Deliverable 2

1. Problem statement : The dataset is a bunch of mushrooms with 23 known features. One of those features is whether it's safe to eat. The goal is to build a classifier that can predict that label.

2. Data Preprocessing:

   Dataset link is https://www.kaggle.com/datasets/uciml/mushroom-classification

   The dataset contains 8124 mushrooms, and 23 features. Every feature has string values, so I used sklearn's label encoder to get numerical values. In my data preprocessing, I dropped the "class" feature, which is our label. I then found the correlation matrix between the rest of the features and the label. By dropping all the features that have less than 0.2 correlation value with the label, I am left with 11 relevant features.

3. Machine learning model :
   a. in my first deliverable, I stated that I could use either a random forest classifier or an SVM model. Well since my data isn't too complicated, I tried both and the accuracy for the SVM model was slightly higher. The random forest classifier demanded some hyperparameter tuning to get an optimal accuracy (of around 96%-97%), and the SVM classifier reached an 98% accuracy with no further optimization. So I chose to use sklearn's SVM classifier.
   b. I split the data with respect to a 80/20 ratio in favor of training. Originally I had a 70/30 split but adding more training data got me a better accuracy (by around 0.2% on average). This makes sense because the dataset is comprised of hypothetical mushrooms, so every mushroom matters in a sense.
   c. I tested my model on the test dataset. I don't think its overfitting because it only loses around 0.5% accuracy between training and testing phases. It certainly isn't overfitting either.
   d. mostly no challenges, just a few hiccups messing around with the label encoder from sklearn but reading the documentation about it helped.

4. Performance:
   Here is the confusion matrix :

```
array([[851,   1],
       [ 38, 735]], dtype=int64)
```

   This is for the test set, which is comprised of 1625 mushrooms. We got only 1 false positive and 38 false negatives. I am very happy with the performance, because the confusion matrix shows that a large majority of the mistakes my classifier commits are false negatives, which means that the mushroom is actually safe to eat.

   The accuracy score is 97.9% on the training set and 97.6% on the testing set.

5. Next steps:

I looked at what other people did on Kaggle to solve the inaccuracy problem, and I found that people who used SVM/ random forest classifiers/Naïve Bayes calculations all got accuracies of about 97-98%. The people who reached an accuracy of 100% used logistical regression for the most part, or other models that I do not yet understand. I don't plan on changing my whole project to use logistical regression to get an extra 2% accuracy, this is good enough.