Lab 11 Python for Artificial Intelligence

```
------------------------------------------------------------------------
```
Q1: Run the following programme

```python
from sklearn import svm
from sklearn import datasets
from sklearn import metrics
X, y = datasets.load_iris(return_X_y=True)
clf = svm.SVC()
clf = svm.SVC(gamma=0.001, C=100.)
clf.fit(X, y)

import pickle
s = pickle.dumps(clf)
clf2 = pickle.loads(s)
y_pred=clf2.predict(X)
print("Accuracy:",metrics.accuracy_score(y, y_pred))
```

- **Change the values of** gamma **and** C **and check the Accuracy.**
- **Print the Precision:",metrics.precision_score(y_test, y_pred)**
- **Print the "Recall:",metrics.recall_score(y_test, y_pred)**
- **Use different values for X.**
- **Test SVM using different dataset: Diabetes dataset**

```
------------------------------------------------------------------------
```
Q2: Run the following programme

```python
# Sample Decision Tree Classifier
from sklearn import datasets
from sklearn import metrics
from sklearn.tree import DecisionTreeClassifier
# load the iris datasets
dataset = datasets.load_iris()
# fit a CART model to the data
model = DecisionTreeClassifier()
model.fit(dataset.data, dataset.target)
print(model)
# make predictions
expected = dataset.target
predicted = model.predict(dataset.data)
# summarize the fit of the model
print(metrics.classification_report(expected, predicted))
print(metrics.confusion_matrix(expected, predicted))
```

- **Test the Decision Tree algorithm using different dataset: Diabetes dataset**
- **Print the Accuracy.**

```
------------------------------------------------------------------------
```

```python
# Q3: Run the following programme

from sklearn import cluster, datasets
X_iris, y_iris = datasets.load_iris(return_X_y=True)
k_means = cluster.KMeans(n_clusters=3)
k_means.fit(X_iris)
KMeans(n_clusters=3)
print(k_means.labels_[::10])
print(y_iris[::10])
print(kmeans.cluster_centers_)

    - Run the K-Means using a different number of clusters: 2 and 4.
    - Run plt.scatter(X[:,0],X[:,1], c=kmeans.labels_, cmap='rainbow')

-------------------------------------------------------------------
```