1. Introduction: (3 pts) a brief overview of the problem, your methodology and your final results

Exploring the wilderness of *Mushroomia*, I would like to be able to identify which mushrooms are edible, in order to forage supplies for my daily mushroom soup. Using the data from the *National Archives on Mushrooms*, I trained machine learning models to help predict whether mushroom data collected from the *Shroomster Pro Max* indicates a poisonous or edible mushroom. In order to determine which features to keep, I used various diagrams, and added a few new features myself. I pipelined all of the features to normalize them and convert the categorical data into one hot vector form. I then reduced the dimensions of my data using PCA because I found out my model was extremely overfitted. Then, I used Logistic Regression, KNN, Decision Tree, and Random Forest classifiers to model the data. Lastly, I used cross validation for hyperparameter tuning for KNN, Decision Tree, and Random Forest classification. The model with the highest accuracy was KNN classifier at 61% accuracy prior to tuning.

2. Methodology:
(a) Data Loading, Splitting, Exploration and Visualization: (6 pts) what did you explore / visualize? why did you explore / visualize it in this way? what did you learn from this about the data?

I visualized distribution of the numerical data and displayed the categorical variables in bar plot form. The bar plots for the categorical data helped in determining which features did not have much variation. A feature that did not have a lot of variation, meaning most of its data was one category, would not be very useful in classifying most of the data. I used a correlation map to help visualize which features were correlated with each other. I also removed some of the features that had high correlation, because they are redundant and needlessly complicate the model. From the data I learned that a lot of the graphs are very skewed, making it hard to determine whether a feature was worth keeping or not. The numerical data such as cap-diameter and stem-width tended to lean towards the lower side.

(b) Data Pre-Processing: (4 pts) how did you pre-process your data? why did you pre-process your data in this way?

I dropped some features such as veil-type or veil-color because they did not have a lot of variation, meaning they would not be very effective at classifying most data points. Other features such as spore-print-color or stem-root had null values for too many data points, so I removed those features as well. For the remaining null values I just removed

them from the training data. I wanted to balance dropping features and keeping a decent enough amount of training data to help train the models.

(c) Data Augmentation: (4 pts) what additional features did you create? why do you think they are useful?

The additional features I created are cap diameter divided by stem width and cap diameter divided by stem height. Because I did this before Data Pre-processing, the numerical data is not normalized, meaning addition or subtraction does not make as much sense. Thus, I did a multiplication and division of every possible combination of the 3 numerical features. I chose the best 2 by using a correlation plot and choosing the 2 features with the least correlation with the other features to ensure they are not replaceable by any of the other features and add a unique metric for classification.

(d) Statistical Hypothesis Testing: (10 pts) what relationships between variables did you investigate? were the results statistically significant? what did you learn from this about the data?

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.2328	0.038	-32.352	0.000	-1.307	-1.158
stem-height	0.0352	0.005	7.527	0.000	0.026	0.044
stem-width	0.0146	0.003	4.624	0.000	0.008	0.021
cap-diam-div-stem-width	0.3881	0.084	4.618	0.000	0.223	0.553
cap-diam-div-stem-height	0.3517	0.043	8.225	0.000	0.268	0.435

I investigated the relationship between each of the numerical features with the prediction. This includes stem-height, stem-width, cap-diam-div-stem-width, and cap-diam-div-stem-height. The most important thing I looked out for was the confidence interval. If any of the confidence intervals include zero, that means the feature does not have a significant impact on the classification of the mushroom. None of the intervals included zero, meaning that all of my features were statistically significant. I learned that all of my features do impact the classification of the mushroom.

(e) Models of your choice (2 distinct models): (6 pts) what models did you choose to implement? why are they appropriate for the problem?

I used KNN and Decision trees as my models. KNN is appropriate because it is simple and is able to achieve a high accuracy. It is also very adaptable to overfitting or underfitting depending on the number of neighbors it looks at. KNN is also entirely dependent on the distribution of the data. Decision trees are a very different type of classifier. It creates a tree of rules to determine if a mushroom is poisonous or not. Decision trees were appropriate, as they show how a different type of classifier would perform on the data. They are also easily able to adapt to too many or too little features depending on the size of the trees. KNN also tends to perform better on smaller data sets, while Decision trees tend to perform better on larger data sets. This discrepancy means that at least one of the two is likely to perform well on the data.

(f) Ensemble Method (1 ensemble method): (3 pts) what ensemble method did you use? why is it a good fit for this problem?

I chose Random Forest Classifier because it is essentially a more robust Decision Tree Classifier. The decision tree classifier had the higher recall value, so I thought the Random Forest would perform even better. Also by taking the average of multiple samples, the Random Forest Classifier could help prevent overfitting, as I saw was happening to my K Nearest Neighbors classifiers. It is also very easy to adjust the Random Forest Classifier by altering its tree size, similar to the Decision Tree.

(g) Hyper-parameter Tuning: (3 pts) how did you tune the hyperparameter for each model? what were the best parameters you found?

I used Grid Search for the tuning of all the hyperparameters. For each of the models I looked at the documentation for different possible parameters. The numbers I chose were either arbitrary or based on recommendations from the documentation. For K Nearest Neighbors the most optimal parameters were: n_neighbors=5, weights=distance, algorithm=brute. For the Decision Tree the best parameters were: max_depth=8, criterion=gini. For the Random Forest Classifier, the best parameters were: n_estimators=15, max_features=None. Because I printed out all of the parameters, the GridSearch also printed out many other parameters.

3. Results: (4 pts) Compare the performance of each of the models you implemented using the standard evaluation metrics. Discuss the cross-validation strategy you used and why you chose this strategy.
before:

KNN:
Accuracy: 0.611551

Precision: 0.725606
Recall: 0.516103
F1: 0.603181
Decision Tree:
Accuracy: 0.571942
Precision: 0.650607
Recall: 0.543639
F1: 0.592333
Random Forest:
Accuracy: 0.591194
Precision: 0.766386
Recall: 0.410467
F1: 0.534606

After Cross-val
KNN:
Accuracy: 0.462878
Precision: 0.540535
Recall: 0.406924
F1: 0.464309
Decision Tree:
Accuracy: 0.506448
Precision: 0.638492
Recall: 0.316264
F1: 0.423002
Random Forest:
Accuracy: 0.545965
Precision: 0.690116
Recall: 0.374396
F1: 0.485437

We can see that KNN performs extremely well when evaluating specifically against the test set. However, once cross validation is done, we see that the KNN model actually is not that generalizable. As predicted, the Random Forest Classifier performed significantly better than Decision Tree before and after cross validation. The Cross Validation strategy I did was 10-fold cross validation. I chose this number because it is a middle ground that ensures that at least 90% of the data is being used to train and will still complete in a timely manner.

4. Conclusion: (7 pts) Decide which model you would use on your adventure through Mushroomia. Explain why you chose this model based on the results. Discuss any limitations of your project (e.g. limitations of model, data analysis and visualization, concerns with the raw data in National Archives on Mushrooms).

If I had to take a model with me based on the current parameters, I would take the Random Forest model after Cross Validation. After the cross validation, the model is more generalizable on newer data than before, meaning it is better than the previous models for my purpose. I also choose Random Forest primarily because it has the highest precision value of 69%. The precision score indicates how many of my edible mushroom predictions are actually edible. This is important because misclassifying a poisonous mushroom as edible is much worse than misclassifying a non-poisonous mushroom.

There are a couple of limitations with my models. I could have analyzed the parameters more to help narrow down the features more. Instead I just used PCA to reduce the dimensions for me. Another limitation is time. If I had more time I could perform a far more robust GridSearch to find much better parameters for my models than the values that I chose arbitrarily.