Jordan Dube
CMSC 435 Assignment 3
Table and Answers to Question 2

2.1. **List and briefly describe** the two algorithms that you selected. You should **name** the algorithms and briefly explain **why** you selected them and what **type of models** they produce.

1. Perceptron
    a. This Operator learns a linear classifier called Single Perceptron which finds separating hyperplane (if existent). Works well with numerical values. Repeats training rounds, altering weights and "learns" with each round. Terminates when specified number of rounds is met. I chose it because there are not many parameters and I have previous experience with Perceptrons from CMSC 409, an AI class. It produces a set of weights which define a hyperplane that most closely separates the points into classes.
2. Decision Tree
    a. This Operator generates a decision tree model, which can be used for classification and regression. It builds and prunes so as to minimize depth and maximize accuracy. When an object is then fed into it, it traverses each node, making decisions until it reaches a leaf, which is the predicted class of the object. I selected it because I saw how the parameters affect the output from the lecture in class (it is also the operator "provided" in the demo .rmp file). It produces a set of rules that define a Tree that patterns can traverse to find approximate classifications.

2.2 Using the table shown below, **report the accuracies** for the four algorithms and the three test types. The accuracy values must be reported with two digits after the decimal point, e.g., 91.05. You must include the accuracies of the models that use the default parameters and the best values of the key parameters. In total, you have 4*3*2 = 24 results to report. **Name the key parameters and list their best selected values** for each model and each test type; leave this part of the table empty if there are no parameters. Use the provided template of the table and upload the answer to this question as .jpg, .png or .pdf file.

Any values that are kept at default did not affect the accuracy of the model (enough to be reflected in performance vectors).

Reported information	Test type	k-NN	Naïve Bayes	Perceptron	Decision Tree
Accuracy with default parameters	Entire dataset	92.37%	87.36%	90.97%	91.39%
	50%	90.36%	87.83%	90.68%	90.93%
	Cross-validation	90.96% +/- 0.15%	87.15% +/- 0.55%	86.72% +/- 7.43%	90.80% +/- 0.49%
Accuracy with best parameters	Entire dataset	100%	87.36%	91.29%	95.62%
	50%	92.29%	87.83%	90.95%	91.77%
	Cross-validation	91.25% +/- 0.22%	87.15% +/- 0.55%	89.94% +/- 1.65%	90.89% +/- 0.57%
List names of parameters		K Measure types	n/a	Rounds Learning rate	Criterion Maximal Depth Confidence
List selected best values of parameters (in the same order as in the list of names)	Entire dataset	2 Mixed measures	n/a	20 0.05	Gini index 31 0.01
	50%	10 Mixed Measures	n/a	8 0.05	Accuracy 10 0.1
	Cross-validation	9 Mixed Measures	n/a	50 0.05	Accuracy 20 0.1

2.3 You should obtain 100% accuracy for at least one method and one type of test. Which type of test produced this accuracy value? Do you think 100% accuracy is a good result if we assume that data in this dataset, including the yes/no Class feature, is noisy?

1. K-NN modified to have k-value of 2 and Measure Type of Mixed Measures tested on the entire dataset had the 100% accuracy.

2. No, this is not a good result if the dataset is noisy because 100% accuracy means that it is correctly classifying the noise, which is not a robust model, i.e. OVERFITTING.

2.4 **Provide** "confusion matrix" for the most accurate result computed based on the **three-fold cross-validation experiments** (selected among the 8 corresponding experiments). This is the matrix in the PerformanceVector view. Use this matrix to **explain** whether this predictor would be better suited to identify proteins that interact with nucleic acids (Class = *Yes*), proteins that do not interact with nucleic acids (Class = *No*), or both types of proteins. Upload the answer to this question as .jpg, .png or .pdf file.

1.

	True No	True Yes	Class Precision
Pred. No	7771	673	92.03%
Pred. Yes	88	263	74.93%
Class Recall	98.88%	28.10%	

Accuracy of k-NN model for 3-fold cross validation with k of 2 and Measure Type of Mixed Measures

2. This predictor would be better suited to predicting proteins that do not interact with nucleic acids (Class = *No*) because it correctly identified "No" 98.88% of the time, while only correctly predicting "Yes" 28.1% of the time, so it has a **much** higher chance of being correct for "No" classifications.

2.5 Using the results from the **three-fold cross-validation tests, briefly discuss** whether trying multiple algorithms and adjusting their parameters helped you in developing a more accurate predictive model compared to the results that rely on the simple Naïve Bayes method. Argue **whether or not** this amount of improvement over the Naïve Bayes is large – try your best to **justify your argument**. Consider the fact that it is trivial/easy to produce predictions that are 89.36% accurate if we simply predict all proteins with label "No".

1. Yes, testing the parameters allowed me to consistently produce models that outperformed the Naïve Bayes 3-fold cross validation model (91.25%, 89.94%, 90.89% vs the 87.15% of the naïve model).

2. This improvement is large, since there is an average of 3.543 percentage point increase in accuracy across all 3-fold cross validation models compared to the naïve bayes

model. Since the naïve Bayes model performs worse than the trivial “always say no” approach, the gain exhibited by the other 3-fold cross validation models becomes diminished, yet still an improvement. Since the data can be noisy, however, this increase in accuracy even compared to the trivial model is significant (Since very high accuracy can be undesired since it can be attributed to problems with overfitting/incorrect modeling).