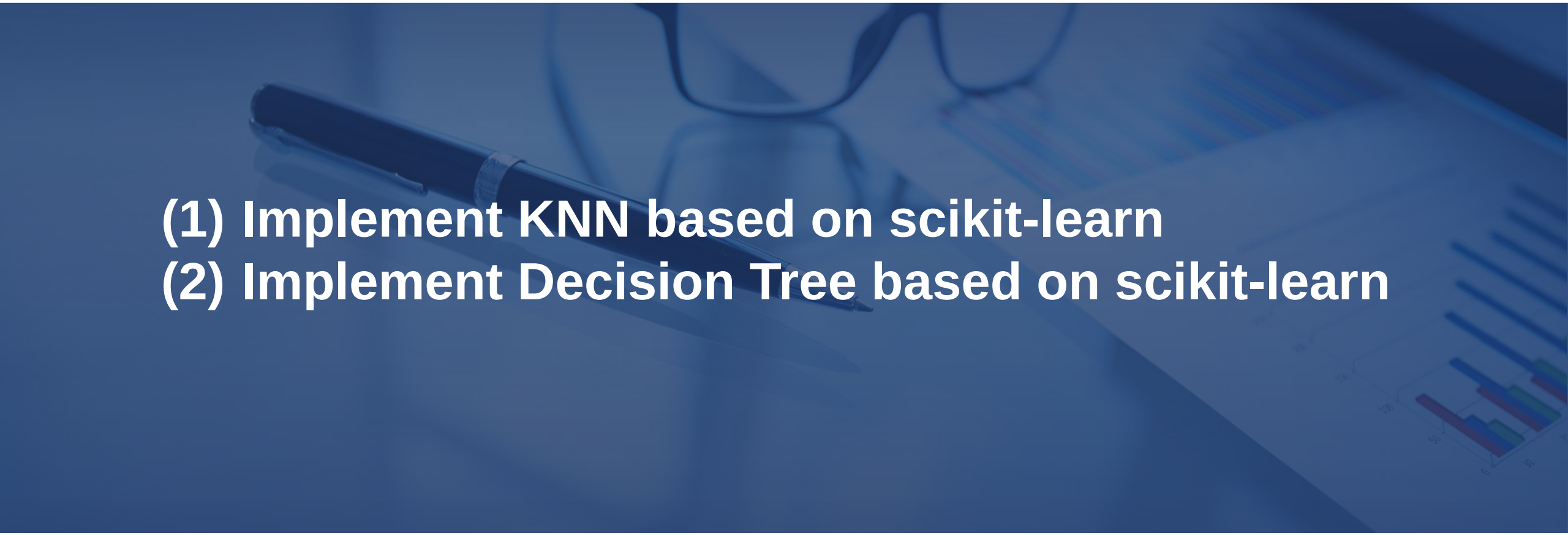
Knowledge Discovery and Data Mining

Lab 6 KNN & Decision Tree

Xuan Song
Songx@sustech.edu.cn

Topics



(1) Implement KNN based on scikit-learn
(2) Implement Decision Tree based on scikit-learn

Data

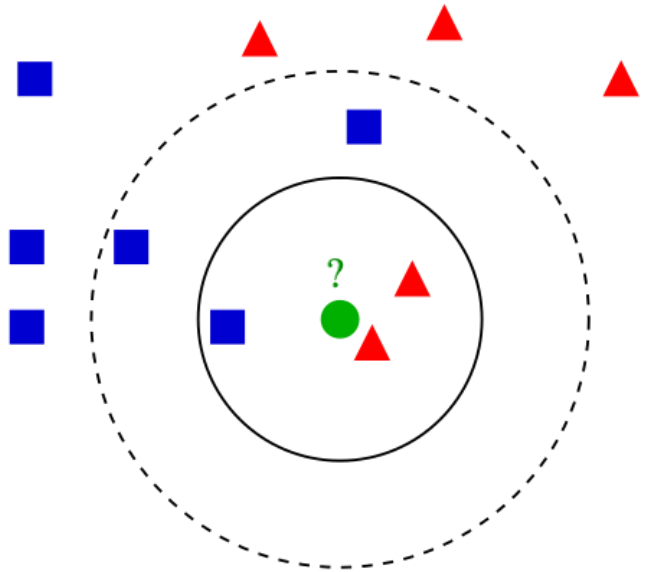
- Dataset :Iris Data Set
  - Attribute Information:
    - 1. sepal length in cm
    - 2. sepal width in cm
    - 3. petal length in cm
    - 4. petal width in cm
  - Class:
    - Iris Setosa
    - Iris Versicolour
    - Iris Virginica

Attribute				Class
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa
5.4	3.9	1.7	0.4	Iris-setosa
4.6	3.4	1.4	0.3	Iris-setosa
5	3.4	1.5	0.2	Iris-setosa
4.4	2.9	1.4	0.2	Iris-setosa
4.9	3.1	1.5	0.1	Iris-setosa
5.4	3.7	1.5	0.2	Iris-setosa
4.8	3.4	1.6	0.2	Iris-setosa
4.8	3	1.4	0.1	Iris-setosa
4.3	3	1.1	0.1	Iris-setosa
5.8	4	1.2	0.2	Iris-setosa
5.7	4.4	1.5	0.4	Iris-setosa
5.4	3.9	1.3	0.4	Iris-setosa

k Nearest Neighbor (kNN) Classification

- sklearn.neighbors.KNeighborsClassifier

*class sklearn.neighbors.KNeighborsClassifier(n_neighbors=3, *, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None, **kwargs)*



https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier

Implement KNN based on scikit-learn

1. Load data from csv files.
2. Data cleaning.
3. Get explanatory variables and dependent variables from the given data.

4. Build a KNN model based on scikit-learn. (e.g., k = 1~12)

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=k)
```

5. Fit the model.

```
knn.fit(X_train, y_train)
```

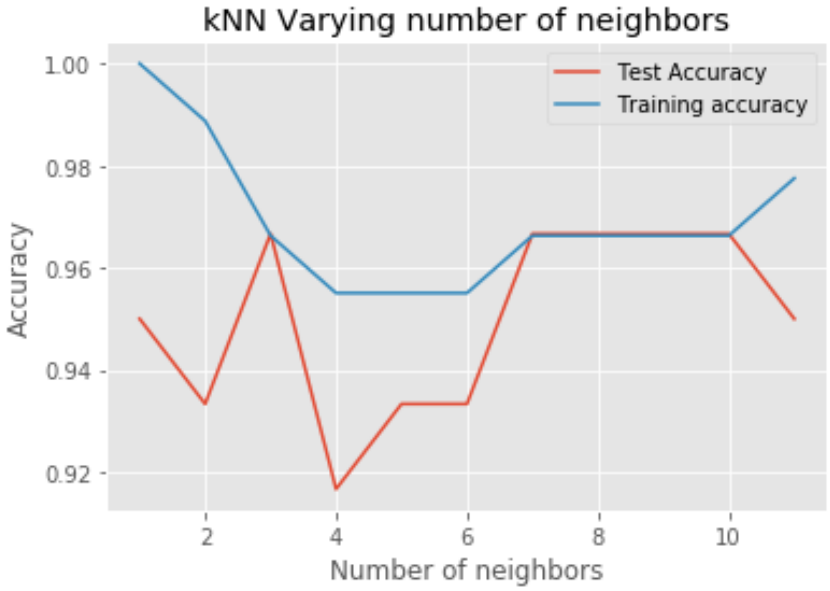
Implement KNN based on scikit-learn

6. Compute accuracy on the training set and test set.

```
1: for-loop
2: train_accuracy[i] = knn.score(X_train, y_train)
   test_accuracy[i]  = knn.score(X_test, y_test)
```

7. Generate a plot of accuracy to choose a good value of *k*.

```
plt.title('kNN Varying number of neighbors')
plt.plot(neighbors, test_accuracy, label='Test Accuracy')
plt.plot(neighbors, train_accuracy, label='Training accuracy')
plt.legend()
plt.xlabel('Number of neighbors')
plt.ylabel('Accuracy')
plt.show()
```



Implement KNN based on scikit-learn

8. Setup a KNN classifier with *k* neighbors

```
knn = KNeighborsClassifier(n_neighbors=3)
```

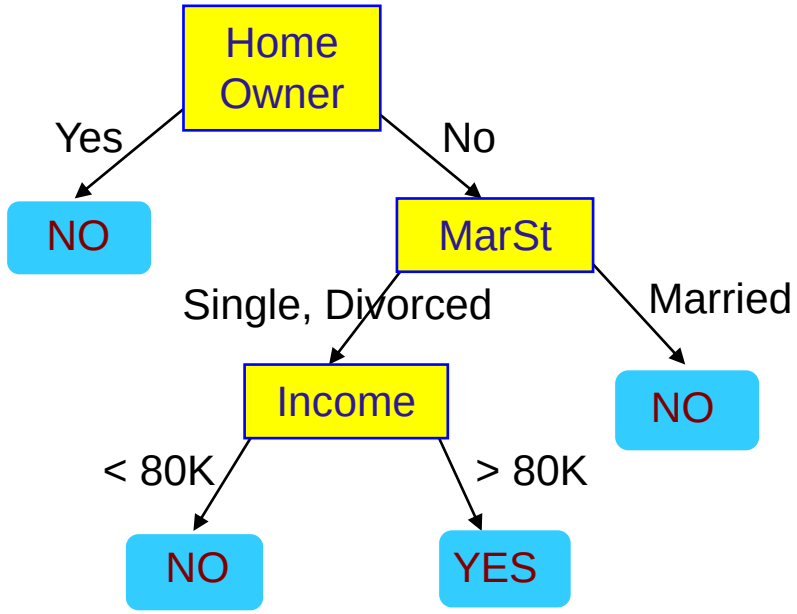
9. Fit the model and get the accuracy of test set.

```
knn.fit(X_train,y_train)
knn.score(X_test,y_test)
```

Decision Tree

- sklearn.tree.DecisionTreeClassifier

class sklearn.tree.DecisionTreeClassifier(, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, ccp_alpha=0.0)*



https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

Implement Decision Tree based on scikit-learn

- Prerequisite:
  1. Dataset: Iris Data Set
  2. Installation of Visualization Environment (optional):
     (1) Install **Graphviz** from http://www.graphviz.org/download/
     (2) Add the bin directory of **Graphviz** to Path
     (3) *pip install graphviz* for your jupyter kernel
     (4) *pip install pydotplus* for your jupyter kernel

If jupyter notebook can't find the Graphviz when you run your code, try to add the following code:

```
import os
os.environ["PATH"] += os.pathsep + 'C:/Program Files/Graphviz/bin/'
```

where, 'C:/Program Files/Graphviz/bin/' is the installation location of your Graphviz.

Implement Decision Tree based on scikit-learn

1. Load data from csv files.
2. Data cleaning.
3. Get explanatory variables and dependent variables from the given data.
4. Build a Decision Tree model based on scikit-learn.

```
from sklearn import tree
clf = tree.DecisionTreeClassifier()
```

5. Fit the model.

```
clf = clf.fit(X1,y1)
```

Implement Decision Tree based on scikit-learn

6. Visualization

The first option

```
from sklearn import tree
from IPython.display import Image
import pydotplus

feature_name = ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
class_name = ['setosa', 'versicolor', 'virginica']
### use the function export_graphviz() to export the tree to Graphviz format ###
dot_data = tree.export_graphviz(clf,  filled=True, rounded=True,
                                feature_names=feature_name, class_names=class_name, special_characters=True, out_file=None)
graph = pydotplus.graph_from_dot_data(dot_data)
Image(graph.create_png())
```

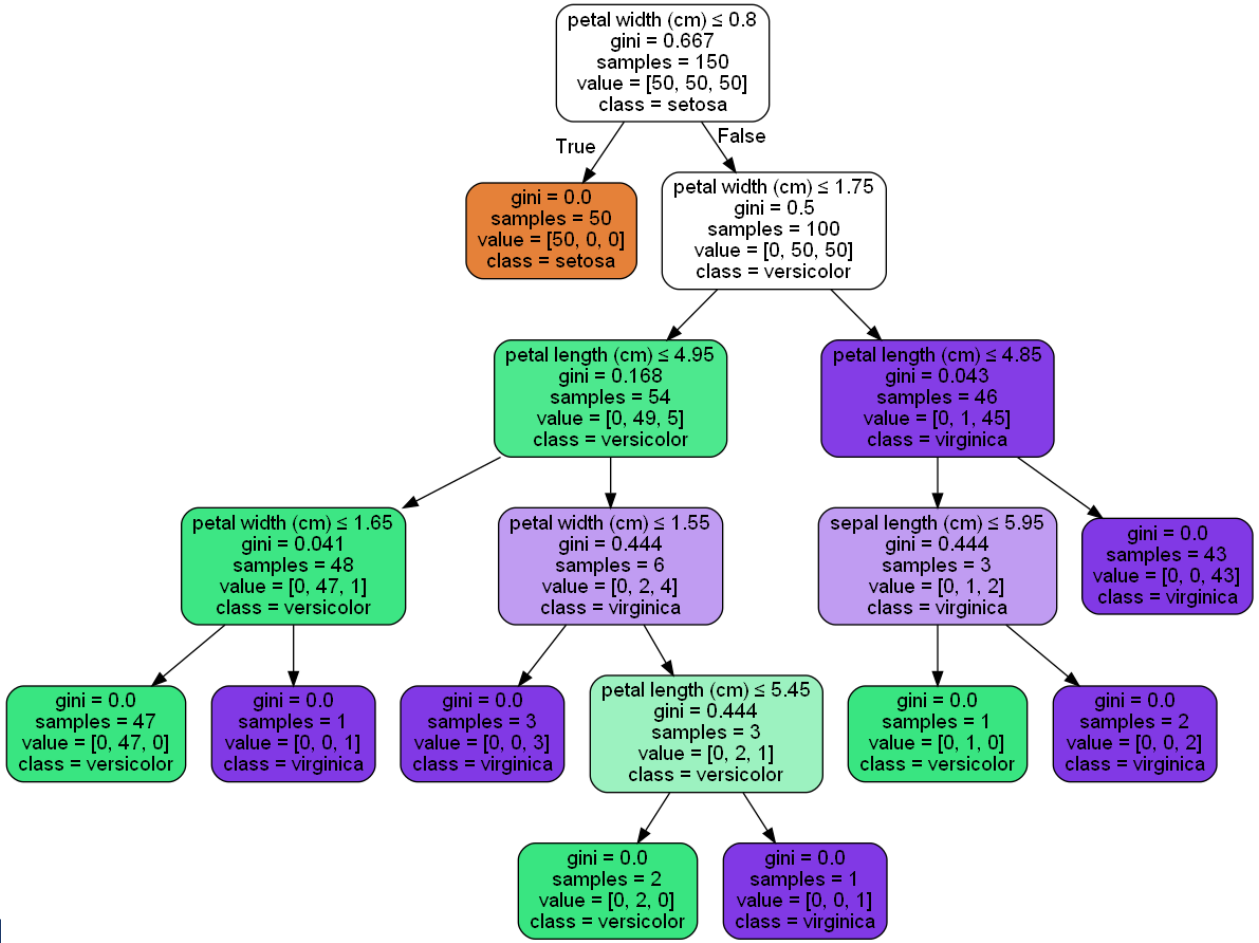
The second option

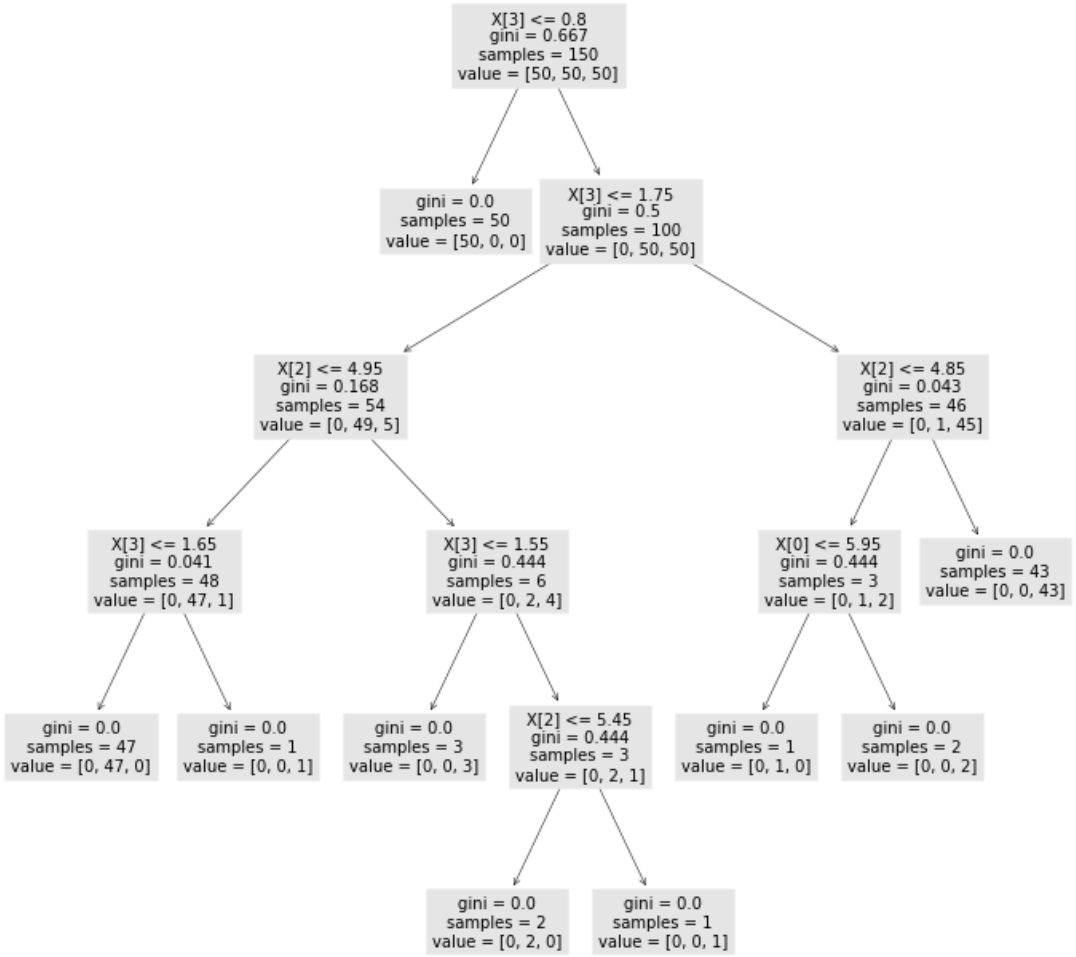
```
plt.figure(figsize=(12,12))  # set plot size (denoted in inches)
tree.plot_tree(clf, fontsize=10)
plt.show()
```

If you want to predict other test data, you can use clf.predict(x_test), or refer to https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

Implement Decision Tree based on scikit-learn

6. Visualization





If you want to predict other test data, you can use clf.predict(x_test), or refer to https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

Tasks

- Implement KNN based on a given data set.
- Implement Decision Tree based on a given data set.

End of Lab6