be easy to use and intuitive to any user.

Dataset

The dataset used to train and test the machine learning model was provided by the [redacted] Mushroom foraging has become incredibly popular and the data helps foragers understand which mushrooms are safe.

I looked at the data and ensured there was enough information on both edible and poisonous mushrooms; having a balanced dataset is important. I also checked for any missing data containing NaN or blanks; luckily there wasn't any data missing for all rows. The confusion matrix (displayed in the Data Visualization section) also shows that the category "veil-type" is useless because there is only one unique value for every row.




```
data.isnull().sum()  # Display the counts of null values

class                     0
cap-shape                 0
cap-surface               0
cap-color                 0
bruises                   0
odor                      0
gill-attachment           0
gill-spacing              0
gill-size                 0
gill-color                0
stalk-shape               0
stalk-root                0
stalk-surface-above-ring  0
stalk-surface-below-ring  0
stalk-color-above-ring    0
stalk-color-below-ring    0
veil-type                 0
veil-color                0
ring-number               0
ring-type                 0
spore-print-color         0
population                0
habitat                   0
```

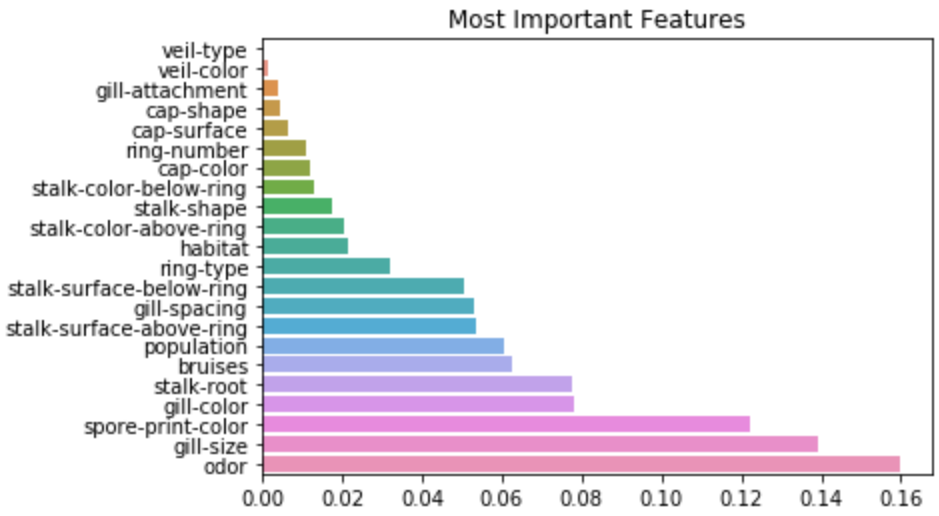
```
data.nunique()  # Get the unique counts for each category

class                      2
cap-shape                  6
cap-surface                4
cap-color                 10
bruises                    2
odor                       9
gill-attachment            2
gill-spacing               2
gill-size                  2
gill-color                12
stalk-shape                2
stalk-root                 5
stalk-surface-above-ring   4
stalk-surface-below-ring   4
stalk-color-above-ring     9
stalk-color-below-ring     9
veil-type                  1
veil-color                 4
ring-number                3
ring-type                  5
spore-print-color          9
population                 6
habitat                    7
```

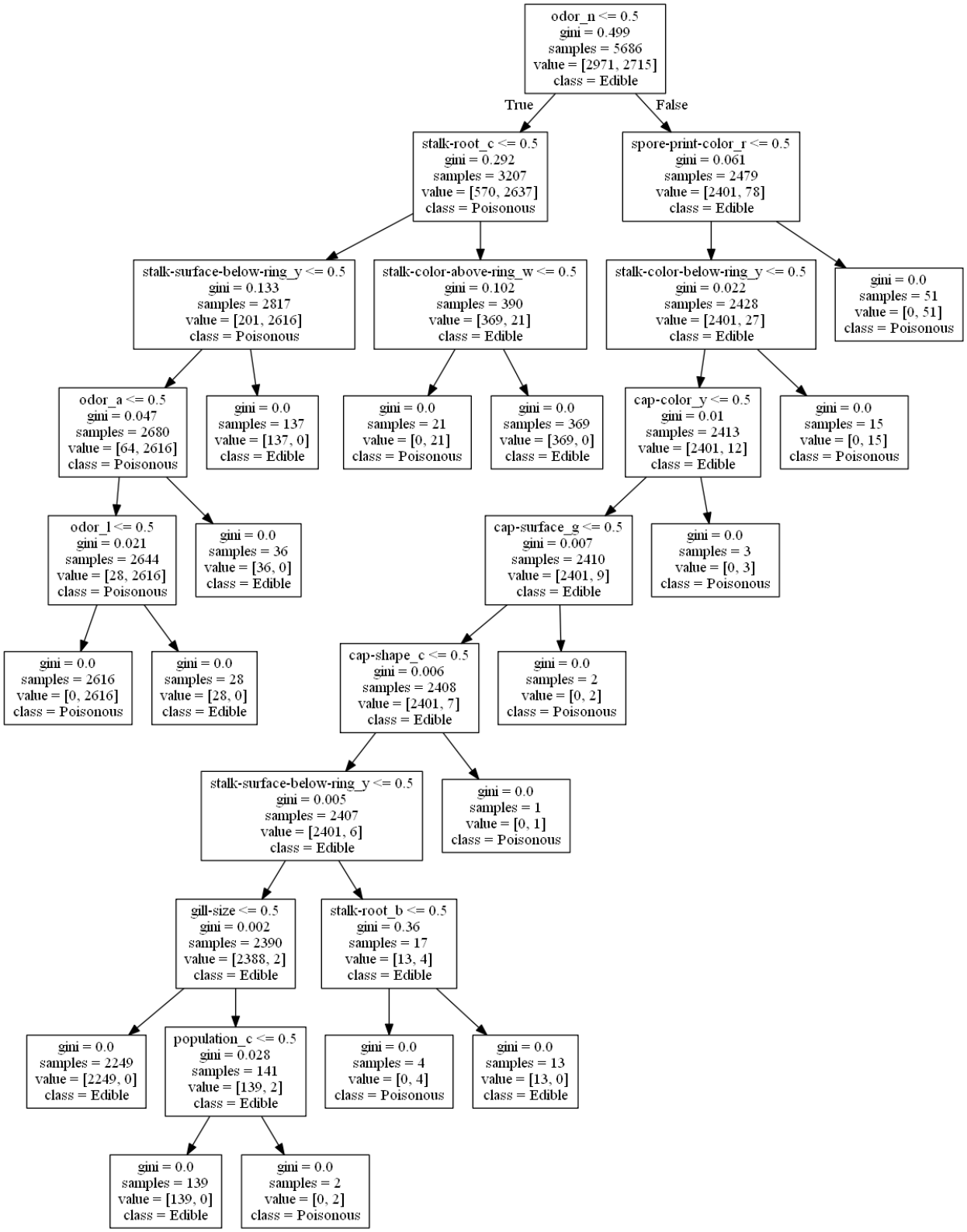
Looking at this output for unique values in each column, we see that veil-type only has 1 unique item. This means that is would be useless for classifying the mushroom, as all the mushrooms in this dataset has that veil-type. We can drop this column using:

```
data.drop(columns='veil-type')
```

I wanted to ensure that our data was not being overfitted by having all of the parameters, which is why I decided to find the most important features and only use those for the predictive model.



I also did some testing with a decision classification tree and achieved this tree that can help classify the mushroom. This can be put into someone's pocket when going out foraging.


odor_n <= 0.5
gini = 0.499
samples = 5686
value = [2971, 2715]
class = Edible
True
False
stalk-root_c <= 0.5
gini = 0.292
samples = 3207
value = [570, 2637]
class = Poisonous
spore-print-color_r <= 0.5
gini = 0.061
samples = 2479
value = [2401, 78]
class = Edible
stalk-surface-below-ring_y <= 0.5
gini = 0.133
samples = 2817
value = [201, 2616]
class = Poisonous
stalk-color-above-ring_w <= 0.5
gini = 0.102
samples = 390
value = [369, 21]
class = Edible
stalk-color-below-ring_y <= 0.5
gini = 0.022
samples = 2428
value = [2401, 27]
class = Edible
gini = 0.0
samples = 51
value = [0, 51]
class = Poisonous
odor_a <= 0.5
gini = 0.047
samples = 2680
value = [64, 2616]
class = Poisonous
gini = 0.0
samples = 137
value = [137, 0]
class = Edible
gini = 0.0
samples = 21
value = [0, 21]
class = Poisonous
gini = 0.0
samples = 369
value = [369, 0]
class = Edible
cap-color_y <= 0.5
gini = 0.01
samples = 2413
value = [2401, 12]
class = Edible
gini = 0.0
samples = 15
value = [0, 15]
class = Poisonous
odor_l <= 0.5
gini = 0.021
samples = 2644
value = [28, 2616]
class = Poisonous
gini = 0.0
samples = 36
value = [36, 0]
class = Edible
cap-surface_g <= 0.5
gini = 0.007
samples = 2410
value = [2401, 9]
class = Edible
gini = 0.0
samples = 3
value = [0, 3]
class = Poisonous
gini = 0.0
samples = 2616
value = [0, 2616]
class = Poisonous
gini = 0.0
samples = 28
value = [28, 0]
class = Edible
cap-shape_c <= 0.5
gini = 0.006
samples = 2408
value = [2401, 7]
class = Edible
gini = 0.0
samples = 2
value = [0, 2]
class = Poisonous
stalk-surface-below-ring_y <= 0.5
gini = 0.005
samples = 2407
value = [2401, 6]
class = Edible
gini = 0.0
samples = 1
value = [0, 1]
class = Poisonous
gill-size <= 0.5
gini = 0.002
samples = 2390
value = [2388, 2]
class = Edible
stalk-root_b <= 0.5
gini = 0.36
samples = 17
value = [13, 4]
class = Edible
gini = 0.0
samples = 2249
value = [2249, 0]
class = Edible
population_c <= 0.5
gini = 0.028
samples = 141
value = [139, 2]
class = Edible
gini = 0.0
samples = 4
value = [0, 4]
class = Poisonous
gini = 0.0
samples = 13
value = [13, 0]
class = Edible
gini = 0.0
samples = 139
value = [139, 0]
class = Edible
gini = 0.0
samples = 2
value = [0, 2]
class = Poisonous

Code

Originally, the dataset was incredibly clean and not much needed to be done to clean the data. Only one column/category was to be dropped due to only having one unique value. I also removed the “class” column because that is what is to be predicted.

Python has multiple libraries at its disposal such as Pandas to read and manipulate the data from the CSV file. Initial data was given in single character formats, therefore it had to be encoded to numbers to run the machine learning algorithm. Multiple algorithms were run and used to determine which model would have the best results. Recursive Feature Elimination ranked the variables/categories that were to be eliminated because they were of low importance. To prevent overfitting from having too many parameters, the most important features were found and only the top four were used. NumPy, another library used in Python helped clean up the dataset.

The main machine learning model used was the Random Forest Classifier that split up the dataset into 30 different trees. Each tree determines whether the mushroom is poisonous or edible. I decided to use the Random Forest Classifier because it is greatly used for as a binary classifier: yes or no, 1 or 0. In the case of classifying mushrooms: edible or poisonous.

Hypothesis Assessment

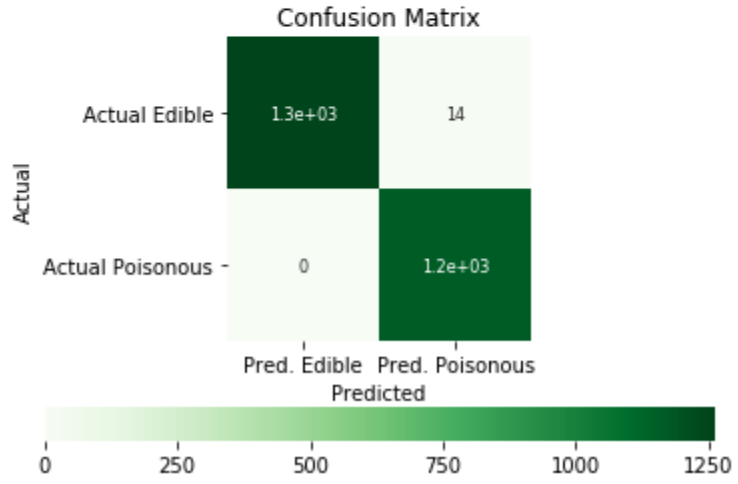
Hypothesis: A predictive model will be able to identify and classify whether a mushroom is poisonous or edible.

The model was trained on 30% of the data and achieved the following results below. Even on 10% of the data, it achieved an accuracy score of 99%.

The hypothesis of whether a machine learning model can predict whether a mushroom is poisonous or edible has strong support.

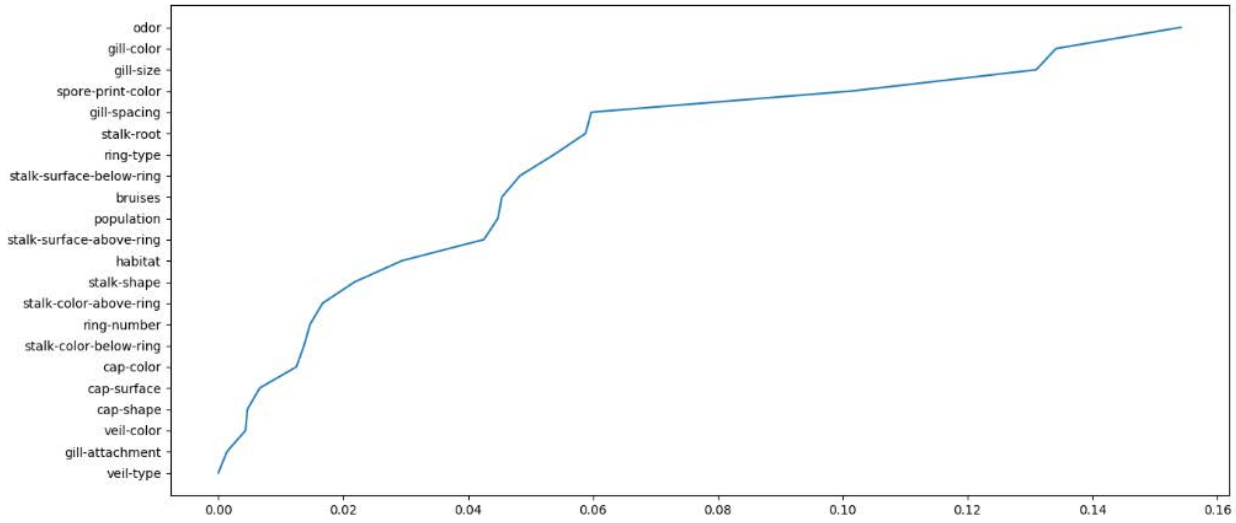
Accuracy 0.9942575881870386

	precision	recall	f1-score	support
0	1.00	0.99	0.99	1274
1	0.99	1.00	0.99	1164
accuracy			0.99	2438
macro avg	0.99	0.99	0.99	2438
weighted avg	0.99	0.99	0.99	2438

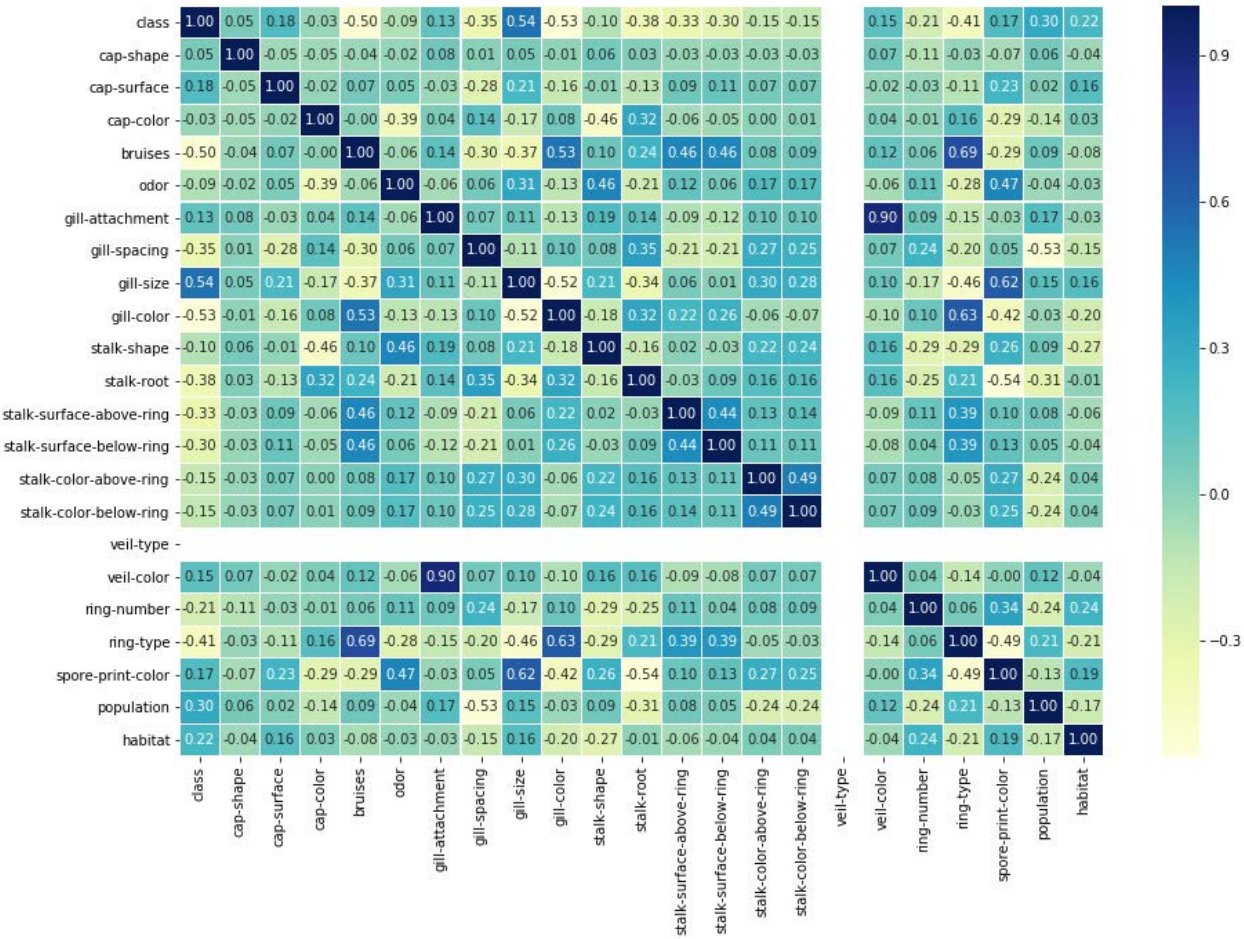


Visualization and Data Display

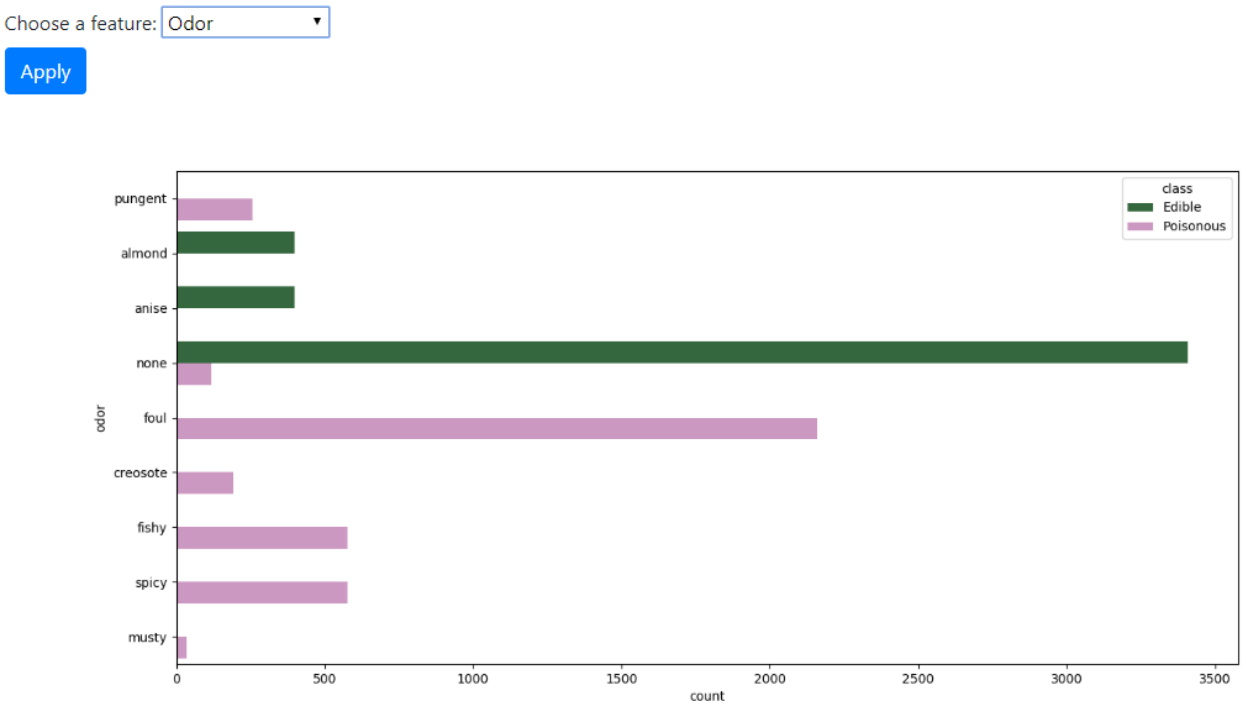
Data is visualized through the Visualizations page. The line graph shows the most important features to classify the mushroom which is used in the predictive model. This helped prevent overfitting of data when training the machine learning model.



Notice how the top four features help classify over 10% of the mushroom.



The correlation matrix shows how strong each category meshes well with other categories. I can see from veil-type that it is completely whited out because it doesn't help classify the mushroom at all.



The bar chart feature count helps visualize how many mushrooms are classified as poisonous or edible based off a specific feature. This gives the user a greater idea and can eliminate certain mushrooms from the start.

Accuracy Assessment

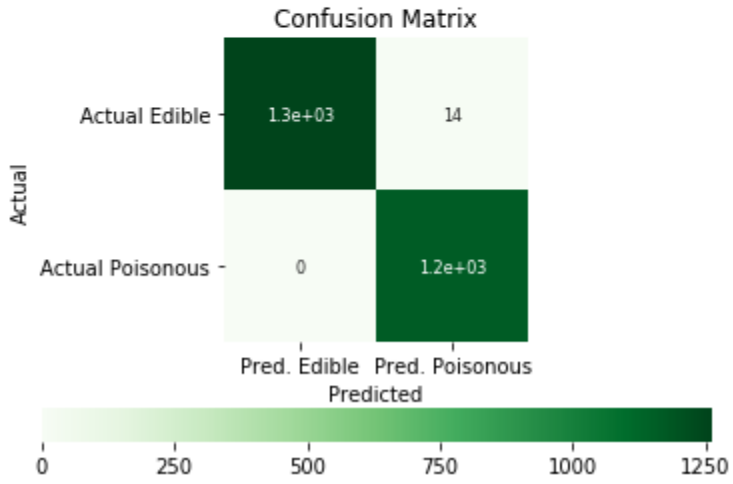
The accuracy of the product is determined by the results achieved on the test dataset using a library for Python called Scikit-Learn. It can be visualized in a confusion matrix below.

```
# Split the data into training and testing sets
train_X, test_X, train_y, test_y = train_test_split(
    X,
    y,
    test_size=0.3,
    random_state=101
)
```

```
Accuracy 0.9942575881870386
              precision    recall  f1-score   support

           0       1.00      0.99      0.99      1274
           1       0.99      1.00      0.99      1164

    accuracy                           0.99      2438
   macro avg       0.99      0.99      0.99      2438
weighted avg       0.99      0.99      0.99      2438
```



Other tests were used such as K-Means clustering and Decision Tree Classification, but the K-Means clustering only had about an 88% accuracy rate. K-Means clustering would occasionally flip the edible and poisonous features, which sometimes make the accuracy be about 10% or less which became incredibly unreliable. The Decision Tree Classification also had about a 99% accuracy rate which is on par with the Random Forest Classifier.

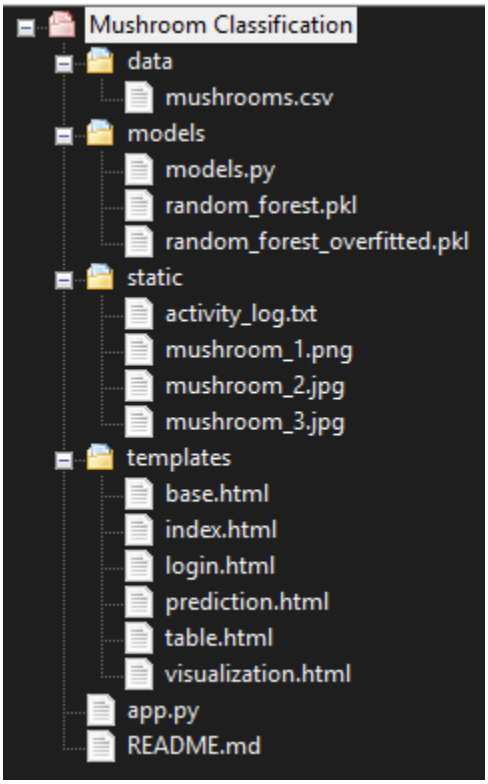
The model can be retrained with the models.py file to include more features or a new test dataset.

Application Testing

Unit testing was performed on the dataset to ensure that data was being parsed correctly. I wanted to ensure that the data was in the correct state before implementing other visualizations that would be using the data. I also wanted to ensure that I was reaching a high accuracy with the machine learning model; I didn't want a model that wasn't above 80%. The confusion matrix helped with understanding how well the machine learning model was working.

After data testing was finished, I decided to implement the web application. Originally, I was going to allow the user to type in input but realized how error prone this could get. I opted to go for drop down values that allowed users to only select specific values. This data is then passed using JavaScript as a POST request to retrieve a JSON object that will display the message of whether the mushroom is edible or poisonous.

Application Files



The file structure can be examined above.

- data
  - mushrooms.csv
    - The CSV file containing the dataset
- models
  - models.py
    - The Python script containing the predictive machine learning models.
  - random_forest.pkl
    - The pickle file that is used to save machine learning models.
  - random_forest_overfitted.pkl

        - The pickle file that is used to save machine learning models.

- static
    - activity_log.txt
        - Contains the logging of website activity for monitoring health.
    - mushroom_#.png
        - Any additional images that are to be added to the website should be contained inside the static folder.
- templates
    - Contains the .html files to navigate through the website.
- app.py
    - Main entry-point of the application.
- requirements.txt
    - Needed to get the environment setup to run the application properly

User Guide

The application is submitted via a .zip archive. Extract the folder onto your machine. Python 3.7 is required to run the application along with a pip3 package manager.

Ensure you are in the working directory of the main project and run this command:

- py -m venv env

Let the packages install. You now want to move into the virtual environment using this command:

- env\Scripts\activate

This will start you up in a virtual environment (you should see your CLI change to something like this:

```
(env) C:\Users\
```

The (env) lets you know you are in the virtual environment. Now run the command:

- pip install -r requirements.txt

Let all packages install. Once all packages have been installed, run either of these commands:

- py app.py
- python3 app.py

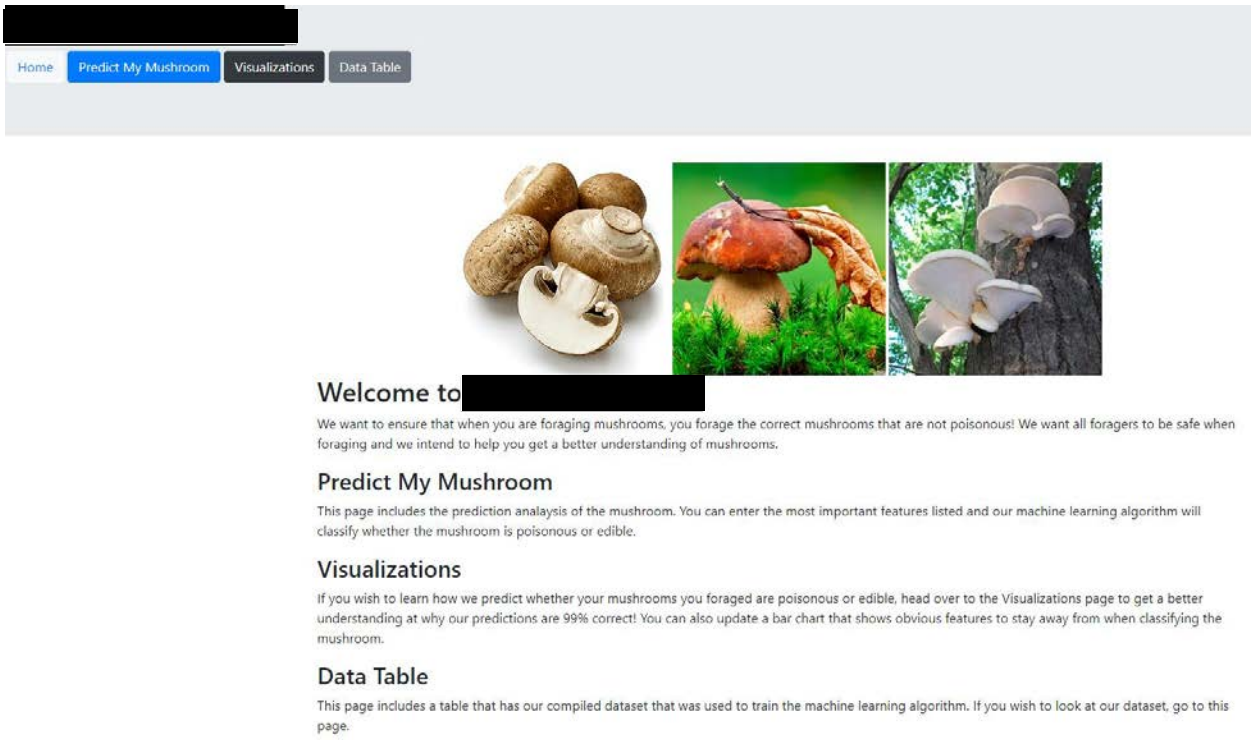
You will see the application has started and see a message that says, “Running on http://[redacted]”. Open any browser and navigate to the URL. You will be directed to the login page. The following credentials can be used to login:

Username: admin

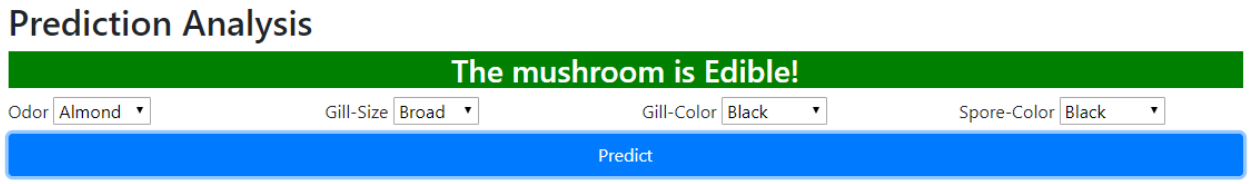
Password: admin

Note, these credentials should not be used for the production environment.

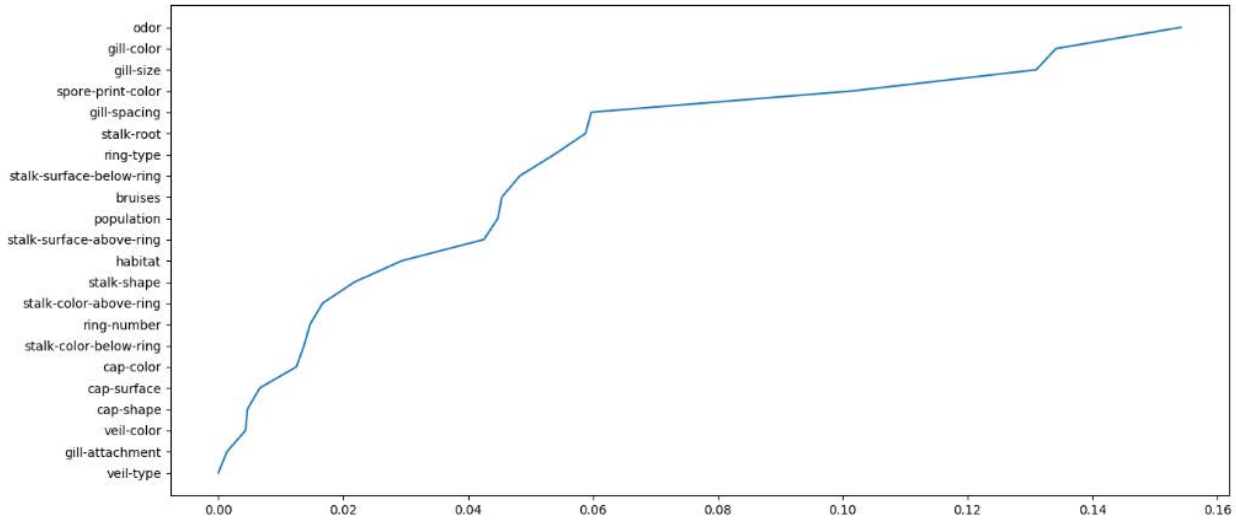
After login, you will be on the home page that has a description containing each webpage available.

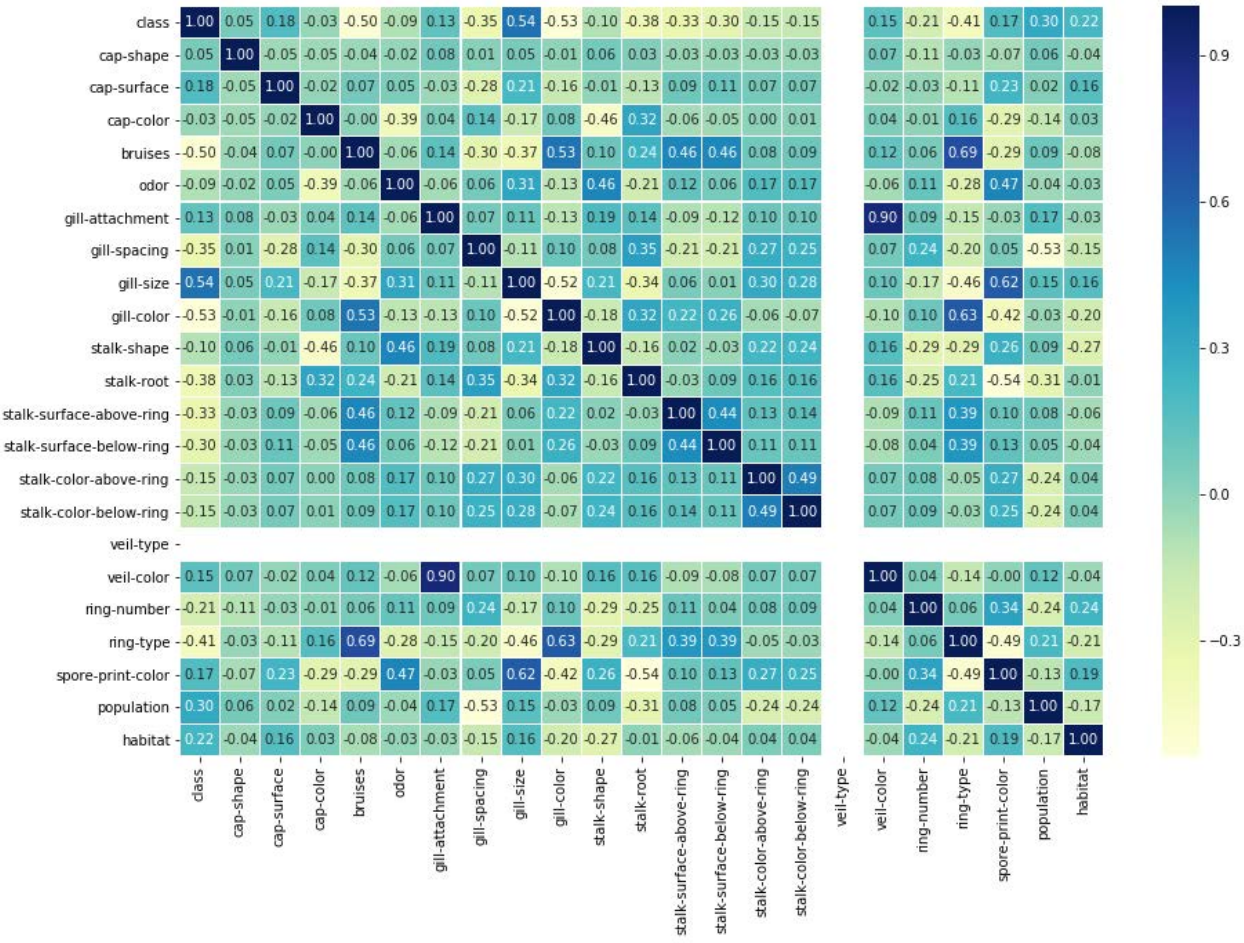


The user interface is meant to be intuitive and easy to use. If you wish to predict your mushroom, click the button “Predict My Mushroom”. This will allow you to select the features for your mushroom and get a prediction.



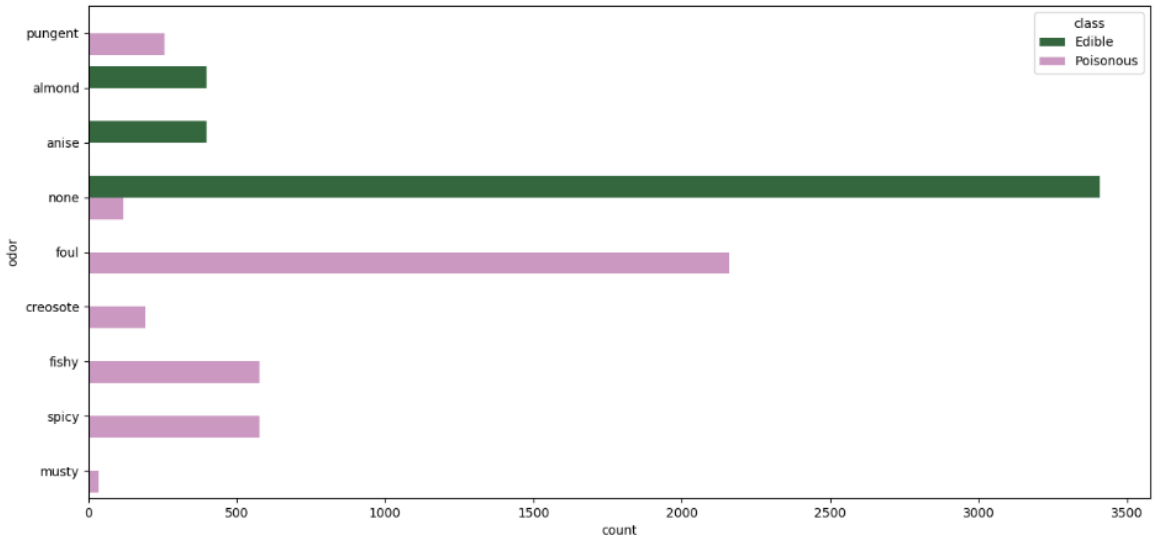
For visualizations and an interactive bar chart, head over to “Visualizations”.


odor
gill-color
gill-size
spore-print-color
gill-spacing
stalk-root
ring-type
stalk-surface-below-ring
bruises
population
stalk-surface-above-ring
habitat
stalk-shape
stalk-color-above-ring
ring-number
stalk-color-below-ring
cap-color
cap-surface
cap-shape
veil-color
gill-attachment
veil-type
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16


class 1.00 0.05 0.18 -0.03 -0.50 -0.09 0.13 -0.35 0.54 -0.53 -0.10 -0.38 -0.33 -0.30 -0.15 -0.15 0.15 -0.21 -0.41 0.17 0.30 0.22
cap-shape 0.05 1.00 -0.05 -0.05 -0.04 -0.02 0.08 0.01 0.05 -0.01 0.06 0.03 -0.03 -0.03 -0.03 -0.03 0.07 -0.11 -0.03 -0.07 0.06 -0.04
cap-surface 0.18 -0.05 1.00 -0.02 0.07 0.05 -0.03 -0.28 0.21 -0.16 -0.01 -0.13 0.09 0.11 0.07 0.07 -0.02 -0.03 -0.11 0.23 0.02 0.16
cap-color -0.03 -0.05 -0.02 1.00 -0.00 -0.39 0.04 0.14 -0.17 0.08 -0.46 0.32 -0.06 -0.05 0.00 0.01 0.04 -0.01 0.16 -0.29 -0.14 0.03
bruises -0.50 -0.04 0.07 -0.00 1.00 -0.06 0.14 -0.30 -0.37 0.53 0.10 0.24 0.46 0.46 0.08 0.09 0.12 0.06 0.69 -0.29 0.09 -0.08
odor -0.09 -0.02 0.05 -0.39 -0.06 1.00 -0.06 0.06 0.31 -0.13 0.46 -0.21 0.12 0.06 0.17 0.17 -0.06 0.11 -0.28 0.47 -0.04 -0.03
gill-attachment 0.13 0.08 -0.03 0.04 0.14 -0.06 1.00 0.07 0.11 -0.13 0.19 0.14 -0.09 -0.12 0.10 0.10 0.90 0.09 -0.15 -0.03 0.17 -0.03
gill-spacing -0.35 0.01 -0.28 0.14 -0.30 0.06 0.07 1.00 -0.11 0.10 0.08 0.35 -0.21 -0.21 0.27 0.25 0.07 0.24 -0.20 0.05 -0.53 -0.15
gill-size 0.54 0.05 0.21 -0.17 -0.37 0.31 0.11 -0.11 1.00 -0.52 0.21 -0.34 0.06 0.01 0.30 0.28 0.10 -0.17 -0.46 0.62 0.15 0.16
gill-color -0.53 -0.01 -0.16 0.08 0.53 -0.13 -0.13 0.10 -0.52 1.00 -0.18 0.32 0.22 0.26 -0.06 -0.07 -0.10 0.10 0.63 -0.42 -0.03 -0.20
stalk-shape -0.10 0.06 -0.01 -0.46 0.10 0.46 0.19 0.08 0.21 -0.18 1.00 -0.16 0.02 -0.03 0.22 0.24 0.16 -0.29 -0.29 0.26 0.09 -0.27
stalk-root -0.38 0.03 -0.13 0.32 0.24 -0.21 0.14 0.35 -0.34 0.32 -0.16 1.00 -0.03 0.09 0.16 0.16 0.16 -0.25 0.21 -0.54 -0.31 -0.01
stalk-surface-above-ring -0.33 -0.03 0.09 -0.06 0.46 0.12 -0.09 -0.21 0.06 0.22 0.02 -0.03 1.00 0.44 0.13 0.14 -0.09 0.11 0.39 0.10 0.08 -0.06
stalk-surface-below-ring -0.30 -0.03 0.11 -0.05 0.46 0.06 -0.12 -0.21 0.01 0.26 -0.03 0.09 0.44 1.00 0.11 0.11 -0.08 0.04 0.39 0.13 0.05 -0.04
stalk-color-above-ring -0.15 -0.03 0.07 0.00 0.08 0.17 0.10 0.27 0.30 -0.06 0.22 0.16 0.13 0.11 1.00 0.49 0.07 0.08 -0.05 0.27 -0.24 0.04
stalk-color-below-ring -0.15 -0.03 0.07 0.01 0.09 0.17 0.10 0.25 0.28 -0.07 0.24 0.16 0.14 0.11 0.49 1.00 0.07 0.09 -0.03 0.25 -0.24 0.04
veil-type
veil-color 0.15 0.07 -0.02 0.04 0.12 -0.06 0.90 0.07 0.10 -0.10 0.16 0.16 -0.09 -0.08 0.07 0.07 1.00 0.04 -0.14 -0.00 0.12 -0.04
ring-number -0.21 -0.11 -0.03 -0.01 0.06 0.11 0.09 0.24 -0.17 0.10 -0.29 -0.25 0.11 0.04 0.08 0.09 0.04 1.00 0.06 0.34 -0.24 0.24
ring-type -0.41 -0.03 -0.11 0.16 0.69 -0.28 -0.15 -0.20 -0.46 0.63 -0.29 0.21 0.39 0.39 -0.05 -0.03 -0.14 0.06 1.00 -0.49 0.21 -0.21
spore-print-color 0.17 -0.07 0.23 -0.29 -0.29 0.47 -0.03 0.05 0.62 -0.42 0.26 -0.54 0.10 0.13 0.27 0.25 -0.00 0.34 -0.49 1.00 -0.13 0.19
population 0.30 0.06 0.02 -0.14 0.09 -0.04 0.17 -0.53 0.15 -0.03 0.09 -0.31 0.08 0.05 -0.24 -0.24 0.12 -0.24 0.21 -0.13 1.00 -0.17
habitat 0.22 -0.04 0.16 0.03 -0.08 -0.03 -0.03 -0.15 0.16 -0.20 -0.27 -0.01 -0.06 -0.04 0.04 0.04 -0.04 0.24 -0.21 0.19 -0.17 1.00
class
cap-shape
cap-surface
cap-color
bruises
odor
gill-attachment
gill-spacing
gill-size
gill-color
stalk-shape
stalk-root
stalk-surface-above-ring
stalk-surface-below-ring
stalk-color-above-ring
stalk-color-below-ring
veil-type
veil-color
ring-number
ring-type
spore-print-color
population
habitat
0.9
0.6
0.3
0.0
−0.3

Choose a feature: Odor

Apply



To see the dataset and use an interactive table for data exploration, head over to “Data Table.”

	class	cap-shape	cap-surface	cap-color	
Search	Search class	Search cap-shape	Search cap-surface	Search cap-color	Search bru
0	Poisonous	convex	smooth	brown	bruises
1	Edible	convex	smooth	yellow	bruises
2	Edible	bell	smooth	white	bruises
3	Poisonous	convex	scaly	white	bruises
4	Edible	convex	smooth	gray	no
5	Edible	convex	scaly	yellow	bruises
6	Edible	bell	smooth	white	bruises
7	Edible	bell	scaly	white	bruises
8	Poisonous	convex	scaly	white	bruises
9	Edible	bell	smooth	yellow	bruises

Bibliography

Mohamed Sami (2018, November 10). The Waterfall Model, a different perspective. Retrieved from https://melsatar.blog/2018/02/16/the-waterfall-model-a-different-perspective/

UCI Repository (1987). *Mushroom Data Set*. Retrieved from https://archive.ics.uci.edu/ml/datasets/mushroom

Verbanas, P. (2013, September 26). Poisonous mushrooms pose danger as more people forage for locally grown food, experts say. Retrieved from https://medicalxpress.com/news/2013-09-poisonous-mushrooms-pose-danger-people.html

Voa (2018, November 21). Mushroom Hunting Gains Popularity in US. Retrieved from https://learningenglish.voanews.com/a/mushroom-hunting-gains-popularity-in-u-s-/4662025.html

Yiu, T. (2019, August 14). Understanding Random Forest. Retrieved from https://towardsdatascience.com/understanding-random-forest-58381e0602d2