CSE 6363-007: Machine Learning Assignment 3

Name: Ananthula, Vineeth Kumar. UTA ID: 1001953922

Non-Linear SVM:

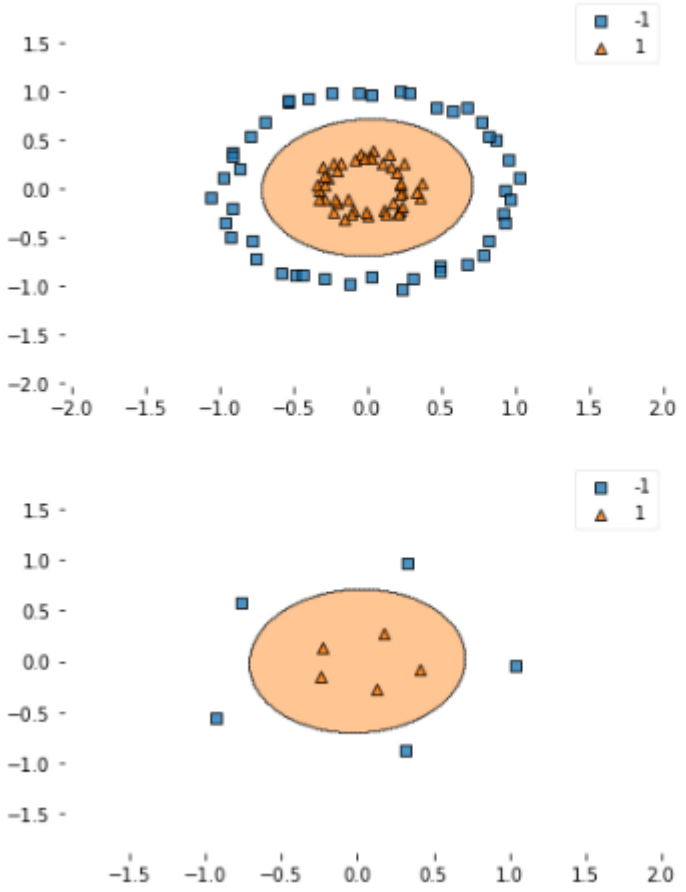
The following code has been added:

```
    # TODO: the negative case
    else:
        L1 = alpha1 + s*(alpha2-L)
        H1 = alpha1 + s*(alpha2-H)

        f1 = y1*(e1+self._b) - alpha1*k11 - s*alpha2*k12
        f2 = y2*(e2+self._b) - alpha2*k22 - s*alpha1*k12

        Lobj = L1*f1 + L*f2 + s*L1*L*k12 + 0.5*L*L*k22 + 0.5*L1*L1*k11
        Hobj = H1*f1 + H*f2 + s*H1*H*k12 + 0.5*H*H*k22 + 0.5*H1*H1*k11

        if Lobj < Hobj- (1e-3):
            a2 = L
        elif Lobj > Hobj+ (1e-3):
            a2 = H
        else:
            a2 = alpha2

        # print(f"[DEBUG] smo_step: eta = {eta}")

    if np.abs(a2 - alpha2) < 1e-3 * (a2 + alpha2 + 1e-3):
        return 0

    a1 = alpha1 + s * (alpha2 - a2)
```

Plotting Dataset

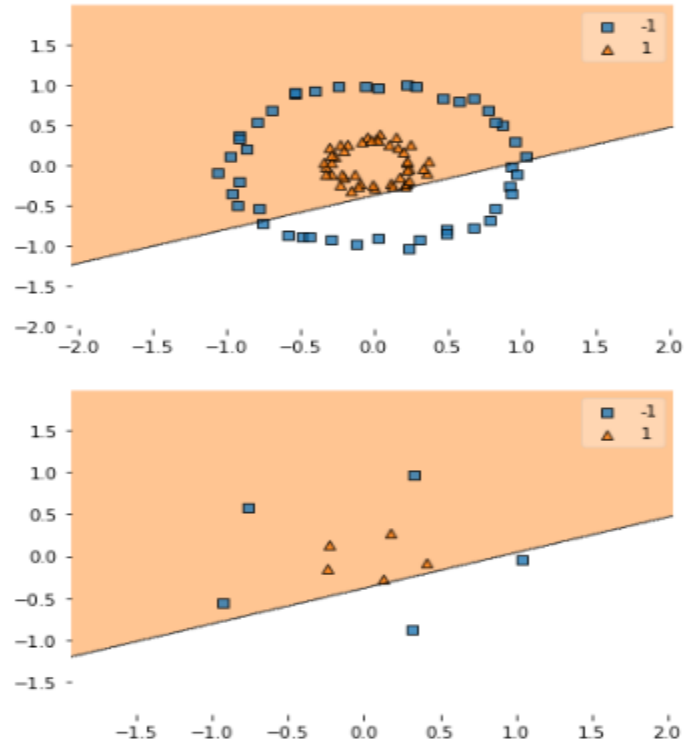
Using sklearn.SVC with Poly Kernel: (accuracy = 100 %)

```
Accuracy of poly SKLEARN SVC NON-LINEAR on non-linear dataset =  100.0
```



Using sklearn.SVC with Linear Kernel: (accuracy = 100 %)

```
Accuracy of linear SKLEARN SVC LINEAR on non-linear dataset =  100.0
```



Using defined with Poly(accuracy = 100 %) and Linear (accuracy = 70 %) Kernel:

Calculating accuracy for Non-Linear my defined SVM for kernel is poly and linear

```
pred = model.predict(X_test)
pred1 = model1.predict(X_test)
print("Accuracy of Non-Linear my defined SVM for kernel is poly =", accuracy_score(y_test,pred)*100)
print("Accuracy of Non-Linear my defined SVM for kernel is linear =", accuracy_score(y_test,pred1)*100)
```

```
Accuracy of Non-Linear my defined SVM for kernel is poly = 100.0
Accuracy of Non-Linear my defined SVM for kernel is linear = 70.0
```

My Defined SVM – Kernel Poly	SKLEARN SVC – Kernel Poly	My Defined SVM – Kernel Linear	SKLEARN SVC – Kernel Linear
100 %	100%	70%	100%

Multi-class SVM:

Accuracy of linear SVM with my defined Multi-class SVM

```
preds = multisvm.predict(X_test_multi)
print("Linear My Defined Multi-class SVM, accuracy =", accuracy_score(y_test_multi,preds)*100)
```

```
Linear My Defined Multi-class SVM, accuracy = 93.33333333333333
```

Accuracy of non linear SVM with my defined Multi-class SVM

```
multisvm2 = MultiSvm(k=3, kernel='poly', c=5.0)   # k here is the number of classes
multisvm2.fit(X_train_multi ,y_train_multi)
```

```
preds = multisvm2.predict(X_test_multi)
print("Non Linear My Defined SVM multi class, accuracy =", accuracy_score(y_test_multi,preds)*100)
```

```
Non Linear My Defined SVM multi class, accuracy = 100.0
```

SKLEARN SVC LINEAR AND NON-LINEAR

```
lin_clf = sksvm.LinearSVC()
lin_clf.fit(X_train_multi, y_train_multi)

pred = lin_clf.predict(X_test_multi)
print("SK Learn Linear Multi-class SVM, accuracy =",accuracy_score(y_test_multi,pred)*100)

clf = sksvm.SVC(kernel='poly',degree=2)
clf.fit(X_train_multi, y_train_multi)
pred = clf.predict(X_test_multi)
print("SK Learn non Linear Multi-class SVM, accuracy =", accuracy_score(y_test_multi,pred)*100)
```

```
SK Learn Linear Multi-class SVM, accuracy = 100.0
SK Learn non Linear Multi-class SVM, accuracy = 100.0
```

My Defined Linear Multi-class SVM	SKLEARN SVC Linear	My Defined Non-Linear Multi-class SVM	SKLEARN SVC Non-Linear
93.33 %	100%	100%	100%