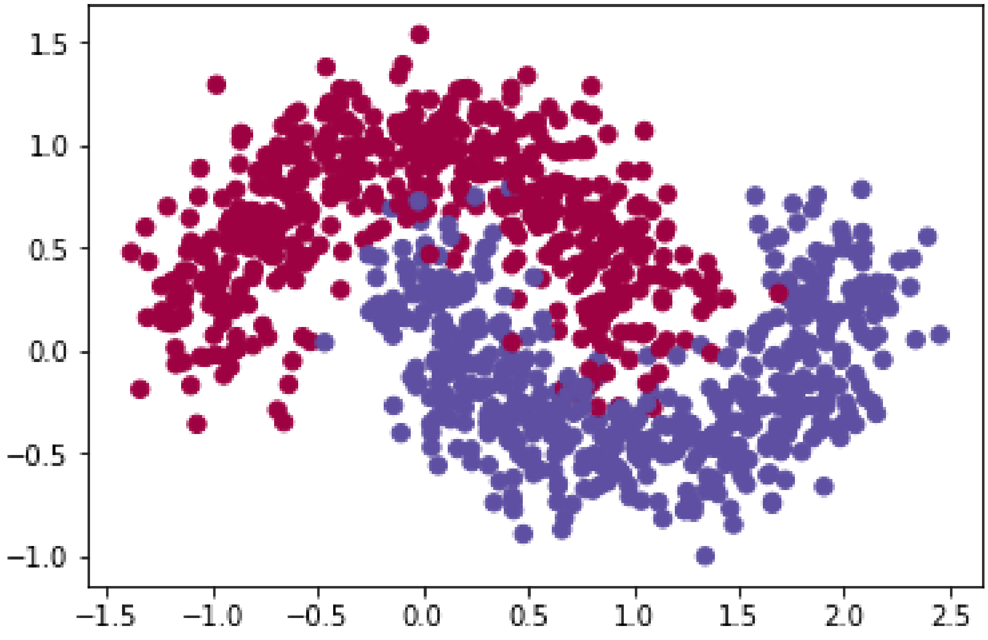
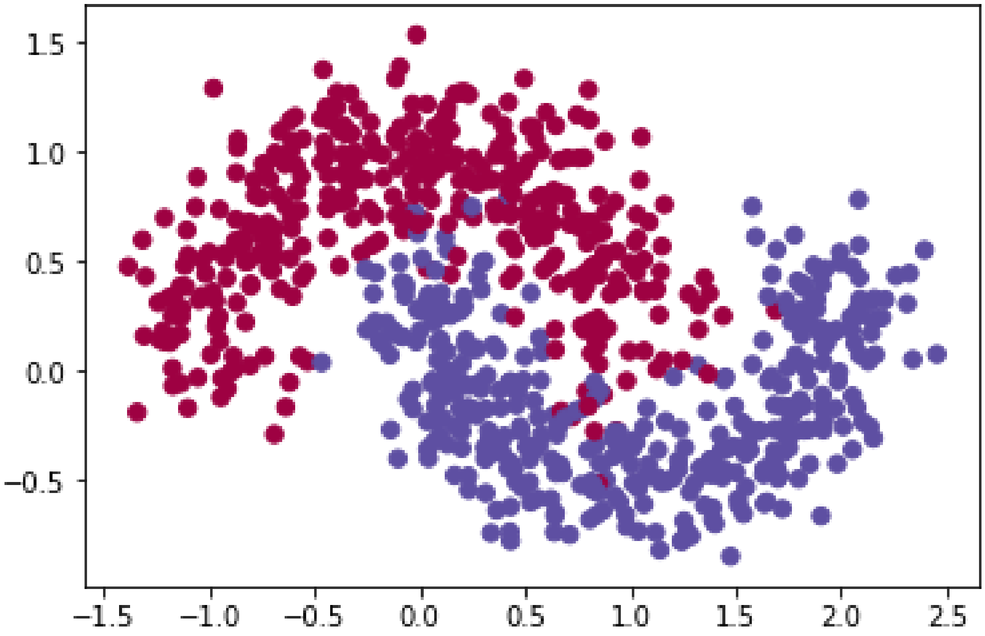
ROC-AUC_k_nn

July 21, 2018

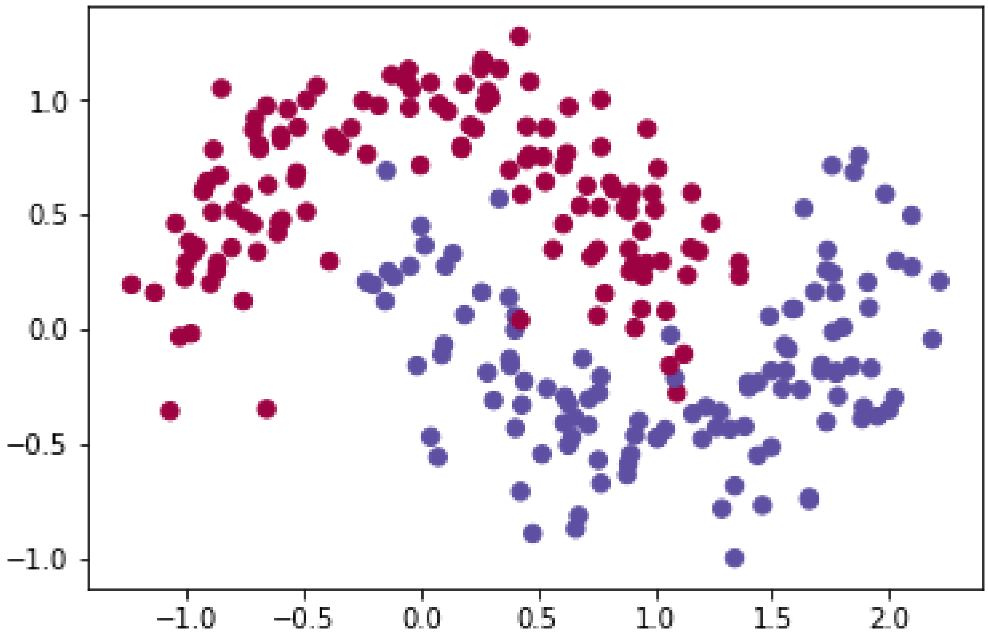
```
In [1]: import numpy as np

In [2]: import matplotlib.pyplot as plt

In [3]: from sklearn.metrics import roc_auc_score

In [4]: from sklearn.neighbors import KNeighborsClassifier

In [5]: from sklearn import datasets #Importamos el conjunto de datos

In [6]: from sklearn.model_selection import train_test_split

In [7]: np.random.seed(0)

In [8]: X, y = datasets.make_moons(1000, noise=0.20)

In [9]: #Dividimos nuestros datos en "conjunto de entrenamiento y de prueba

In [10]: X_train, X_test, y_train, y_test = train_test_split(X, y)

In [11]: plt.scatter(X[:,0], X[:,1], s=40, c=y, cmap=plt.cm.Spectral)

Out[11]: <matplotlib.collections.PathCollection at 0x109a4cd50>
```



```
In [12]: plt.scatter(X_train[:,0], X_train[:,1], s=40, c=y_train, cmap=plt.cm.Spectral)

Out[12]: <matplotlib.collections.PathCollection at 0x109b7ac90>
```



```
In [13]: plt.scatter(X_test[:,0], X_test[:,1], s=40, c=y_test, cmap=plt.cm.Spectral)

Out[13]: <matplotlib.collections.PathCollection at 0x109c69450>
```



```
In [14]: classifiers=[(KNeighborsClassifier(n_neighbors=3),"3NN"),
                     (KNeighborsClassifier(n_neighbors=11),"11NN"),
                     (KNeighborsClassifier(n_neighbors=5, metric='manhattan'),"5NNm"),
                     (KNeighborsClassifier(n_neighbors=1),"1NN")]
```

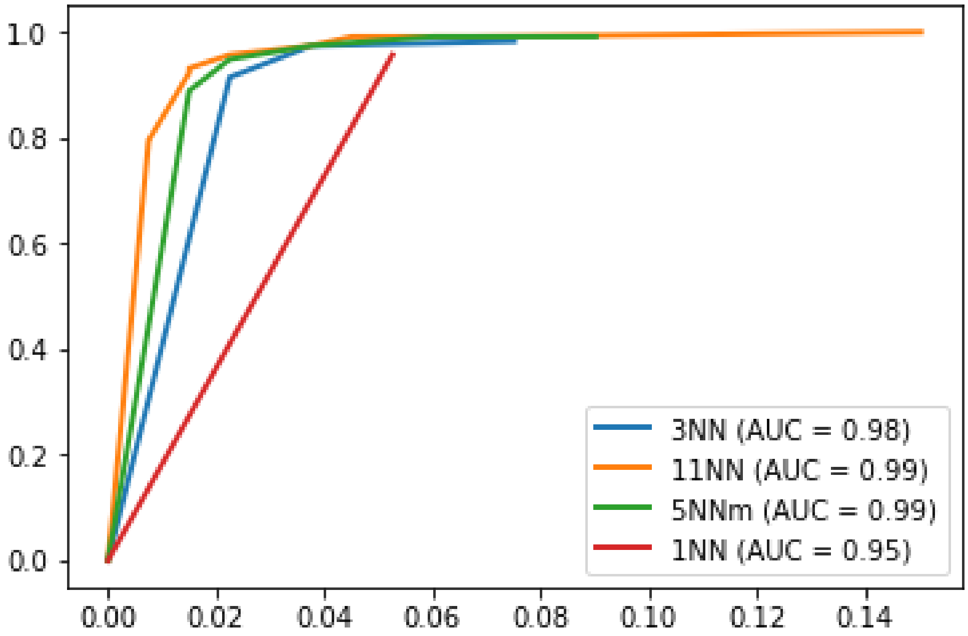
```
In [15]: classifiers
```

```
Out[15]: [(KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                       metric_params=None, n_jobs=1, n_neighbors=3, p=2,
                       weights='uniform'), '3NN'),
          (KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                       metric_params=None, n_jobs=1, n_neighbors=11, p=2,
                       weights='uniform'), '11NN'),
          (KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='manhattan',
                       metric_params=None, n_jobs=1, n_neighbors=5, p=2,
                       weights='uniform'), '5NNm'),
          (KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                       metric_params=None, n_jobs=1, n_neighbors=1, p=2,
                       weights='uniform'), '1NN')]
```

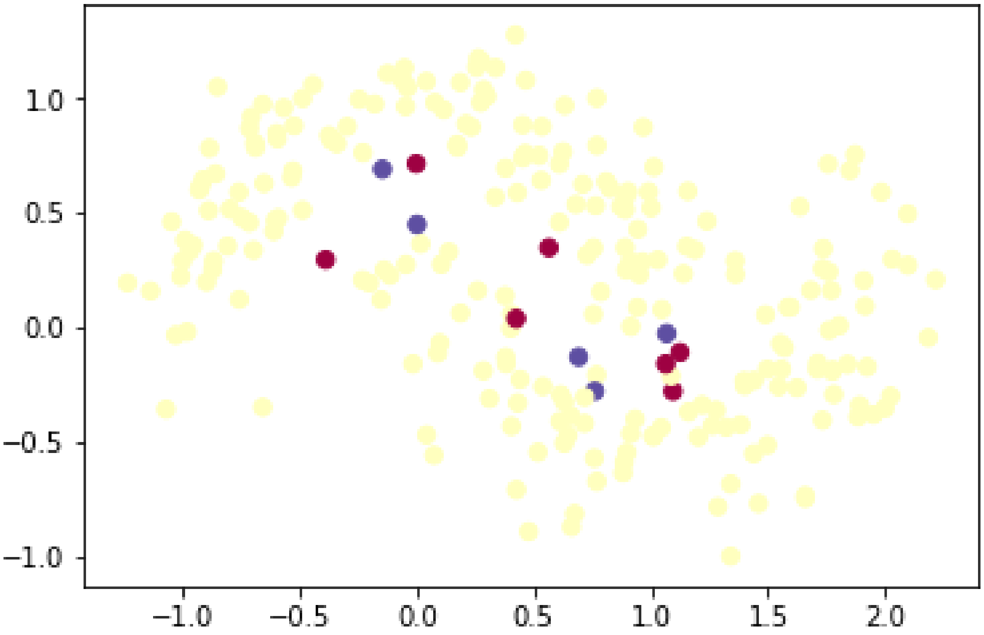
```
In [16]: #Para cada clasificador se grafica la ROC
```

```
In [17]: for clf, name in classifiers:
             clf.fit (X_train,y_train)
             ROC=[]
             for gamma in np.linspace(0,1,1000):
```

```
        err1=np.count_nonzero(clf.predict_proba(X_test[y_test==0,:])[:,1]<=gamma)
        err2=np.count_nonzero(clf.predict_proba(X_test[y_test==1,:])[:,1]>gamma)
        err1=float(err1)/np.count_nonzero(y_test==0)
        err2=float(err2)/np.count_nonzero(y_test==1)
        ROC.append([err1,err2])
    ROC=np.array(ROC)
    ROC=ROC[::-1,:]
    auc=roc_auc_score(y_test,clf.predict_proba(X_test)[:,1])
    plt.plot(1-ROC[:,0],ROC[:,1], linewidth=2, label="%s (AUC = %.2f)" %(name,auc))
    plt.legend()
```



```
In [18]: diff_test=y_test-clf.predict_proba(X_test)[:,1]

In [19]: plt.scatter(X_test[:,0], X_test[:,1], s=40, c=diff_test, cmap=plt.cm.Spectral)

Out[19]: <matplotlib.collections.PathCollection at 0x109d66190>
```


1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0