Final Project Report

**Important Information:**

* This project was completed in a team of 3, including Gerom Pagaduan, Dan Reuter, and myself.
* My role was to test three different classifiers and discover key information about the mushrooms and their features
* The three classifiers I used were KNN, SVM, and Logistic Regression. The purpose of selecting each of these models was to have as many unique classifiers to discover which classifiers preformed the best. My teammates focused on other techniques, such as Neural Networks
* This project is actually built off of a lot of my previous work, primarily my midterm that took a similar testing approach but with a simpler dataset. This project is far more thorough, optimized, and required applying concepts such a preconditioning data (more on that later).
* Because it is based on work from the class, the code I based this project on originated from Dr. Jiang, specifically the code used for the classifiers in the three different methods
* Packages:
  + Sklearn: KNN, SVM, and Logistic Regression classifiers. For data preparation: label encoder, and train\_test\_split. For test results: classification\_report, accuracy\_score, and confusion\_matrix
  + Numpy: mostly for the use of numpy-arrays
  + Pandas: almost all dataframe manipulation

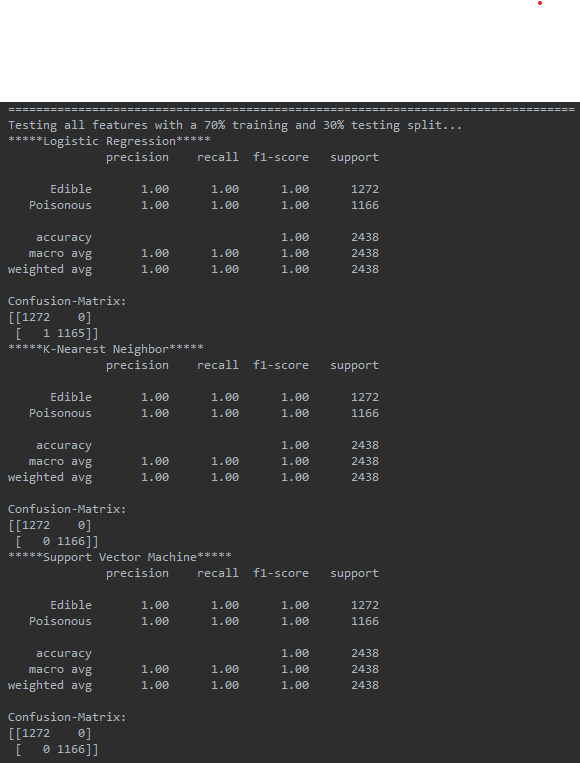
**Project Details:**

* This project uses utilizes a dataset of 8124 mushrooms with 22 categorical features. The classification of these mushrooms is whether or not the mushroom is poisonous or edible
* The original location we received the dataset from was at this location on the Kaggle website: <https://www.kaggle.com/uciml/mushroom-classification>
* Features include: 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, and habitat
* Extensive data preprocessing was required to use this dataset. With several categories within each feature (Some had as many as 6 categories in one column), we knew that we needed to change the data into something numerical in order to read this data and preform learning on it. Gerom and Dan decided to try and change the letters in each category to a unique number, such as a = 1, b = 2, c = 3, etc. However, I decided to take the data-frame and convert each feature into dummy variables. What this means is each column is split into k-1 new columns where k is the number of categories inside that column. The new columns that are generated represent binary true/false values for whether or not that specific sample has that feature. For example, a cap-shape has 6 different cap shape categories, bell=b, conical=c, convex=x, flat=f, knobbed=k, and sunken=s. Dummying this one column results in 5 new columns with binary values. The reason it is always k-1 values is because the last category is implied if all the other categories are false.
* Dummying the data had significantly better accuracy scores than changing each value to unique numbers, but the model still scored pretty well with the unique number style despite the unique number approach being more of a brute force approach to data preprocessing.
* Once the dataset is dummied, the original 22 features turn into 95 columns of binary values. Another interesting observation is that one of our categories, veil-type has only one value in its column and when dummied is removed from the generated data-frame. This makes sense because k=1 and dummying generates k-1 columns. So the dummying results in either an empty data-frame or is removed from a larger data-frame. Because this column has no variability, it also represents a constant and does not help our model learn.
* The first half of this project first tests several different ratios of training and testing data. There is a list at the beginning of this project that allows for as many different combinations to be tested where the list is the list of all the desired training data percentages (testing is always 100% - each training %)

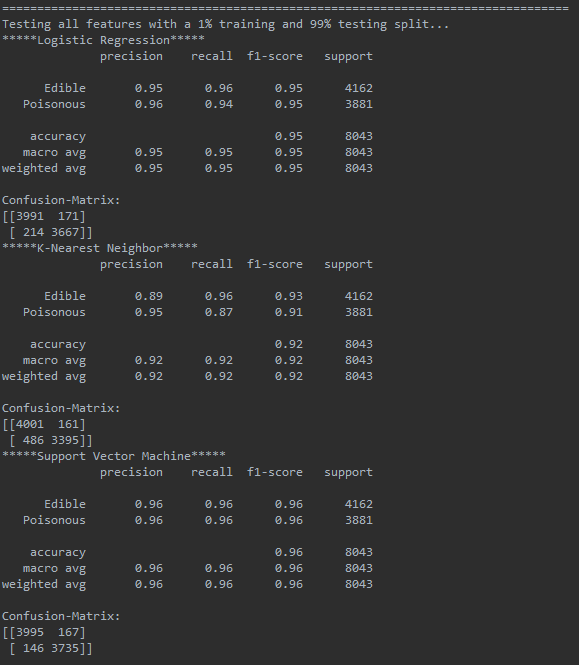
69 training\_percentages = [.7, .4, .01]

* The second half of this project tests each feature individually. The decision to add this analysis was influenced by how high my testing accuracy was with all three classifiers. In fact, even with a 1% training split and 99% testing, the models all performed around 95% accuracy which is astounding. This led me to believe that there is a key feature in the data that is the culprit of such high accuracy. Upon further investigation, I was correct.

Results with a70% training and 30% testing

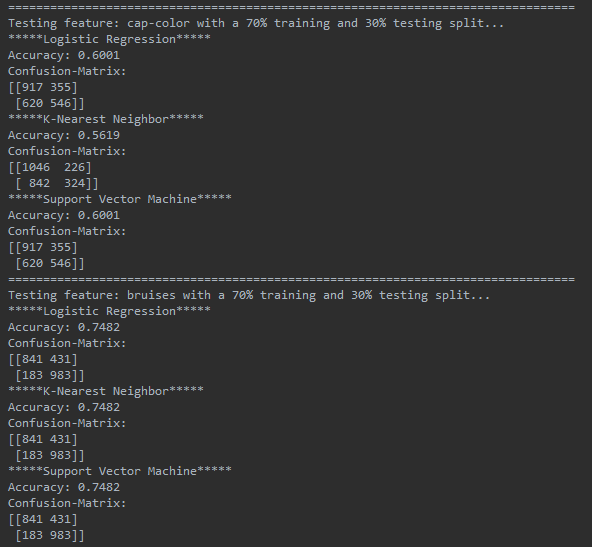


Results with a1% training and 99% testing



* With the feature test, each of the 22 features are split into individual data-frames that are then dummied into their respective binary forms. Each data-frame is stored separately in a dictionary with the key being the original feature name.
* Testing is performed on each of the data-frames with all three models to test their importance to classifying the mushrooms. The testing split is 30% training and 70% testing

Example of some of the results from the features test



**Results:**

After preforming the features test, it is became clear why the models were so accurate despite small training sizes. The three features that resulted in the best accuracies were Odor, spore print color, and gill color. With just odor alone, the models were all able to achieve around 98% accuracy without any other features (which is crazy!), followed by a score of 88% from spore-print color and 80% from gill color. With the performance of the models, it is still evident that SVM is still the best preforming classifier with small training percentages with Logistic regression second and KNN last. However, the differences in accuracy are pretty small and once a substantial amount of data is used, the models all perform about the same.

The individual feature test results sorted by SVM accuracy are as follows:

1. odor, average accuracy: 0.9865
2. spore-print-color, average accuracy: 0.8819
3. gill-color, average accuracy: 0.8076
4. ring-type, average accuracy: 0.7904
5. stalk-surface-above-ring, average accuracy: 0.7797
6. stalk-surface-below-ring, average accuracy: 0.774
7. gill-size, average accuracy: 0.7506
8. bruises, average accuracy: 0.7482
9. population, average accuracy: 0.7281
10. stalk-color-above-ring, average accuracy: 0.726
11. stalk-color-below-ring, average accuracy: 0.7223
12. habitat, average accuracy: 0.6932
13. stalk-root, average accuracy: 0.6468
14. gill-spacing, average accuracy: 0.6079
15. cap-color, average accuracy: 0.6001
16. cap-surface, average accuracy: 0.5726
17. cap-shape, average accuracy: 0.5722
18. stalk-shape, average accuracy: 0.5455
19. ring-number, average accuracy: 0.5304
20. veil-color, average accuracy: 0.5226
21. gill-attachment, average accuracy: 0.5217
22. veil-type, average accuracy: NONE, not tested because it has only one category

Full Test Results

Generating dummy variable table...  
 class\_p cap-shape\_c cap-shape\_f ... habitat\_p habitat\_u habitat\_w  
0 1 0 0 ... 0 1 0  
1 0 0 0 ... 0 0 0  
2 0 0 0 ... 0 0 0  
3 1 0 0 ... 0 1 0  
4 0 0 0 ... 0 0 0  
... ... ... ... ... ... ... ...  
8119 0 0 0 ... 0 0 0  
8120 0 0 0 ... 0 0 0  
8121 0 0 1 ... 0 0 0  
8122 1 0 0 ... 0 0 0  
8123 0 0 0 ... 0 0 0  
  
[8124 rows x 96 columns]  
  
Splitting data into X and Y components...  
 cap-shape\_c cap-shape\_f cap-shape\_k ... habitat\_p habitat\_u habitat\_w  
0 0 0 0 ... 0 1 0  
1 0 0 0 ... 0 0 0  
2 0 0 0 ... 0 0 0  
3 0 0 0 ... 0 1 0  
4 0 0 0 ... 0 0 0  
... ... ... ... ... ... ... ...  
8119 0 0 1 ... 0 0 0  
8120 0 0 0 ... 0 0 0  
8121 0 1 0 ... 0 0 0  
8122 0 0 1 ... 0 0 0  
8123 0 0 0 ... 0 0 0  
  
[8124 rows x 95 columns]  
0 1  
1 0  
2 0  
3 1  
4 0  
 ..  
8119 0  
8120 0  
8121 0  
8122 1  
8123 0  
Name: class\_p, Length: 8124, dtype: uint8  
=================================================================================  
Testing all features with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
 precision recall f1-score support  
  
 Edible 1.00 1.00 1.00 1272  
 Poisonous 1.00 1.00 1.00 1166  
  
 accuracy 1.00 2438  
 macro avg 1.00 1.00 1.00 2438  
weighted avg 1.00 1.00 1.00 2438  
  
Confusion-Matrix:  
[[1272 0]  
 [ 1 1165]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
 precision recall f1-score support  
  
 Edible 1.00 1.00 1.00 1272  
 Poisonous 1.00 1.00 1.00 1166  
  
 accuracy 1.00 2438  
 macro avg 1.00 1.00 1.00 2438  
weighted avg 1.00 1.00 1.00 2438  
  
Confusion-Matrix:  
[[1272 0]  
 [ 0 1166]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
 precision recall f1-score support  
  
 Edible 1.00 1.00 1.00 1272  
 Poisonous 1.00 1.00 1.00 1166  
  
 accuracy 1.00 2438  
 macro avg 1.00 1.00 1.00 2438  
weighted avg 1.00 1.00 1.00 2438  
  
Confusion-Matrix:  
[[1272 0]  
 [ 0 1166]]  
=================================================================================  
Testing all features with a 40% training and 60% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
 precision recall f1-score support  
  
 Edible 1.00 1.00 1.00 2542  
 Poisonous 1.00 1.00 1.00 2333  
  
 accuracy 1.00 4875  
 macro avg 1.00 1.00 1.00 4875  
weighted avg 1.00 1.00 1.00 4875  
  
Confusion-Matrix:  
[[2542 0]  
 [ 5 2328]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
 precision recall f1-score support  
  
 Edible 1.00 1.00 1.00 2542  
 Poisonous 1.00 1.00 1.00 2333  
  
 accuracy 1.00 4875  
 macro avg 1.00 1.00 1.00 4875  
weighted avg 1.00 1.00 1.00 4875  
  
Confusion-Matrix:  
[[2542 0]  
 [ 0 2333]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
 precision recall f1-score support  
  
 Edible 1.00 1.00 1.00 2542  
 Poisonous 1.00 1.00 1.00 2333  
  
 accuracy 1.00 4875  
 macro avg 1.00 1.00 1.00 4875  
weighted avg 1.00 1.00 1.00 4875  
  
Confusion-Matrix:  
[[2542 0]  
 [ 0 2333]]  
=================================================================================  
Testing all features with a 1% training and 99% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
 precision recall f1-score support  
  
 Edible 0.95 0.96 0.95 4162  
 Poisonous 0.96 0.94 0.95 3881  
  
 accuracy 0.95 8043  
 macro avg 0.95 0.95 0.95 8043  
weighted avg 0.95 0.95 0.95 8043  
  
Confusion-Matrix:  
[[3991 171]  
 [ 214 3667]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
 precision recall f1-score support  
  
 Edible 0.89 0.96 0.93 4162  
 Poisonous 0.95 0.87 0.91 3881  
  
 accuracy 0.92 8043  
 macro avg 0.92 0.92 0.92 8043  
weighted avg 0.92 0.92 0.92 8043  
  
Confusion-Matrix:  
[[4001 161]  
 [ 486 3395]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
 precision recall f1-score support  
  
 Edible 0.96 0.96 0.96 4162  
 Poisonous 0.96 0.96 0.96 3881  
  
 accuracy 0.96 8043  
 macro avg 0.96 0.96 0.96 8043  
weighted avg 0.96 0.96 0.96 8043  
  
Confusion-Matrix:  
[[3995 167]  
 [ 146 3735]]  
  
Preparing individual feature subsets  
=================================================================================  
Testing feature: cap-shape with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.5718  
Confusion-Matrix:  
[[1207 65]  
 [ 979 187]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.5681  
Confusion-Matrix:  
[[728 544]  
 [509 657]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.5722  
Confusion-Matrix:  
[[1207 65]  
 [ 978 188]]  
=================================================================================  
Testing feature: cap-surface with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.5726  
Confusion-Matrix:  
[[464 808]  
 [234 932]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.5496  
Confusion-Matrix:  
[[824 448]  
 [650 516]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.5726  
Confusion-Matrix:  
[[464 808]  
 [234 932]]  
=================================================================================  
Testing feature: cap-color with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.6001  
Confusion-Matrix:  
[[917 355]  
 [620 546]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.5619  
Confusion-Matrix:  
[[1046 226]  
 [ 842 324]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.6001  
Confusion-Matrix:  
[[917 355]  
 [620 546]]  
=================================================================================  
Testing feature: bruises with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.7482  
Confusion-Matrix:  
[[841 431]  
 [183 983]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.7482  
Confusion-Matrix:  
[[841 431]  
 [183 983]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.7482  
Confusion-Matrix:  
[[841 431]  
 [183 983]]  
=================================================================================  
Testing feature: odor with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.9865  
Confusion-Matrix:  
[[1272 0]  
 [ 33 1133]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.9865  
Confusion-Matrix:  
[[1272 0]  
 [ 33 1133]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.9865  
Confusion-Matrix:  
[[1272 0]  
 [ 33 1133]]  
=================================================================================  
Testing feature: gill-attachment with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.5217  
Confusion-Matrix:  
[[1272 0]  
 [1166 0]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.5217  
Confusion-Matrix:  
[[1272 0]  
 [1166 0]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.5217  
Confusion-Matrix:  
[[1272 0]  
 [1166 0]]  
=================================================================================  
Testing feature: gill-spacing with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.6079  
Confusion-Matrix:  
[[ 348 924]  
 [ 32 1134]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.5217  
Confusion-Matrix:  
[[1272 0]  
 [1166 0]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.6079  
Confusion-Matrix:  
[[ 348 924]  
 [ 32 1134]]  
=================================================================================  
Testing feature: gill-size with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.7506  
Confusion-Matrix:  
[[1186 86]  
 [ 522 644]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.7506  
Confusion-Matrix:  
[[1186 86]  
 [ 522 644]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.7506  
Confusion-Matrix:  
[[1186 86]  
 [ 522 644]]  
=================================================================================  
Testing feature: gill-color with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.8076  
Confusion-Matrix:  
[[1133 139]  
 [ 330 836]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.8002  
Confusion-Matrix:  
[[1110 162]  
 [ 325 841]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.8076  
Confusion-Matrix:  
[[1133 139]  
 [ 330 836]]  
=================================================================================  
Testing feature: stalk-shape with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.5455  
Confusion-Matrix:  
[[776 496]  
 [612 554]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.5455  
Confusion-Matrix:  
[[776 496]  
 [612 554]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.5455  
Confusion-Matrix:  
[[776 496]  
 [612 554]]  
=================================================================================  
Testing feature: stalk-root with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.6468  
Confusion-Matrix:  
[[1053 219]  
 [ 642 524]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.5168  
Confusion-Matrix:  
[[692 580]  
 [598 568]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.6468  
Confusion-Matrix:  
[[1053 219]  
 [ 642 524]]  
=================================================================================  
Testing feature: stalk-surface-above-ring with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.7797  
Confusion-Matrix:  
[[1229 43]  
 [ 494 672]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.7797  
Confusion-Matrix:  
[[1229 43]  
 [ 494 672]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.7797  
Confusion-Matrix:  
[[1229 43]  
 [ 494 672]]  
=================================================================================  
Testing feature: stalk-surface-below-ring with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.774  
Confusion-Matrix:  
[[1232 40]  
 [ 511 655]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.7551  
Confusion-Matrix:  
[[1168 104]  
 [ 493 673]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.774  
Confusion-Matrix:  
[[1232 40]  
 [ 511 655]]  
=================================================================================  
Testing feature: stalk-color-above-ring with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.726  
Confusion-Matrix:  
[[1102 170]  
 [ 498 668]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.4984  
Confusion-Matrix:  
[[441 831]  
 [392 774]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.726  
Confusion-Matrix:  
[[1102 170]  
 [ 498 668]]  
=================================================================================  
Testing feature: stalk-color-below-ring with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.7223  
Confusion-Matrix:  
[[1085 187]  
 [ 490 676]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.5915  
Confusion-Matrix:  
[[ 276 996]  
 [ 0 1166]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.7223  
Confusion-Matrix:  
[[1085 187]  
 [ 490 676]]  
=================================================================================  
Testing feature: veil-color with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.5226  
Confusion-Matrix:  
[[1272 0]  
 [1164 2]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.5062  
Confusion-Matrix:  
[[ 68 1204]  
 [ 0 1166]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.5226  
Confusion-Matrix:  
[[1272 0]  
 [1164 2]]  
=================================================================================  
Testing feature: ring-number with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.5304  
Confusion-Matrix:  
[[ 148 1124]  
 [ 21 1145]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.525  
Confusion-Matrix:  
[[1272 0]  
 [1158 8]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.5304  
Confusion-Matrix:  
[[ 148 1124]  
 [ 21 1145]]  
=================================================================================  
Testing feature: ring-type with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.7904  
Confusion-Matrix:  
[[996 276]  
 [235 931]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.7904  
Confusion-Matrix:  
[[996 276]  
 [235 931]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.7904  
Confusion-Matrix:  
[[996 276]  
 [235 931]]  
=================================================================================  
Testing feature: spore-print-color with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.8819  
Confusion-Matrix:  
[[1100 172]  
 [ 116 1050]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.8819  
Confusion-Matrix:  
[[1100 172]  
 [ 116 1050]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.8819  
Confusion-Matrix:  
[[1100 172]  
 [ 116 1050]]  
=================================================================================  
Testing feature: population with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.7281  
Confusion-Matrix:  
[[901 371]  
 [292 874]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.7281  
Confusion-Matrix:  
[[901 371]  
 [292 874]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.7281  
Confusion-Matrix:  
[[901 371]  
 [292 874]]  
=================================================================================  
Testing feature: habitat with a 70% training and 30% testing split...  
\*\*\*\*\*Logistic Regression\*\*\*\*\*  
Accuracy: 0.6932  
Confusion-Matrix:  
[[1130 142]  
 [ 606 560]]  
\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*  
Accuracy: 0.6542  
Confusion-Matrix:  
[[1208 64]  
 [ 779 387]]  
\*\*\*\*\*Support Vector Machine\*\*\*\*\*  
Accuracy: 0.6932  
Confusion-Matrix:  
[[1130 142]  
 [ 606 560]]