Develop predictive models that can determine given a particular compound whether it is active (1) or not (0).

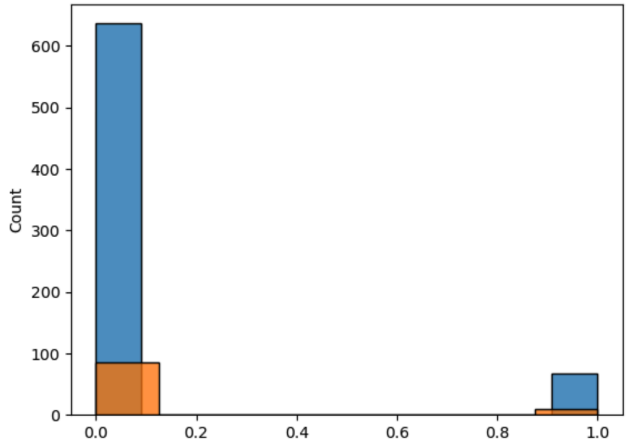
Drugs are typically small organic molecules that achieve their desired activity by binding to a target site on a receptor. The first step in the discovery of a new drug is usually to identify and isolate the receptor to which it should bind, followed by testing many small molecules for their ability to bind to the target site. This leaves researchers with the task of determining what separates the active (binding) compounds from the inactive (non-binding) ones. Such a determination can then be used in the design of new compounds that not only bind, but also have all the other properties required for a drug (solubility, oral absorption, lack of side effects, appropriate duration of action, toxicity, etc.).

The given dataset has the class labels as the first number in the rows in binary format 0 or 1. The features are represented in binary format as well, the rest of the numbers in a row represent the indices of the 1s in each record, and the indices represent the active compounds. I divided the class labels and indices into train Y and train X. The indices need to be mapped as 1 in each record in the corresponding index, in a compressed sparse row matrix for memory and speed optimization purposes. I created a class called “MatrixParser” to have two methods - The first method is called “calculateMatrixColumn”, this method will find out the max index in the records in the dataset and return the max index; the second method “parseIndexToMatrix” will take the returned value from the first method, and create an array of the max index length with only 0s for each data record and assign the indices that exist in each data record to 1 in the corresponding indices in the array. The parsed array for each data record will be appended to a “res” array and the “res” array will be transformed into a compressed sparse row matrix. Checking the sparse matrix for “res” of train X, we can see only the active compound, 1, in rows and columns. The same operation is applied to the test set.

Looking at the train X matrix and the characteristics of the dataset, the values are already normalized since they are already in binary format and they are not on different scales. However, some values can be dependent on others and redundancy can be eliminated to reduce the noise and dimensionality of the dataset for better accuracy. For my experiments with three types of classifiers, I applied “SelectKBest” with the chi2 score function or mutual information classification score function for feature selection [1]-[3]. With the chi2 test, the irrelevant features can be eliminated and I can choose k numbers of the top most relevant features, this is valid for this dataset because the features are non-negative. With mutual information classification, I can select the k numbers of the top most relevant features to the class label, this is valid for this dataset because the class label is discrete. For some experiments, I applied either chi2 SelectKBest or/and mutual information classification SelectKBest with different experimental k values to experiment with model accuracies based on F1 scores from cross-validation and from the validation set. The interpretations for experiments can be found in the last column of the tables.

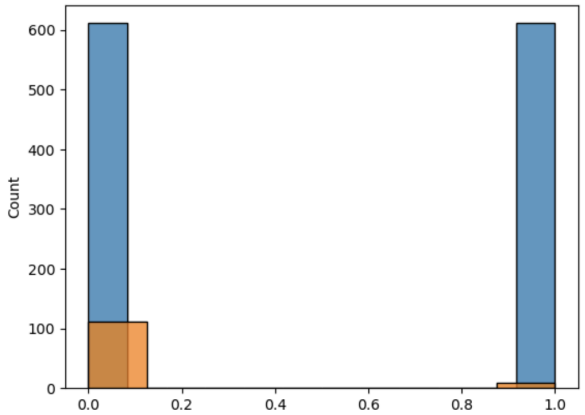
I divided train X to train X, Y, and validation X, Y. The reason is that I want to have a validation set to validate the model performance. Before dividing, train X and Y are shuffled to keep the randomness of the train set and validation set so that our model is not constantly learning the same data which could lead to overfitting. By comparing the cross-validation F1 score and validation set F1 score, I can actually see how the model performs on unseen data.

The dataset is very imbalanced as stated in the homework description and the graph drawn below. The blue represents train Y and the orange represents validation Y. From this graph, the difference between the total counts of labels 1 and 0 is really obviously significant. This is why we need to apply resampling.



I chose to apply random oversampling [4] to over-sample the underrepresented label in the train set. The reason is that the dataset is already small, it is helpful for model training and accuracy with more data, so we over-sample instead of under-sample. The new train, validation, and test ratio is roughly 72, 7, 21. See the screenshot below. Each experiment has slightly different values of train and validation class label amounts, details can be found in the screenshots column in the tables.

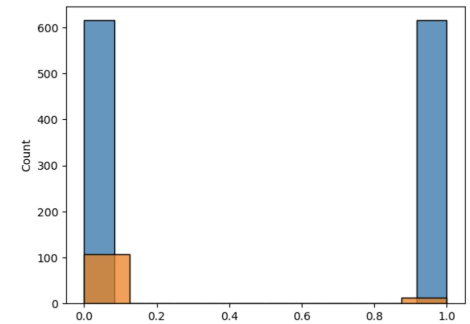
```
train/val/test ratio:  0.7222222222222222 0.07092198581560284 0.20685579196217493
Train label is 1:  611 Train label is 0:  611
Validation label is 1:  9 validation label is 0:  111
```

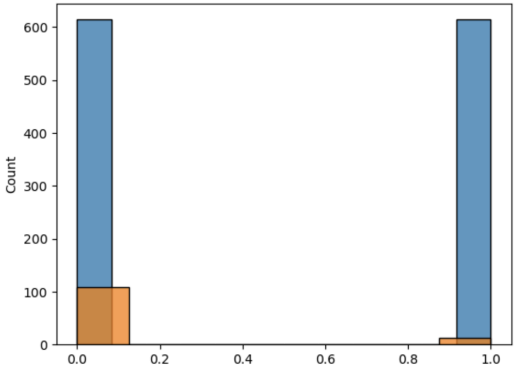


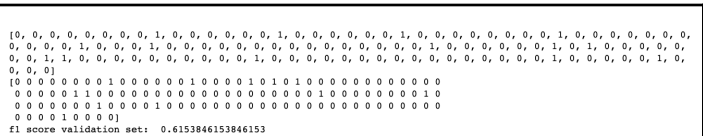
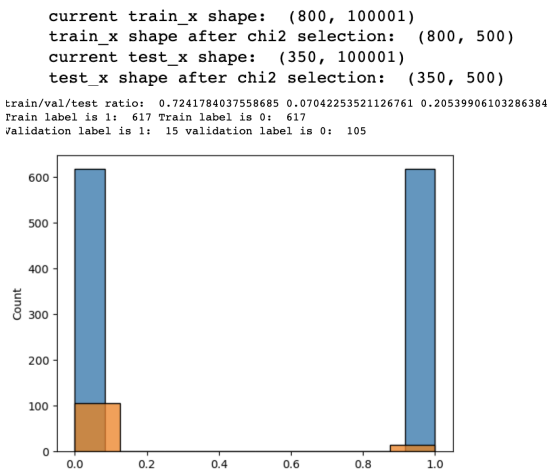
For cross-validation, I chose to use stratified k-fold cross-validation [5]. The reason is each fold can have equal instances of target class to prevent bias. For the Naive Bayes classifier, I choose to use Bernoulli Naive Bayes [6] because the dataset has discrete binary features and labels, for the same reason, Perceptron [7], [8] is chosen for the Neural Network classifier.

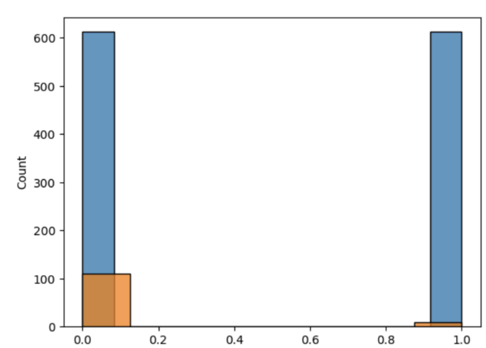
The random_state parameter is different for each experiment for reproducibility. The parameter is applied in the shuffle of train X and Y, train validation split, random oversampling, stratified k fold, decision tree classifier, and perceptron. The highlighted column is the final submission in Miner.

Stratified Cross Validation Experiments - Decision Tree					
random_state	params applied	CV n_splits	f1_scores	screenshots	interpretation

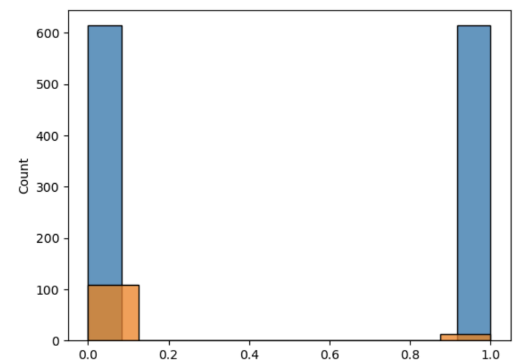
1	not specified	10	average f1 score: 0.98 validation set f1 score: 0.50	train/val/test ratio: 0.7235294117647059 0.07058823529411765 0.20588235294117646 Train label is 1: 615 Train label is 0: 615 Validation label is 1: 13 validation label is 0: 107  tree depth: 10 train label ratio: 0.5004516711833785 0.4995483288166215 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.976 confusion matrix: [[59 3] [0 61]] tree depth: 9 train label ratio: 0.5004516711833785 0.4995483288166215 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9838709677419354 confusion matrix: [[60 2] [0 61]] tree depth: 10 train label ratio: 0.5004516711833785 0.4995483288166215 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9682539682539683 confusion matrix: [[58 4] [0 61]] tree depth: 9 train label ratio: 0.5004516711833785 0.4995483288166215 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9606299212598425 confusion matrix: [[57 5] [0 61]] tree depth: 11 train label ratio: 0.5004516711833785 0.4995483288166215 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9457364341085273 confusion matrix: [[55 7] [0 61]] tree depth: 9 train label ratio: 0.4995483288166215 0.5004516711833785 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.9919999999999999 confusion matrix: [[60 1] [0 62]] tree depth: 9 train label ratio: 0.4995483288166215 0.5004516711833785 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.9465648854961832 confusion matrix: [[54 7] [0 62]] tree depth: 10 train label ratio: 0.4995483288166215 0.5004516711833785 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.9612403100775194 confusion matrix: [[56 5] [0 62]] tree depth: 11 train label ratio: 0.4995483288166215 0.5004516711833785 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.9763779527559054 confusion matrix: [[58 3] [0 62]] tree depth: 8 train label ratio: 0.4995483288166215 0.5004516711833785 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.9763779527559054 confusion matrix: [[58 3] [0 62]] average f1 score: 0.9763779527559054 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0] [0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0] f1 score validation set: 0.4999999999999999	No manual max_depth param .etc., just let the tree grow to pure regions. The average F1 score is really high, trees grow to a depth from 8 to 11. The validation set score is very different from the cross-validation F1 score. The model could be overfitting the training set. diff = 0.48

4	max_depth=8	10	average f1 score: 0.97 validation set f1 score: 0.62	train/val/test ratio: 0.7232037691401649 0.0706713780918728 0.2061248527679623 Train label is 1: 614 Train label is 0: 614 Validation label is 1: 12 validation label is 0: 108  tree depth: 8 train label ratio: 0.5004524886877828 0.4995475113122172 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9838709677419354 confusion matrix: [[60 2] [0 61]] tree depth: 8 train label ratio: 0.5004524886877828 0.4995475113122172 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9354838709677418 confusion matrix: [[57 5] [3 58]] tree depth: 8 train label ratio: 0.5004524886877828 0.4995475113122172 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9918699186991871 confusion matrix: [[61 1] [0 61]] tree depth: 8 train label ratio: 0.5004524886877828 0.4995475113122172 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9756097560975611 confusion matrix: [[60 2] [1 60]] tree depth: 8 train label ratio: 0.4995475113122172 0.5004524886877828 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.976 confusion matrix: [[59 2] [1 61]] tree depth: 8 train label ratio: 0.4995475113122172 0.5004524886877828 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.96875 confusion matrix: [[57 4] [0 62]] tree depth: 8 train label ratio: 0.4995475113122172 0.5004524886877828 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.9672131147540983 confusion matrix: [[60 1] [3 59]] tree depth: 8 train label ratio: 0.4995475113122172 0.5004524886877828 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.9838709677419355 confusion matrix: [[60 1] [1 61]] tree depth: 8 train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9752066115702478 confusion matrix: [[60 1] [2 59]] tree depth: 8 train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9677419354838709 confusion matrix: [[58 3] [1 60]] average f1 score: 0.9677419354838709	The max_depth is limited to prevent the model from learning training data too precisely. The validation score is closer to the average cross-validation score. Applying more parameters to see if we can improve accuracy. diff = 0.35

					
6	SelectKBest(chi2, k=500) max_depth=8	10	average f1 score: 0.90 validation set f1 score: 0.79		After reexaminaiting the shape of train_x, there is 100001 features for each record. I decided to apply chi2 select k best to eliminate irrelevant features in order to reduce noise and improve accuracy. the f1 scores between cross-validation and validation set are even closer. Applying the chi2 feature selection helped with overall accuracy. diff = 0.11

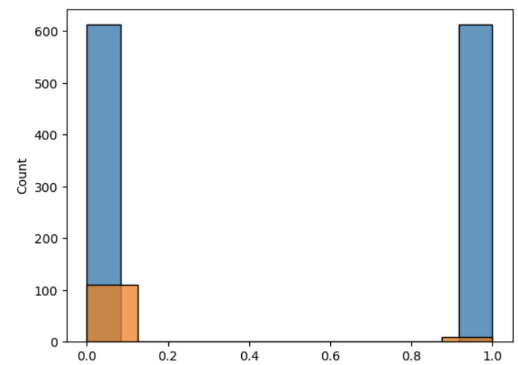
				tree depth: 8 train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.923076923076923 confusion matrix: [[61 1] [8 54]] tree depth: 8 train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9682539682539683 confusion matrix: [[59 3] [1 61]] tree depth: 8 train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.8925619834710744 confusion matrix: [[57 5] [8 54]] tree depth: 8 train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9322033898305085 confusion matrix: [[61 1] [7 55]] tree depth: 8 train label ratio: 0.5004500450045004 0.49954995499549953 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9256198347107439 confusion matrix: [[58 4] [5 56]] tree depth: 8 train label ratio: 0.5004500450045004 0.49954995499549953 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9666666666666667 confusion matrix: [[61 1] [3 58]] tree depth: 8 train label ratio: 0.5004500450045004 0.49954995499549953 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9752066115702478 confusion matrix: [[61 1] [2 59]] tree depth: 8 train label ratio: 0.49954995499549953 0.5004500450045004 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.9279999999999999 confusion matrix: [[56 5] [4 58]] tree depth: 8 train label ratio: 0.49954995499549953 0.5004500450045004 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.9682539682539683 confusion matrix: [[58 3] [1 61]] tree depth: 8 train label ratio: 0.49954995499549953 0.5004500450045004 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.8983050847457628 confusion matrix: [[58 3] [9 53]] average f1 score: 0.8983050847457628 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0] [0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0] f1 score validation set: 0.7878787878787877	
7	SelectKBest(chi2, k=500) max_depth=8, min_samples_split=30, min_samples_leaf=10	10	average f1 score: 0.94 validation set f1 score: 0.73	train/val/test ratio: 0.7225501770956316 0.07083825265643448 0.2066115702479339 Train label is 1: 612 Train label is 0: 612 Validation label is 1: 10 validation label is 0: 110 	Apply min_samples_split as 30 so that if in a node that samples is less than 30, no split. So that the decision boundary does not get too specific. Apply min_samples_leaf as 10 so that one node has to have at least 10 samples to prevent overfitting. With the limitation of min_samples_split and min_samples_leaf to prevent overfitting. It looks like applying

				tree depth: 8 train label ratio: 0.5004541326067211 0.49954586739327883 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.944 confusion matrix: [[57 5] [2 59]] tree depth: 8 train label ratio: 0.5004541326067211 0.49954586739327883 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9401709401709402 confusion matrix: [[61 1] [6 55]] tree depth: 8 train label ratio: 0.49954586739327883 0.5004541326067211 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.930232558139535 confusion matrix: [[54 7] [2 60]] tree depth: 8 train label ratio: 0.49954586739327883 0.5004541326067211 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.9333333333333333 confusion matrix: [[59 2] [6 56]] tree depth: 8 train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9016393442622952 confusion matrix: [[55 6] [6 55]] tree depth: 8 train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9217391304347826 confusion matrix: [[60 1] [8 53]] tree depth: 8 train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.95 confusion matrix: [[59 2] [4 57]] tree depth: 8 train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9365079365079365 confusion matrix: [[55 6] [2 59]] tree depth: 8 train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.8813559322033898 confusion matrix: [[56 5] [9 52]] tree depth: 8 train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9421487603305784 confusion matrix: [[58 3] [4 57]] average f1 score: 0.9421487603305784 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] f1 score validation set: 0.7272727272727272	min_samples_split and min_samples_leaf did not help with the accuracy difference. It could be because the minimum sample numbers are too big for such a small dataset. I will lower the minimum split and leaf a bit and see what happens. I will increase the max_depth limit in case more depth is needed. diff = 0.21

8	SelectKBest(chi2, k=500) max_depth=15, min_samples_split=10, min_samples_leaf=4	10	average f1 score: 0.94 validation set f1 score: 0.69	train/val/test ratio: 0.7232037691401649 0.0706713780918728 0.2061248527679623 Train label is 1: 614 Train label is 0: 614 Validation label is 1: 12 validation label is 0: 108  tree depth: 15 train label ratio: 0.5004524886877828 0.4995475113122172 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9677419354838709 confusion matrix: [[59 3] [1 60]] tree depth: 15 train label ratio: 0.5004524886877828 0.4995475113122172 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9586776859504132 confusion matrix: [[60 2] [3 58]] tree depth: 15 train label ratio: 0.5004524886877828 0.4995475113122172 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.959349593495935 confusion matrix: [[59 3] [2 59]] tree depth: 15 train label ratio: 0.5004524886877828 0.4995475113122172 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9752066115702478 confusion matrix: [[61 1] [2 59]] tree depth: 15 train label ratio: 0.4995475113122172 0.5004524886877828 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.9672131147540983 confusion matrix: [[60 1] [3 59]] tree depth: 15 train label ratio: 0.4995475113122172 0.5004524886877828 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.975609756097561 confusion matrix: [[60 1] [2 60]] tree depth: 15 train label ratio: 0.4995475113122172 0.5004524886877828 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.9763779527559054 confusion matrix: [[58 3] [0 62]] tree depth: 15 train label ratio: 0.4995475113122172 0.5004524886877828 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.9682539682539683 confusion matrix: [[58 3] [1 61]] tree depth: 15 train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9677419354838709 confusion matrix: [[58 3] [1 60]] tree depth: 15 train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9411764705882353 confusion matrix: [[59 2] [5 56]] average f1 score: 0.9411764705882353 [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1] [0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0] f1 score validation set: 0.6896551724137931	The accuracy difference increased, lowering number of minimum sample split and leaf did not help with the accuracy. The next experiment will increase the minimum parameters. diff = 0.25

<table>
<tr>
<td>11</td>
<td>SelectKBest(chi2, k=500)

min_samples_split=22, min_samples_leaf=13</td>
<td>10</td>
<td>average f1 score: 0.95

validation set f1 score: 0.67</td>
<td>train/val/test ratio: 0.7225501770956316 0.07083825265643448 0.2066115702479339
Train label is 1: 612 Train label is 0: 612
Validation label is 1: 10 validation label is 0: 110
 

tree depth: 8
train label ratio: 0.5004541326067211 0.49954586739327883
test label ratio: 0.4959349593495935 0.5040650406504065
f1 score: 0.8965517241379309
confusion matrix:
 [[59 3]
 [9 52]]
tree depth: 9
train label ratio: 0.5004541326067211 0.49954586739327883
test label ratio: 0.4959349593495935 0.5040650406504065
f1 score: 0.9059829059829059
confusion matrix:
 [[59 3]
 [8 53]]
tree depth: 8
train label ratio: 0.49954586739327883 0.5004541326067211
test label ratio: 0.5040650406504065 0.4959349593495935
f1 score: 0.9075630252100839
confusion matrix:
 [[58 3]
 [8 54]]
tree depth: 8
train label ratio: 0.49954586739327883 0.5004541326067211
test label ratio: 0.5040650406504065 0.4959349593495935
f1 score: 0.9516129032258065
confusion matrix:
 [[58 3]
 [3 59]]
tree depth: 9
train label ratio: 0.5 0.5
test label ratio: 0.5 0.5
f1 score: 0.9833333333333333
confusion matrix:
 [[61 0]
 [2 59]]
tree depth: 9
train label ratio: 0.5 0.5
test label ratio: 0.5 0.5
f1 score: 0.9043478260869565
confusion matrix:
 [[59 2]
 [9 52]]
tree depth: 9
train label ratio: 0.5 0.5
test label ratio: 0.5 0.5
f1 score: 0.944
confusion matrix:
 [[56 5]
 [2 59]]
tree depth: 9
train label ratio: 0.5 0.5
test label ratio: 0.5 0.5
f1 score: 0.9193548387096775
confusion matrix:
 [[55 6]
 [4 57]]
tree depth: 8
train label ratio: 0.5 0.5
test label ratio: 0.5 0.5
f1 score: 0.9421487603305784
confusion matrix:
 [[58 3]
 [4 57]]
tree depth: 9
train label ratio: 0.5 0.5
test label ratio: 0.5 0.5
f1 score: 0.95
confusion matrix:
 [[59 2]
 [4 57]]
average f1 score: 0.95
[1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1]
[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0
 0 0 0 0 1 0 0 0 0]
f1 score validation set: 0.6666666666666666</td>
<td>The accuracy difference increased comparing to random_state 8.

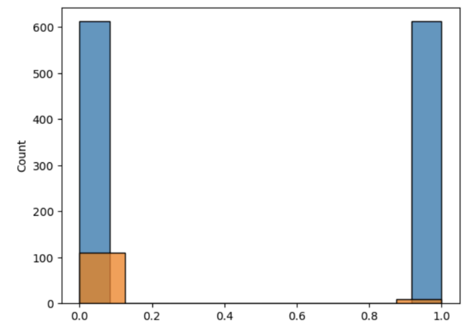
Out of all experiments, random_state 6 works the best so far.

Going to adjust max_depth to see if any value can produce a better result than random_state 6.

diff = 0.28</td>
</tr>
</table>

<table>
<tr>
<td>13</td>
<td>SelectKBest(chi2, k=500)

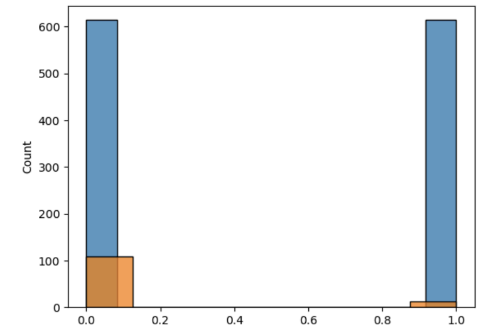
max_depth=7</td>
<td>10</td>
<td>average f1 score: 0.90

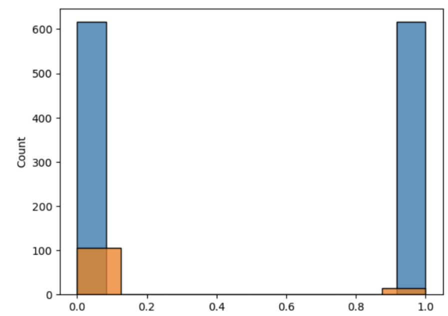
validation set f1 score: 0.67</td>
<td>train/val/test ratio: 0.7225501770956316 0.07083825265643448 0.2066115702479339
Train label is 1: 612 Train label is 0: 612
Validation label is 1: 10 validation label is 0: 110
 

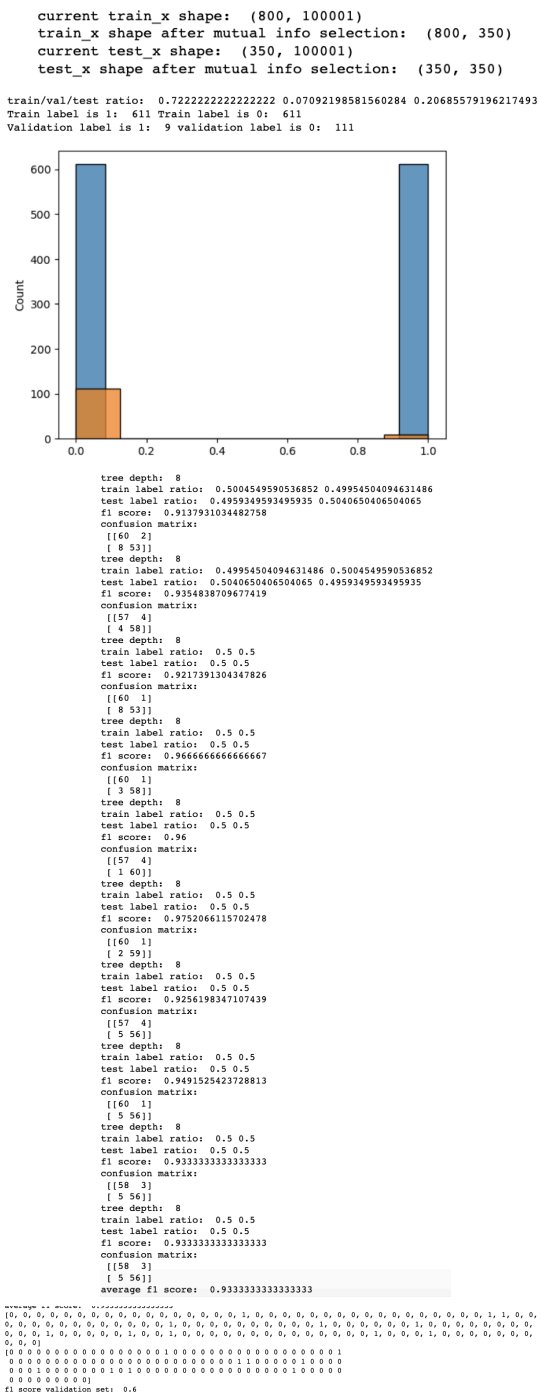
tree depth: 7
train label ratio: 0.5004541326067211 0.49954586739327883
test label ratio: 0.4959349593495935 0.5040650406504065
f1 score: 0.9401709401709402
confusion matrix:
 [[61 1]
 [6 55]]
tree depth: 7
train label ratio: 0.5004541326067211 0.49954586739327883
test label ratio: 0.4959349593495935 0.5040650406504065
f1 score: 0.9203539823008848
confusion matrix:
 [[62 0]
 [9 52]]
tree depth: 7
train label ratio: 0.49954586739327883 0.5004541326067211
test label ratio: 0.5040650406504065 0.4959349593495935
f1 score: 0.9152542372881356
confusion matrix:
 [[59 2]
 [8 54]]
tree depth: 7
train label ratio: 0.49954586739327883 0.5004541326067211
test label ratio: 0.5040650406504065 0.4959349593495935
f1 score: 0.8888888888888888
confusion matrix:
 [[58 3]
 [10 52]]
tree depth: 7
train label ratio: 0.5 0.5
test label ratio: 0.5 0.5
f1 score: 0.9661016949152543
confusion matrix:
 [[61 0]
 [4 57]]
tree depth: 7
train label ratio: 0.5 0.5
test label ratio: 0.5 0.5
f1 score: 0.8771929824561403
confusion matrix:
 [[58 3]
 [11 50]]
tree depth: 7
train label ratio: 0.5 0.5
test label ratio: 0.5 0.5
f1 score: 0.8869565217391304
confusion matrix:
 [[58 3]
 [10 51]]
tree depth: 7
train label ratio: 0.5 0.5
test label ratio: 0.5 0.5
f1 score: 0.9401709401709402
confusion matrix:
 [[60 1]
 [6 55]]
tree depth: 7
train label ratio: 0.5 0.5
test label ratio: 0.5 0.5
f1 score: 0.8947368421052632
confusion matrix:
 [[59 2]
 [10 51]]
tree depth: 7
train label ratio: 0.5 0.5
test label ratio: 0.5 0.5
f1 score: 0.9043478260869565
confusion matrix:
 [[59 2]
 [9 52]]
average f1 score: 0.9043478260869565
[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0]
[0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
 0 0 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 1]
f1 score validation set: 0.6666666666666666</td>
<td>Comparing to random_state 6, the accuracy difference increased between average cross-validation F1 score and validation set F1 score.

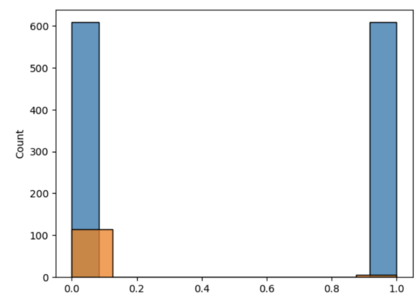
Going to try using mutual information classification with select k best next.

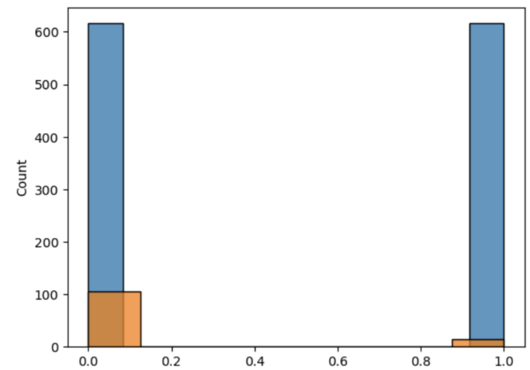
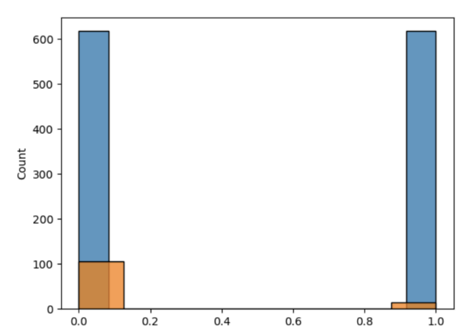
diff = 0.23</td>
</tr>
</table>

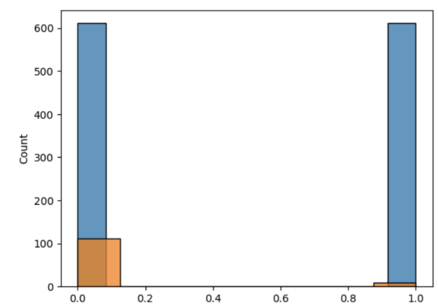
35	SelectKBest(score_func=mutual_info, k=500)	10	average f1 score: 0.97 validation set f1 score: 0.64	current train_x shape: (800, 100001) train_x shape after mutual info selection: (800, 500) current test_x shape: (350, 100001) test_x shape after mutual info selection: (350, 500) train/val/test ratio: 0.7232037691401649 0.0706713780918728 0.2061248527679623 Train label is 1: 614 Train label is 0: 614 Validation label is 1: 12 validation label is 0: 108  tree depth: 65 train label ratio: 0.5004524886877828 0.4995475113122172 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9401709401709402 confusion matrix: [[61 1] [6 55]] tree depth: 62 train label ratio: 0.5004524886877828 0.4995475113122172 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.976 confusion matrix: [[59 3] [0 61]] tree depth: 63 train label ratio: 0.5004524886877828 0.4995475113122172 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9677419354838709 confusion matrix: [[59 3] [1 60]] tree depth: 60 train label ratio: 0.5004524886877828 0.4995475113122172 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9666666666666667 confusion matrix: [[61 1] [3 58]] tree depth: 64 train label ratio: 0.4995475113122172 0.5004524886877828 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.976 confusion matrix: [[59 2] [1 61]] tree depth: 60 train label ratio: 0.4995475113122172 0.5004524886877828 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.9836065573770492 confusion matrix: [[61 0] [2 60]] tree depth: 63 train label ratio: 0.4995475113122172 0.5004524886877828 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.959349593495935 confusion matrix: [[59 2] [3 59]] tree depth: 65 train label ratio: 0.4995475113122172 0.5004524886877828 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.967741935483871 confusion matrix: [[59 2] [2 60]] tree depth: 63 train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9672131147540983 confusion matrix: [[59 2] [2 59]] tree depth: 66 train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9672131147540983 confusion matrix: [[59 2] [2 59]] average f1 score: 0.9672131147540983 [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0] [1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0] f1 score validation set: 0.64	Adding mutual info select k best feature selection did not improve the accuracies overall. The tree depth grows to 66, limiting the max_depth can help with the accuracy potentially. The next experiment will apply max_dpeth limitation with mutual information classification. diff = 0.33

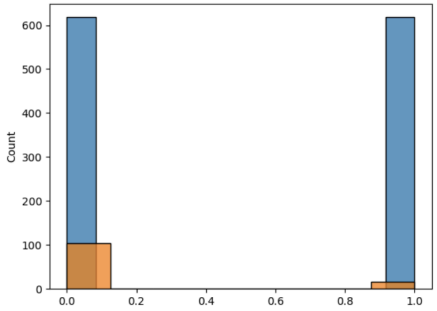
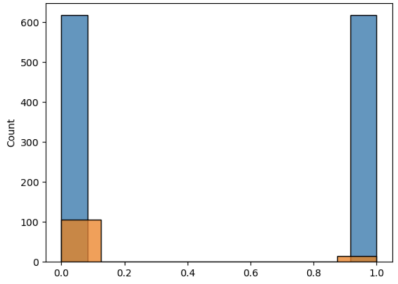
40	SelectKBest(score_func=mutual_info, k=500) max_depth = 8	10	average f1 score: 0.92 validation set f1 score: 0.69	train/val/test ratio: 0.7238542890716804 0.07050528789659224 0.2056404230317274 Train label is 1: 616 Train label is 0: 616 Validation label is 1: 14 validation label is 0: 106  tree depth: 8 train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9243697478991597 confusion matrix: [[60 2] [7 55]] tree depth: 8 train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9333333333333333 confusion matrix: [[60 2] [6 56]] tree depth: 8 train label ratio: 0.5004508566275925 0.4995491433724076 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9572649572649572 confusion matrix: [[62 0] [5 56]] tree depth: 8 train label ratio: 0.5004508566275925 0.4995491433724076 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9322033898305084 confusion matrix: [[60 2] [6 55]] tree depth: 8 train label ratio: 0.5004508566275925 0.4995491433724076 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.9833333333333333 confusion matrix: [[62 0] [2 59]] tree depth: 8 train label ratio: 0.5004508566275925 0.4995491433724076 test label ratio: 0.4959349593495935 0.5040650406504065 f1 score: 0.95 confusion matrix: [[60 2] [4 57]] tree depth: 8 train label ratio: 0.4995491433724076 0.5004508566275925 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.9365079365079365 confusion matrix: [[56 5] [3 59]] tree depth: 8 train label ratio: 0.4995491433724076 0.5004508566275925 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.9411764705882352 confusion matrix: [[60 1] [6 56]] tree depth: 8 train label ratio: 0.4995491433724076 0.5004508566275925 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.9137931034482759 confusion matrix: [[60 1] [9 53]] tree depth: 8 train label ratio: 0.4995491433724076 0.5004508566275925 test label ratio: 0.5040650406504065 0.4959349593495935 f1 score: 0.9243697478991597 confusion matrix: [[59 2] [7 55]] average f1 score: 0.9243697478991597 [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0] [0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0] f1 score validation set: 0.6923076923076924	The accuracy difference is very similarly bad comparing to other experiments above. But compared to random_state 35, with the limitation of max_depth, the accuracy difference decreased. diff = 0.23

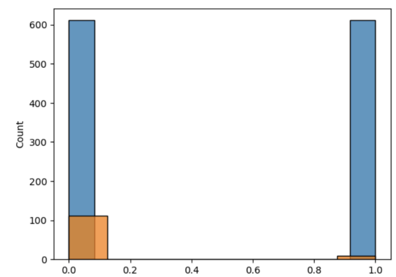
41	SelectKBest(score_func=mutual_info, k=350) max_depth = 8	10	average f1 score: 0.93 validation set f1 score: 0.60		Lowering the k in select k best increased the accuracy difference. It looks like using the chi2 score function to select the top k features generally have better accuracy result than using mutual information classification. diff = 0.33

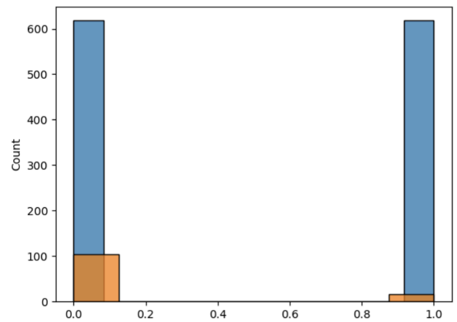
Stratified Cross Validation Experiments - Naive Bayes					
random_ state	params applied	CV n_splits	f1_scores	screenshots	interpretation
3	force_alpha =True	5	average f1 score: 0.98 validation set f1 score: 0.0	train/val/test ratio: 0.7212336892052195 0.0711743772241993 0.20759193357058126 Train label is 1: 608 Train label is 0: 608 Validation label is 1: 6 validation label is 0: 114  train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9834710743801653 confusion matrix: [[121 1] [3 119]] train label ratio: 0.5005138746145941 0.499486125385406 test label ratio: 0.49794238683127573 0.5020576131687243 f1 score: 0.9958847736625513 confusion matrix: [[121 1] [0 121]] train label ratio: 0.5005138746145941 0.499486125385406 test label ratio: 0.49794238683127573 0.5020576131687243 f1 score: 0.9745762711864406 confusion matrix: [[122 0] [6 115]] train label ratio: 0.499486125385406 0.5005138746145941 test label ratio: 0.5020576131687243 0.49794238683127573 f1 score: 0.9876543209876544 confusion matrix: [[120 1] [2 120]] train label ratio: 0.499486125385406 0.5005138746145941 test label ratio: 0.5020576131687243 0.49794238683127573 f1 score: 0.979253112033195 confusion matrix: [[120 1] [4 118]] average f1 score: 0.979253112033195 [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] [0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0] f1 score validation set: 0.0	Using smaller cross-validation folds because we do not have a big dataset. The difference between accuracies being really big could be due to the model is overfitting the training set. Naive Bayes also assumes every feature is independent. Next experiment we apply mutual info classification SelectKBest to eliminate dependent features diff = 0.98
15	force_alpha =True mutual_info_kbest = SelectKBest(score_func=mutual_info, k=500)	5	average f1 score: 0.90 validation set f1 score: 0.77	current train_x shape: (800, 100001) train_x shape after mutual info selection: (800, 500) current test_x shape: (350, 100001) test_x shape after mutual info selection: (350, 500)	After eliminating the dependent features and selecting the k best independent features, the accuracy difference between average cross validation F1 score and validation set f1 score is much closer. For the next experiment,

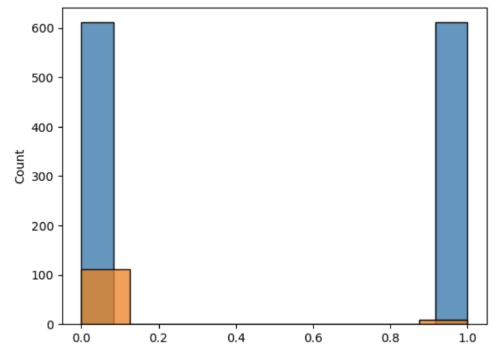
				train/val/test ratio: 0.7238542890716804 0.07050528789659224 0.2056404230317274 Train label is 1: 616 Train label is 0: 616 Validation label is 1: 14 validation label is 0: 106  train label ratio: 0.500507614213198 0.49949238578680205 test label ratio: 0.4979757085020243 0.5020242914979757 f1 score: 0.8493150684931506 confusion matrix: [[121 3] [30 93]] train label ratio: 0.49949238578680205 0.500507614213198 test label ratio: 0.5020242914979757 0.4979757085020243 f1 score: 0.8869565217391304 confusion matrix: [[119 4] [22 102]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9249999999999999 confusion matrix: [[117 6] [12 111]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.8898678414096917 confusion matrix: [[120 3] [22 101]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9026548672566372 confusion matrix: [[122 1] [21 102]] average f1 score: 0.9026548672566372 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0] f1 score validation set: 0.7692307692307692	let's apply both mutual info classfi select k best and chi2 select k best to eliminate dependent and irrelevant features. diff = 0.13
18	force_alpha =True SelectKBest(score_func=mutual_info, k=600) SelectKBest(chi2, k=300)	5	average f1 score: 0.88 validation set f1 score: 0.70	current train_x shape: (800, 100001) train_x shape after mutual info selection: (800, 600) current test_x shape: (350, 100001) test_x shape after mutual info selection: (350, 600) current train_x shape: (800, 600) train_x shape after chi2 selection: (800, 300) current test_x shape: (350, 600) test_x shape after chi2 selection: (350, 300) train/val/test ratio: 0.7241784037558685 0.07042253521126761 0.20539906103286384 Train label is 1: 617 Train label is 0: 617 Validation label is 1: 15 validation label is 0: 105 	The accuracy difference is bigger compared to just using mutual information classification. The next experiment will just use select k best for mutual information classification but make k bigger than what is in random_state 15. diff = 0.18

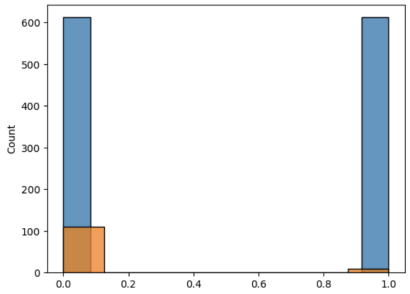
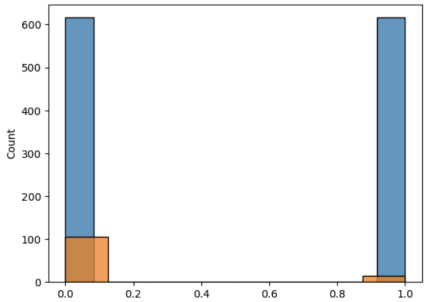
				train label ratio: 0.5005065856129686 0.49949341438703143 test label ratio: 0.4979757085020243 0.5020242914979757 f1 score: 0.8908296943231442 confusion matrix: [[120 4] [21 102]] train label ratio: 0.5005065856129686 0.49949341438703143 test label ratio: 0.4979757085020243 0.5020242914979757 f1 score: 0.8440366972477064 confusion matrix: [[121 3] [31 92]] train label ratio: 0.49949341438703143 0.5005065856129686 test label ratio: 0.5020242914979757 0.4979757085020243 f1 score: 0.9137931034482759 confusion matrix: [[121 2] [18 106]] train label ratio: 0.49949341438703143 0.5005065856129686 test label ratio: 0.5020242914979757 0.4979757085020243 f1 score: 0.9012875536480687 confusion matrix: [[119 4] [19 105]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.8771929824561403 confusion matrix: [[118 5] [23 100]] average f1 score: 0.8771929824561403 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] [0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] f1 score validation set: 0.6956521739130436	
30	force_alpha =True SelectKBest(score_func=mutual_info, k=600)	5	average f1 score: 0.84 validation set f1 score: 0.70	current train_x shape: (800, 100001) train_x shape after mutual info selection: (800, 600) current test_x shape: (350, 100001) test_x shape after mutual info selection: (350, 600) train/val/test ratio: 0.7222222222222222 0.07092198581560284 0.20685579196217493 Train label is 1: 611 Train label is 0: 611 Validation label is 1: 9 validation label is 0: 111  train label ratio: 0.5005117707267144 0.49948822927328557 test label ratio: 0.49795918367346936 0.5020408163265306 f1 score: 0.8387096774193549 confusion matrix: [[119 4] [31 91]] train label ratio: 0.49948822927328557 0.5005117707267144 test label ratio: 0.5020408163265306 0.49795918367346936 f1 score: 0.8532110091743119 confusion matrix: [[120 2] [30 93]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.8465116279069768 confusion matrix: [[120 2] [31 91]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.8849557522123893 confusion matrix: [[118 4] [22 100]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.8425925925925927 confusion matrix: [[119 3] [31 91]] average f1 score: 0.8425925925925927 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] f1 score validation set: 0.7000000000000001	Comparing to random_state 15, increasing the k number decreased the accuracy difference a little bit (0.01). The next experiment will increase the k value more significantly to see if accuracies can get closer. diff = 0.14
31	force_alpha =True	5	average f1 score: 0.88	current train_x shape: (800, 100001) train_x shape after mutual info selection: (800, 800) current test_x shape: (350, 100001) test_x shape after mutual info selection: (350, 800)	The difference between accuracy decreased, it looks

	mutual_info_kbest = SelectKBest(score_func=mutual_info, k=800)		validation set f1 score: 0.80	train/val/test ratio: 0.7245017584994138 0.07033997655334115 0.205158264947245 Train label is 1: 618 Train label is 0: 618 Validation label is 1: 16 validation label is 0: 104  train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.8622222222222222 confusion matrix: [[120 4] [27 97]] train label ratio: 0.4994944388270981 0.5005055611729019 test label ratio: 0.5020242914979757 0.4979757085020243 f1 score: 0.8917748917748919 confusion matrix: [[119 4] [21 103]] train label ratio: 0.4994944388270981 0.5005055611729019 test label ratio: 0.5020242914979757 0.4979757085020243 f1 score: 0.8789237668161436 confusion matrix: [[122 1] [26 98]] train label ratio: 0.5005055611729019 0.4994944388270981 test label ratio: 0.4979757085020243 0.5020242914979757 f1 score: 0.8687782805429864 confusion matrix: [[122 2] [27 96]] train label ratio: 0.5005055611729019 0.4994944388270981 test label ratio: 0.4979757085020243 0.5020242914979757 f1 score: 0.8771929824561403 confusion matrix: [[119 5] [23 100]] average f1 score: 0.8771929824561403 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0] f1 score validation set: 0.7999999999999999	like select top 800 features after mutual information classification is a good number for our model. The next experiment will try a bigger number to see if the accuracy difference keeps decreasing. diff = 0.08
45	force_alpha =True SelectKBest(score_func=mutual_info, k=1000)	5	average f1 score: 0.79 validation set f1 score: 0.74	current train_x shape: (800, 100001) train_x shape after mutual info selection: (800, 1000) current test_x shape: (350, 100001) test_x shape after mutual info selection: (350, 1000) train/val/test ratio: 0.7241784037558685 0.07042253521126761 0.20539906103286384 Train label is 1: 617 Train label is 0: 617 Validation label is 1: 15 validation label is 0: 105  	increasing k to 1000, the accuracy difference decreased. However, the individual accuracy also decreased. Next experiment I want to try using the chi2 score function for top k feature selection. diff = 0.05

				train label ratio: 0.5005065856129686 0.49949341438703143 test label ratio: 0.4979757085020243 0.5020242914979757 f1 score: 0.858447488584475 confusion matrix: [[122 2] [29 94]] train label ratio: 0.5005065856129686 0.49949341438703143 test label ratio: 0.4979757085020243 0.5020242914979757 f1 score: 0.8636363636363636 confusion matrix: [[122 2] [28 95]] train label ratio: 0.49949341438703143 0.5005065856129686 test label ratio: 0.5020242914979757 0.4979757085020243 f1 score: 0.8130841121495326 confusion matrix: [[120 3] [37 87]] train label ratio: 0.49949341438703143 0.5005065856129686 test label ratio: 0.5020242914979757 0.4979757085020243 f1 score: 0.8493150684931507 confusion matrix: [[121 2] [31 93]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.7926267281105991 confusion matrix: [[115 8] [37 86]] average f1 score: 0.7926267281105991 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0] [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0] f1 score validation set: 0.7407407407407408	
52	force_alpha =True SelectKBest(chi2, k=500)	5	average f1 score: 0.82 validation set f1 score: 0.80	current train_x shape: (800, 100001) train_x shape after chi2 selection: (800, 500) current test_x shape: (350, 100001) test_x shape after chi2 selection: (350, 500) train/val/test ratio: 0.7222222222222222 0.07092198581560284 0.20685579196217493 Train label is 1: 611 Train label is 0: 611 Validation label is 1: 9 validation label is 0: 111  train label ratio: 0.5005117707267144 0.49948822927328557 test label ratio: 0.49795918367346936 0.5020408163265306 f1 score: 0.8571428571428572 confusion matrix: [[121 2] [29 93]] train label ratio: 0.49948822927328557 0.5005117707267144 test label ratio: 0.5020408163265306 0.49795918367346936 f1 score: 0.8425925925925926 confusion matrix: [[120 2] [32 91]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.867579908675799 confusion matrix: [[120 2] [27 95]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.7922705314009663 confusion matrix: [[119 3] [40 82]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.8219178082191781 confusion matrix: [[115 7] [32 90]] average f1 score: 0.8219178082191781 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0] f1 score validation set: 0.7999999999999999	Using chi2 score function to select top 500 features improved the accuracy difference to only 0.02. It seems like the chi2 score function feature selection is making the model more accurate. Next experiment will adjust the value of k and see the accuracy change. diff = 0.02
56	force_alpha =True SelectKBest(chi2, k=490)	5	average f1 score: 0.82 validation set f1 score: 0.83	current train_x shape: (800, 100001) train_x shape after chi2 selection: (800, 490) current test_x shape: (350, 100001) test_x shape after chi2 selection: (350, 490)	Changing the k value from 500 to 490 improved the accuracy difference to only 0.01. The validation set F1 score is higher than the cross-validation average F1 score.

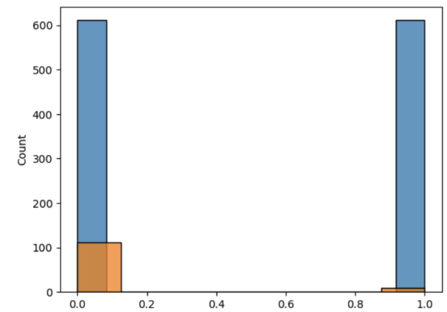
				train/val/test ratio: 0.7245017584994138 0.07033997655334115 0.205158264947245 Train label is 1: 618 Train label is 0: 618 Validation label is 1: 16 validation label is 0: 104  train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.8186046511627908 confusion matrix: [[121 3] [36 88]] train label ratio: 0.5005055611729019 0.4994944388270981 test label ratio: 0.4979757085020243 0.5020242914979757 f1 score: 0.8479262672811059 confusion matrix: [[122 2] [31 92]] train label ratio: 0.5005055611729019 0.4994944388270981 test label ratio: 0.4979757085020243 0.5020242914979757 f1 score: 0.8151658767772512 confusion matrix: [[122 2] [37 86]] train label ratio: 0.4994944388270981 0.5005055611729019 test label ratio: 0.5020242914979757 0.4979757085020243 f1 score: 0.9090909090909091 confusion matrix: [[121 2] [19 105]] train label ratio: 0.4994944388270981 0.5005055611729019 test label ratio: 0.5020242914979757 0.4979757085020243 f1 score: 0.81651376146789 confusion matrix: [[118 5] [35 89]] average f1 score: 0.81651376146789 [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0] [0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0] f1 score validation set: 0.8333333333333334	diff = 0.01

Stratified Cross Validation Experiments - Perceptron					
random_state	params applied	CV n_splits	average f1_score	screenshots	interpretation
20	SelectKBest(score_func=mutual_info, k=500)	5	average f1 score: 0.93 validation set f1 score: 0.60	current train_x shape: (800, 100001) train_x shape after mutual info selection: (800, 500) current test_x shape: (350, 100001) test_x shape after mutual info selection: (350, 500) train/val/test ratio: 0.7222222222222222 0.07092198581560284 0.20685579196217493 Train label is 1: 611 Train label is 0: 611 Validation label is 1: 9 validation label is 0: 111  train label ratio: 0.5005117707267144 0.49948822927328557 test label ratio: 0.49795918367346936 0.5020408163265306 f1 score: 0.9477911646586344 confusion matrix: [[114 9] [4 118]] train label ratio: 0.49948822927328557 0.5005117707267144 test label ratio: 0.5020408163265306 0.49795918367346936 f1 score: 0.9392712550607287 confusion matrix: [[114 8] [7 116]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9375 confusion matrix: [[108 14] [2 120]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.926829268292 6829 confusion matrix: [[112 10] [8 114]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9291338582677166 confusion matrix: [[108 14] [4 118]] average f1 score: 0.9291338582677166 [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0] f1 score validation set: 0.6	Use mutual info classification select k best to eliminate dependent features to reduce dimension of the training and validation set and reduce noise. The average cross validation f1 score is pretty high but the difference between the accuracy for the cross validation and the validation set can be improved. diff = 0.33
21	SelectKBest(chi2, k=500)	5	average f1 score: 0.93 validation set f1 score: 0.50	current train_x shape: (800, 100001) train_x shape after chi2 selection: (800, 500) current test_x shape: (350, 100001) test_x shape after chi2 selection: (350, 500)	Comparing the random_state 20, using chi2 score function to select top k best made the accuracy difference bigger. I will keep mutual info classfication select k best and tune the parameters in

				train/val/test ratio: 0.7225501770956316 0.07083825265643448 0.2066115702479339 Train label is 1: 612 Train label is 0: 612 Validation label is 1: 10 validation label is 0: 110  train label ratio: 0.5005107252298263 0.4994892747701363 test label ratio: 0.4979591836734936 0.5020408163265306 f1 score: 0.897338403041825 confusion matrix: [[100 23] [4 118]] train label ratio: 0.5005107252298263 0.4994892747701363 test label ratio: 0.4979591836734936 0.5020408163265306 f1 score: 0.904 confusion matrix: [[108 15] [9 113]] train label ratio: 0.4994892747701363 0.5005107252298263 test label ratio: 0.5020408163265306 0.4979591836734936 f1 score: 0.9435483870967742 confusion matrix: [[114 8] [6 117]] train label ratio: 0.4994892747701363 0.5005107252298263 test label ratio: 0.5020408163265306 0.4979591836734936 f1 score: 0.9280000000000002 confusion matrix: [[111 11] [7 116]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9317269076305221 confusion matrix: [[111 11] [6 116]] average f1 score: 0.9317269076305221 [0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0] [0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0] f1 score validation set: 0.5	the next experiment. diff = 0.43
22	SelectKBest(score_func=mutual_info, k=500) eta0=0.1	5	average f1 score: 0.96 validation set f1 score: 0.53	current train_x shape: (800, 100001) train_x shape after mutual info selection: (800, 500) current test_x shape: (350, 100001) test_x shape after mutual info selection: (350, 500) train/val/test ratio: 0.7238542890716804 0.07050528789659224 0.2056404230317274 Train label is 1: 616 Train label is 0: 616 Validation label is 1: 14 validation label is 0: 106  	apply a smaller learning rate than the default 1. The difference between the accuracies is still pretty big. Will increase the learning rate and see what happens. diff = 0.43

<table>
<tr><td></td><td></td><td></td><td></td><td>train label ratio: 0.49949238578680205 0.500507614213198
test label ratio: 0.5020242914979757 0.4979757085020243
f1 score: 0.9448818897637796
confusion matrix:
 [[113 10]
 [4 120]]
train label ratio: 0.500507614213198 0.49949238578680205
test label ratio: 0.4979757085020243 0.5020242914979757
f1 score: 0.968
confusion matrix:
 [[118   6]
 [2 121]]
train label ratio: 0.5 0.5
test label ratio: 0.5 0.5
f1 score: 0.9302325581395349
confusion matrix:
 [[108 15]
 [3 120]]
train label ratio: 0.5 0.5
test label ratio: 0.5 0.5
f1 score: 0.8970588235294118
confusion matrix:
 [[96 27]
 [1 122]]
train label ratio: 0.5 0.5
test label ratio: 0.5 0.5
f1 score: 0.9635627530364373
confusion matrix:
 [[118   5]
 [4 119]]
average f1 score: 0.9635627530364373
[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,
0, 1, 0]
[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0
 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 0]
f1 score validation set: 0.5333333333333333</td><td></td></tr>
<tr><td>23</td><td>SelectKBest(score_func=mutual_info, k=500)

eta0=0.5</td><td>5</td><td>average f1 score: 0.95 validation set f1 score: 0.62</td><td>current train_x shape: (800, 100001)
train_x shape after mutual info selection: (800, 500)
current test_x shape: (350, 100001)
test_x shape after mutual info selection: (350, 500)

train/val/test ratio: 0.7222222222222222 0.07092198581560284 0.20685579196217493
Train label is 1: 611 Train label is 0: 611
Validation label is 1: 9 validation label is 0: 111
 

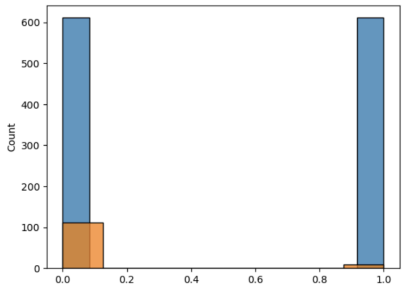
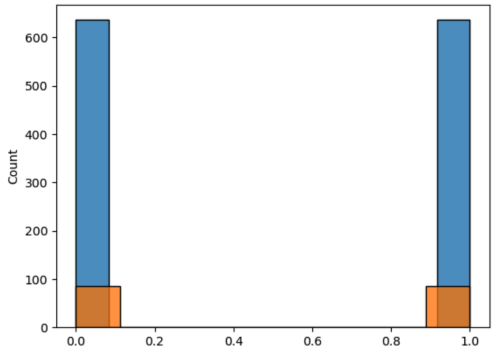
train label ratio: 0.5005117707267144 0.49948822927328557
test label ratio: 0.4979591836734693 6 0.5020408163265306
f1 score: 0.9224489795918367
confusion matrix:
 [[113 10]
 [9 113]]
train label ratio: 0.49948822927328557 0.5005117707267144
test label ratio: 0.5020408163265306 0.49795918367346936
f1 score: 0.9477911646586347
confusion matrix:
 [[114   8]
 [5 118]]
train label ratio: 0.5 0.5
test label ratio: 0.5 0.5
f1 score: 0.933852140077821
confusion matrix:
 [[107 15]
 [2 120]]
train label ratio: 0.5 0.5
test label ratio: 0.5 0.5
f1 score: 0.921161825726141
confusion matrix:
 [[114   8]
 [11 111]]
train label ratio: 0.5 0.5
test label ratio: 0.5 0.5
f1 score: 0.9482071713147411
confusion matrix:
 [[112 10]
 [3 119]]
average f1 score: 0.9482071713147411
[0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0,
0, 0, 0]
[0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0]
f1 score validation set: 0.6206896551724138</td><td>increasing the learning rate helped increase the validation set f1 score and decrease the difference between accuracies. It looks like our model works better with a faster learning rate.

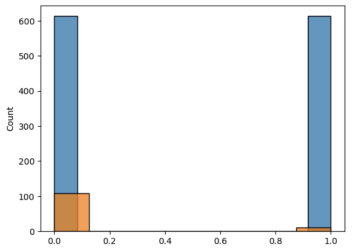
Next will try to increase the learning rate to > 1

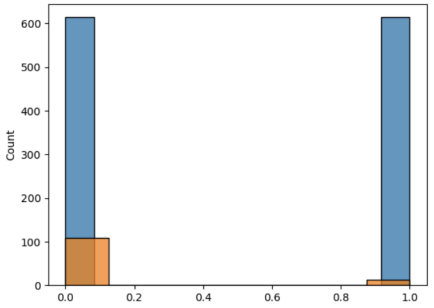
diff = 0.33</td></tr>
<tr><td>24</td><td>SelectKBest(score_func=mutual_info, k=500)

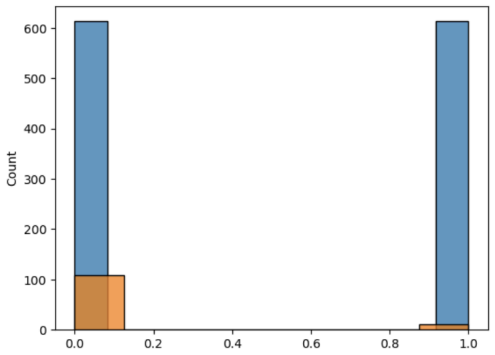
eta0=2.0</td><td>5</td><td>average f1 score: 0.92 validation set f1 score: 0.52</td><td>current train_x shape: (800, 100001)
train_x shape after mutual info selection: (800, 500)
current test_x shape: (350, 100001)
test_x shape after mutual info selection: (350, 500)</td><td>It looks like further increasing the learning rate did not work to bring the accuracy difference closer.

The next experiment will specify the amount of epochs by changing</td></tr>
</table>

				train/val/test ratio: 0.7222222222222222 0.07092198581560284 0.20685579196217493 Train label is 1: 611 Train label is 0: 611 Validation label is 1: 9 validation label is 0: 111  train label ratio: 0.5005117707267144 0.4994882292732855 test label ratio: 0.4979591836734694 0.5020408163265306 f1 score: 0.9291338582677166 confusion matrix: [[109 14] [4 118]] train label ratio: 0.4994882292732855 0.5005117707267144 test label ratio: 0.5020408163265306 0.4979591836734694 f1 score: 0.9333333333333333 confusion matrix: [[109 13] [4 119]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9797570850202428 confusion matrix: [[118 4] [1 121]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9359999999999999 confusion matrix: [[111 11] [5 117]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9224489795918367 confusion matrix: [[112 10] [9 113]] average f1 score: 0.9224489795918367 [0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1] [0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] f1 score validation set: 0.5161290322580644	max_iter. diff = 0.40
26	SelectKBest(score_func=mutual_info, k=500) eta0=0.5 max_iter=100	5	average f1 score: 0.93 validation set f1 score: 0.59	current train_x shape: (800, 100001) train_x shape after chi2 selection: (800, 500) current test_x shape: (350, 100001) test_x shape after chi2 selection: (350, 500) train/val/test ratio: 0.7090301003344481 0.09587513935340022 0.19509476031215162 Train label is 1: 636 Train label is 0: 636 Validation label is 1: 86 validation label is 0: 86  	The default max_iter is 1000, this experiment we set it to 100. The accuracy difference decreased a little bit but not significantly. Next will try a bigger value for max iteration. diff = 0.34

				train label ratio: 0.5004916420845624 0.4995083579154376 test label ratio: 0.4980392156862745 0.5019607843137255 f1 score: 0.9402985074626865 confusion matrix: [[113 15] [1 126]] train label ratio: 0.4995083579154376 0.5004916420845624 test label ratio: 0.5019607843137255 0.4980392156862745 f1 score: 0.9144981412639406 confusion matrix: [[109 18] [5 123]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9618320610687023 confusion matrix: [[118 9] [1 126]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9077490774907748 confusion matrix: [[106 21] [4 123]] train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9402985074626865 confusion matrix: [[112 15] [1 126]] average f1 score: 0.9402985074626865 [0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1] [0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1] f1 score validation set: 0.7951807228915662	
27	SelectKBest(score_func=mutual_info, k=500) eta0=0.5 max_iter=500	5	average f1 score: 0.94 validation set f1 score: 0.63	current train_x shape: (800, 100001) train_x shape after mutual info selection: (800, 500) current test_x shape: (350, 100001) test_x shape after mutual info selection: (350, 500) train/val/test ratio: 0.7228773584905660 0.07075471698113207 0.20636792452830188 Train label is 1: 613 Train label is 0: 613 Validation label is 1: 11 validation label is 0: 109  train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9083969465648856 confusion matrix: [[103 20] [4 119]] train label ratio: 0.5005096839959225 0.4994903160407745 test label ratio: 0.4979591836734694 0.5020408163265306 f1 score: 0.9397590361445783 confusion matrix: [[113 10] [5 117]] train label ratio: 0.5005096839959225 0.4994903160407745 test label ratio: 0.4979591836734694 0.5020408163265306 f1 score: 0.9375 confusion matrix: [[109 14] [2 120]] train label ratio: 0.4994903160407745 0.5005096839959225 test label ratio: 0.5020408163265306 0.4979591836734694 f1 score: 0.7961165048543689 confusion matrix: [[59 63] [0 123]] train label ratio: 0.4994903160407745 0.5005096839959225 test label ratio: 0.5020408163265306 0.4979591836734694 f1 score: 0.9374999999999999 confusion matrix: [[109 13] [3 120]] average f1 score: 0.9374999999999999 [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] f1 score validation set: 0.6249999999999999	It looks like max_iter has minimal impact on our model. Either it is applied or not, the accuracy difference between the average cross-validation score and validation set score are all around 0.30. The problem might not be because of max epochs, it could be something else. Even when the learning rate is adjusted the difference between accuracies is still similar to when it is not specified. Next experiment we will adjust the number of k selected

					diff = 0.31
28	SelectKBest(score_func=mutual_info, k=400)	5	average f1 score: 0.89 validation set f1 score: 0.65	current train_x shape: (800, 100001) train_x shape after mutual info selection: (800, 400) current test_x shape: (350, 100001) test_x shape after mutual info selection: (350, 400) train/val/test ratio: 0.7232037691401649 0.0706713780918728 0.2061248527679623 Train label is 1: 614 Train label is 0: 614 Validation label is 1: 12 validation label is 0: 108  :rain label ratio: 0.5 0.5 :est label ratio: 0.5 0.5 f1 score: 0.9163346613545816 :onfusion matrix: [[110 13] [8 115]] :rain label ratio: 0.5 0.5 :est label ratio: 0.5 0.5 f1 score: 0.9370078740157479 :onfusion matrix: [[111 12] [4 119]] :rain label ratio: 0.5 0.5 :est label ratio: 0.5 0.5 f1 score: 0.9520000000000001 :onfusion matrix: [[115 8] [4 119]] :rain label ratio: 0.5005086469989827 0.49949135300101727 :est label ratio: 0.49795918367346936 0.5020408163265306 f1 score: 0.9382716049382717 :onfusion matrix: [[116 7] [8 114]] :rain label ratio: 0.49949135300101727 0.5005086469989827 :est label ratio: 0.5020408163265306 0.49795918367346936 f1 score: 0.8853754940711462 :onfusion matrix: [[104 18] [11 112]] verage f1 score: 0.8853754940711462 [0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0,), 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1,), 0, 0] [0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0] f1 score validation set: 0.6451612903225806	It seems like decreasing the k in select k best improved the accuracy difference between cross validation and validation set Next experiment I will make the k even smaller. diff = 0.24
29	SelectKBest(score_func=mutual_info, k=250)	5	average f1 score: 0.92 validation set f1 score: 0.60	current train_x shape: (800, 100001) train_x shape after mutual info selection: (800, 250) current test_x shape: (350, 100001) test_x shape after mutual info selection: (350, 250)	It seems like keep decreasing the k number of selected top record did not improve the accuracy. diff = 0.32

				train/val/test ratio: 0.722877358490566 0.07075471698113207 0.20636792452830188 Train label is 1: 613 Train label is 0: 613 Validation label is 1: 11 validation label is 0: 109  train label ratio: 0.5 0.5 test label ratio: 0.5 0.5 f1 score: 0.9135802469135802 confusion matrix: [[114 9] [12 111]] train label ratio: 0.5005096839959225 0.49949031600407745 test label ratio: 0.4979591836734694 0.5020408163265306 f1 score: 0.9205020920502092 confusion matrix: [[116 7] [12 110]] train label ratio: 0.5005096839959225 0.49949031600407745 test label ratio: 0.4979591836734694 0.5020408163265306 f1 score: 0.9306122448979592 confusion matrix: [[114 9] [8 114]] train label ratio: 0.49949031600407745 0.5005096839959225 test label ratio: 0.5020408163265306 0.4979591836734694 f1 score: 0.9392712550607287 confusion matrix: [[114 8] [7 116]] train label ratio: 0.49949031600407745 0.5005096839959225 test label ratio: 0.5020408163265306 0.4979591836734694 f1 score: 0.9294605809128631 confusion matrix: [[116 6] [11 112]] average f1 score: 0.9294605809128631 [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0] [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 1 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] f1 score validation set: 0.6	

References

[1] "sklearn.feature_selection.SelectKBest," *Scikit-learn*. https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html#sklearn.feature_selection.SelectKBest [accessed Oct. 25. 2023.]

[2] "Sklearn.feature_selection.chi2," *Scikit-learn*. https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html#sklearn.feature_selection.chi2 [accessed Oct. 25. 2023.]

[3] "sklearn.feature_selection.mutual_info_classif," *Scikit-learn*. https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_classif.html#sklearn.feature_selection.mutual_info_classif [accessed Oct. 25. 2023.]

[4] "RandomOverSampler — Version 0.11.0." https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.RandomOverSampler.html#imblearn.over_sampling.RandomOverSampler [accessed Oct. 25. 2023.]

[5] "sklearn.model_selection.StratifiedKFold," *Scikit-learn*. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html [accessed Oct. 25. 2023.]

[6] https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.BernoulliNB.html [accessed Oct. 25. 2023.]

[7] "sklearn.linear_model.Perceptron," *Scikit-learn*. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Perceptron.html [accessed Oct. 28. 2023.]

[8] J. Brownlee, "Perceptron algorithm for classification in Python," *MachineLearningMastery.com*, Aug. 05, 2020. https://machinelearningmastery.com/perceptron-algorithm-for-classification-in-python/ [accessed Oct. 28. 2023.]