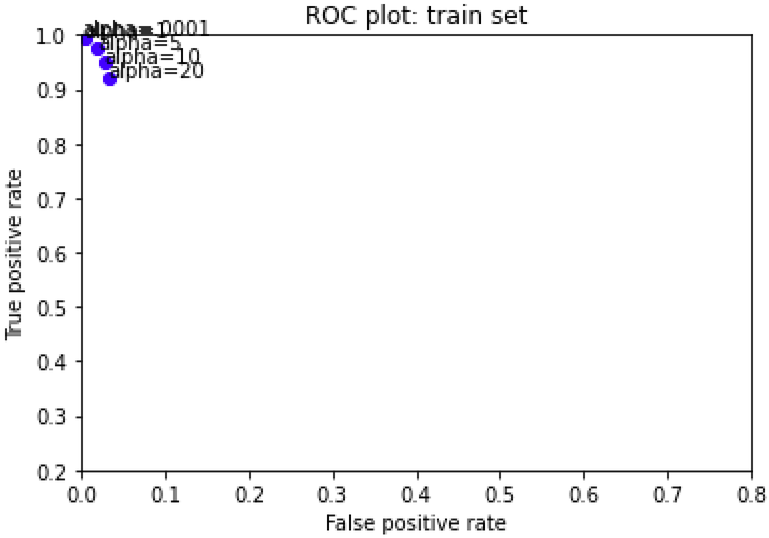
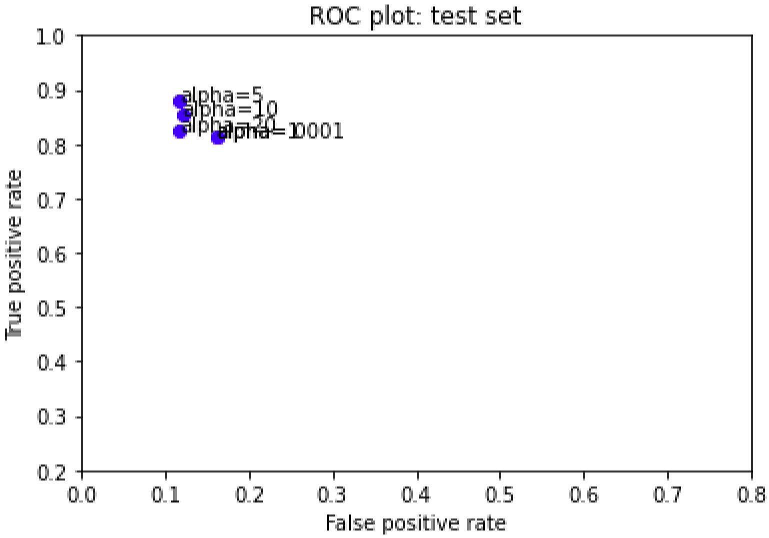

ROC plot: train set
alpha=1
alpha=0.001
alpha=5
alpha=10
alpha=20
True positive rate
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
False positive rate
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8


ROC plot: test set
alpha=5
alpha=10
alpha=20
alpha=1
alpha=0.001
True positive rate
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
False positive rate
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

```
from sklearn import linear_model
svm = linear_model.SGDClassifier(alpha=.0001)
svm.fit(X_train, y_train)

svm_performance_train = BinaryClassificationPerformance(svm.predict(X_train), y_train, 'alpha=.0001')
svm_performance_train.compute_measures()
print(svm_performance_train.performance_measures)
```

```
{'Pos': 10000, 'Neg': 10000, 'TP': 10000, 'TN': 10000, 'FP': 0, 'FN': 0, 'Accuracy': 1.0, 'Precision': 1.0, 'Recall': 1.0, 'desc': 'alpha
=.0001'}
```

```
from sklearn import linear_model
svm1 = linear_model.SGDClassifier(alpha=1)
svm1.fit(X_train, y_train)

svm1_performance_train = BinaryClassificationPerformance(svm1.predict(X_train), y_train, 'alpha=1')
svm1_performance_train.compute_measures()
print(svm1_performance_train.performance_measures)
```

```
{'Pos': 10000, 'Neg': 10000, 'TP': 9960, 'TN': 9962, 'FP': 38, 'FN': 40, 'Accuracy': 0.9961, 'Precision': 0.9961992398479695, 'Recall': 0.
996, 'desc': 'alpha=1'}
```

```
from sklearn import linear_model
svm2 = linear_model.SGDClassifier(alpha=5)
svm2.fit(X_train, y_train)

svm2_performance_train = BinaryClassificationPerformance(svm2.predict(X_train), y_train, 'alpha=5')
svm2_performance_train.compute_measures()
print(svm2_performance_train.performance_measures)
```

```
{'Pos': 10000, 'Neg': 10000, 'TP': 9756, 'TN': 9813, 'FP': 187, 'FN': 244, 'Accuracy': 0.97845, 'Precision': 0.981192798954038, 'Recall':
0.9756, 'desc': 'alpha=5'}
```

```python
from sklearn import linear_model
svm3 = linear_model.SGDClassifier(alpha=10)
svm3.fit(X_train, y_train)

svm3_performance_train = BinaryClassificationPerformance(svm3.predict(X_train), y_train, 'alpha=10')
svm3_performance_train.compute_measures()
print(svm3_performance_train.performance_measures)
```

```
{'Pos': 10000, 'Neg': 10000, 'TP': 9490, 'TN': 9733, 'FP': 267, 'FN': 510, 'Accuracy': 0.96115, 'Precision': 0.9726350312596085, 'Recall':
0.949, 'desc': 'alpha=10'}
```

```python
from sklearn import linear_model
svm4 = linear_model.SGDClassifier(alpha=20)
svm4.fit(X_train, y_train)

svm4_performance_train = BinaryClassificationPerformance(svm4.predict(X_train), y_train, 'alpha=20')
svm4_performance_train.compute_measures()
print(svm4_performance_train.performance_measures)
```

```
{'Pos': 10000, 'Neg': 10000, 'TP': 9221, 'TN': 9677, 'FP': 323, 'FN': 779, 'Accuracy': 0.9449, 'Precision': 0.9661567476948868, 'Recall':
0.9221, 'desc': 'alpha=20'}
```