Optimizing for AutoPVDBOW

The mean of Y is 0.4998928073748526

NN\_1:

elif clf\_type == 'clf\_NN':  
 inputVars = len(XVars)  
 if inputVars / 5 < 8:  
 L\_1 = 8  
 else:  
 L\_1 = round(inputVars / 8)  
  
 if binary:  
 L\_3 = 2  
 else:  
 L\_3 = 7  
  
 clf = MLPClassifier(hidden\_layer\_sizes=(L\_1, 8, L\_3), activation='relu', solver='sgd',  
 max\_iter=10000, random\_state=random\_state)

Dataset: Auto

Train Scores: Accuracy: 0.590 Precision: 0.598Recall: 0.794

Test Scores Accuracy: 0.573 Precision: 0.566 Recall: 0.838

Dataset: AutoPVDBOW

Train Scores: Accuracy: 0.590 Precision: 0.592 Recall: 0.708

Test Scores Accuracy: 0.582 Precision: 0.573 Recall: 0.683

NN\_3:

elif clf\_type == 'clf\_NN':  
 inputVars = len(XVars)  
 if inputVars / 5 < 8:  
 L\_1 = inputVars  
 L\_2 = inputVars  
 else:  
 L\_1 = inputVars  
 L\_2 = inputVars  
  
 if binary:  
 L\_3 = 2  
 else:  
 L\_3 = 7  
  
 clf = MLPClassifier(hidden\_layer\_sizes=(L\_1, L\_2, L\_3), activation='relu', solver='sgd',  
 max\_iter=10000, random\_state=random\_state)

Dataset: Auto

Train Scores: Accuracy: 0.59 Precision: 0.598 Recall: 0.795

Test Scores Accuracy: 0.573 Precision: 0.567 Recall: 0.839

Dataset: AutoPVDBOW

Train Scores: Accuracy: 0.591 Precision: 0.598 Recall: 0.682

Test Scores Accuracy: 0.594 Precision: 0.592 Recall: 0.645

NN\_7: L3 = 5

elif clf\_type == 'clf\_NN':  
 inputVars = len(XVars)  
 if inputVars / 5 < 8:  
 L\_1 = 2\*inputVars  
 L\_2 = inputVars  
 else:  
 L\_1 = 2\*inputVars  
 L\_2 = inputVars  
  
 if binary:  
 L\_3 = 5  
 else:  
 L\_3 = 7  
  
 clf = MLPClassifier(hidden\_layer\_sizes=(L\_1, L\_2, L\_3), activation='tanh', solver='sgd',  
 max\_iter=10000, random\_state=random\_state)

Dataset: Auto

Train Scores: Accuracy: 0.586 Precision: 0.601 Recall: 0.598

Test Scores Accuracy: 0.599 Precision: 0.592 Recall: 0.648

Dataset: AutoPVDBOW

Train Scores: Accuracy: 0.594 Precision: 0.606 Recall: 0.664

Test Scores Accuracy: 0.587 Precision: 0.573 Recall: 0.694

NN\_8: solver = ‘adam’

elif clf\_type == 'clf\_NN':  
 inputVars = len(XVars)  
 if inputVars / 5 < 8:  
 L\_1 = 2\*inputVars  
 L\_2 = inputVars  
 else:  
 L\_1 = 2\*inputVars  
 L\_2 = inputVars  
  
 if binary:  
 L\_3 = 2  
 else:  
 L\_3 = 7  
  
 clf = MLPClassifier(hidden\_layer\_sizes=(L\_1, L\_2, L\_3), activation='relu', solver='adam',  
 max\_iter=10000, random\_state=random\_state)

Dataset: Auto

Train Scores: Accuracy: 0.601 Precision: 0.619 Recall: 0.674

Test Scores Accuracy: 0.587 Precision: 0.59 Recall: 0.777

Dataset: AutoPVDBOW

Train Scores: Accuracy: 0.96 Precision: 0.937 Recall: 0.991 - overfit

Test Scores Accuracy: 0.526 Precision: 0.527 Recall: 0.621

NN\_9: solver = adam, early\_stopping = True

elif clf\_type == 'clf\_NN':  
 inputVars = len(XVars)  
 if inputVars / 5 < 8:  
 L\_1 = 2\*inputVars  
 L\_2 = inputVars  
 else:  
 L\_1 = 2\*inputVars  
 L\_2 = inputVars  
  
 if binary:  
 L\_3 = round(inputVars/2)  
 else:  
 L\_3 = 7  
  
 clf = MLPClassifier(hidden\_layer\_sizes=(L\_1,L\_2,L\_3), activation='relu', solver='adam',  
 max\_iter=10000, random\_state=random\_state,early\_stopping=True)

Dataset: Auto

Train Scores: Accuracy: 0.595 Precision: 0.6 Recall: 0.702

Test Scores Accuracy: 0.581 Precision: 0.563 Recall: 0.736

Dataset: AutoPVDBOW

Train Scores: Accuracy: 0.605 Precision: 0.621 Recall: 0.657

Test Scores Accuracy: 0.592 Precision: 0.585 Recall: 0.793

GB\_1:

else : #clf\_type == 'clf\_GradientBoosting'  
 clf = GradientBoostingClassifier(random\_state=random\_state,learning\_rate=0.01,  
 min\_samples\_split=0.025,min\_samples\_leaf=0.01,max\_depth=8)

Dataset: Auto

Train Scores: Accuracy: 0.506 Precision: 0.491 Recall: 0.412

Test Scores Accuracy: 0.513 Precision: 0.500 Recall: 0.416

Dataset: AutoPVDBOW

Train Scores: Accuracy: 0.512 Precision: 0.5 Recall: 0.441

Test Scores Accuracy: 0.511 Precision: 0.498 Recall: 0.397

GB\_2:

else : #clf\_type == 'clf\_GradientBoosting'  
 clf = GradientBoostingClassifier(random\_state=random\_state,learning\_rate=0.0001,  
 min\_samples\_split=0.01,min\_samples\_leaf=0.001,max\_depth=12)

Dataset: Auto

Train Scores: Accuracy: 0.529 Precision: 0.529 Recall: 1.0

Test Scores Accuracy: 0.523 Precision: 0.523 Recall: 1.0

Dataset: AutoPVDBOW

Train Scores: Accuracy: 0.529 Precision: 0.529 Recall: 1.0

Test Scores Accuracy: 0.524 Precision: 0.524 Recall: 1.0

Optimizing for Auto

The mean of Y is 0.4998928073748526

Logreg\_1:

elif clf\_type == 'clf\_logreg':  
 clf = LogisticRegression(solver='lbfgs', penalty='l2', max\_iter=10000,  
 random\_state=random\_state)

Dataset: Auto

Train Scores: Accuracy: 0.589 Precision: 0.598 Recall: 0.665

Test Scores Accuracy: 0.594 Precision: 0.594 Recall: 0.607

Logreg\_2: fit\_intercept = False

elif clf\_type == 'clf\_logreg':  
 clf = LogisticRegression(solver='lbfgs', penalty='l2', max\_iter=10000,  
 random\_state=random\_state,fit\_intercept=False)

Dataset: Auto

Train Scores: Accuracy: 0.591 Precision: 0.599 Recall: 0.667

Test Scores Accuracy: 0.594 Precision: 0.589 Recall: 0.633

Dataset: AutoPVDBOW

Train Scores: Accuracy: 0.594 Precision: 0.599 Recall: 0.679

Test Scores Accuracy: 0.596 Precision: 0.601 Recall: 0.583

Logreg\_3: penalty=elasticnet

elif clf\_type == 'clf\_logreg':  
 clf = LogisticRegression(solver='saga', penalty='elasticnet', max\_iter=10000,l1\_ratio=0.5,  
 random\_state=random\_state,fit\_intercept=False)

Dataset: Auto

Train Scores: Accuracy: 0.591 Precision: 0.599 Recall: 0.666

Test Scores Accuracy: 0.593 Precision: 0.588 Recall: 0.63

Dataset: AutoPVDBOW

Train Scores: Accuracy: 0.594 Precision: 0.599 Recall: 0.68

Test Scores Accuracy: 0.6 Precision: 0.599 Recall: 0.615

Logreg\_4: L1\_ratio = 0.4

elif clf\_type == 'clf\_logreg':  
 clf = LogisticRegression(solver='saga', penalty='elasticnet', max\_iter=10000,l1\_ratio=0.4,  
 random\_state=random\_state,fit\_intercept=False)

Dataset: Auto

Train Scores: Accuracy: 0.591 Precision: 0.599 Recall: 0.666

Test Scores Accuracy: 0.593 Precision: 0.588 Recall: 0.63

Dataset: AutoPVDBOW

Train Scores: Accuracy: 0.594 Precision: 0.599 Recall: 0.679

Test Scores Accuracy: 0.601 Precision: 0.6 Recall: 0.614

Test Scores Accuracy: 0.599 Precision: 0.592 Recall: 0.648

SGD\_1:

if clf\_type == 'clf\_SGD':  
 clf = SGDClassifier(random\_state=random\_state, shuffle=True, loss='log', max\_iter=10000)

Classification type: TS\_Classifier:Classification algo: clf\_SGD

Dataset: Auto

Train Scores: Accuracy: 0.575 Precision: 0.597 Recall: 0.965

Test Scores Accuracy: 0.591 Precision: 0.6 Recall: 0.968

SGD\_2: fit\_intercept = False

if clf\_type == 'clf\_SGD':  
 clf = SGDClassifier(random\_state=random\_state, shuffle=True, loss='log', max\_iter=10000, fit\_intercept=False)

Dataset: Auto

Train Scores: Accuracy: 0.59 Precision: 0.609 Recall: 0.694

Test Scores Accuracy: 0.591 Precision: 0.595 Recall: 0.754

Dataset: AutoPVDBOW

Train Scores: Accuracy: 0.577 Precision: 0.601 Recall: 0.689

Test Scores Accuracy: 0.574 Precision: 0.601 Recall: 0.929

SGD\_3: learning\_rate = adaptive

if clf\_type == 'clf\_SGD':  
 clf = SGDClassifier(random\_state=random\_state, shuffle=True, loss='log', max\_iter=10000, fit\_intercept=False,  
 learning\_rate='adaptive', eta0=0.0001)

Dataset: Auto

Train Scores: Accuracy: 0.583 Precision: 0.597 Recall: 0.606

Test Scores Accuracy: 0.595 Precision: 0.596 Recall: 0.6

Dataset: AutoPVDBOW

Train Scores: Accuracy: 0.588 Precision: 0.595 Recall: 0.662

Test Scores Accuracy: 0.595 Precision: 0.59 Recall: 0.631

SGD\_4: n\_iter\_no\_change = 30

if clf\_type == 'clf\_SGD':  
 clf = SGDClassifier(random\_state=random\_state, shuffle=True, loss='log', max\_iter=10000, fit\_intercept=False,  
 learning\_rate='adaptive', eta0=0.0001, n\_iter\_no\_change=30)

Dataset: Auto

Train Scores: Accuracy: 0.587 Precision: 0.597 Recall: 0.635

Test Scores Accuracy: 0.594 Precision: 0.59 Recall: 0.625

Dataset: AutoPVDBOW

Train Scores: Accuracy: 0.59 Precision: 0.596 Recall: 0.671

Test Scores Accuracy: 0.597 Precision: 0.591 Recall: 0.644

SGD\_5: n\_iter\_no\_change = 100

if clf\_type == 'clf\_SGD':  
 clf = SGDClassifier(random\_state=random\_state, shuffle=True, loss='log', max\_iter=10000, fit\_intercept=False,  
 learning\_rate='adaptive', eta0=0.0001, n\_iter\_no\_change=100)

Dataset: Auto

Train Scores: Accuracy: 0.59 Precision: 0.599 Recall: 0.663

Test Scores Accuracy: 0.595 Precision: 0.591 Recall: 0.631

Dataset: AutoPVDBOW

Train Scores: Accuracy: 0.591 Precision: 0.6 Recall: 0.681

Test Scores Accuracy: 0.597 Precision: 0.59 Recall: 0.652

SGD\_5: eta0=0.001

if clf\_type == 'clf\_SGD':  
 clf = SGDClassifier(random\_state=random\_state, shuffle=True, loss='log', max\_iter=10000, fit\_intercept=False,  
 learning\_rate='adaptive', eta0=0.001, n\_iter\_no\_change=100)

Dataset: Auto

Train Scores: Accuracy: 0.591 Precision: 0.599 Recall: 0.666

Test Scores Accuracy: 0.594 Precision: 0.591 Recall: 0.627

Dataset: AutoPVDBOW

Train Scores: Accuracy: 0.594 Precision: 0.599 Recall: 0.679

Test Scores Accuracy: 0.597 Precision: 0.593 Recall: 0.631