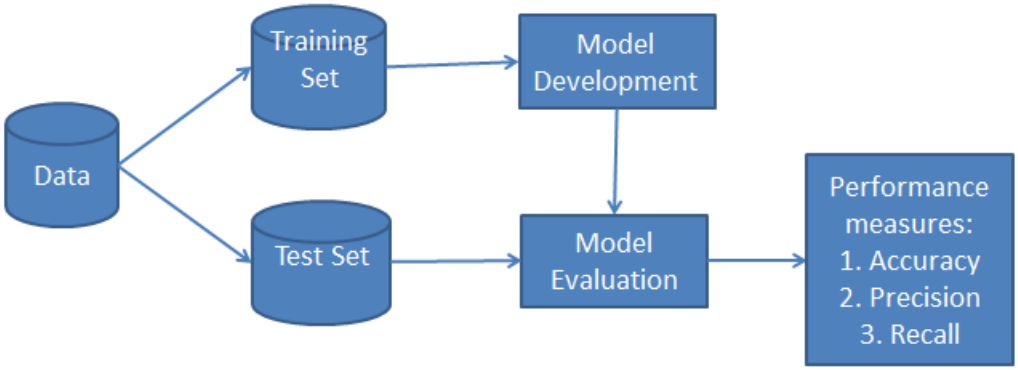
Bài thực hành

Phân loại dữ liệu với các phương pháp tập hợp mô hình

1. Ví dụ minh họa sử dụng giải thuật Bagging của cây quyết định

Trong ví dụ này, học viên làm quen với:

- Nắm được các bước phân loại dữ liệu



- Sử dụng giải thuật Bagging của cây quyết định để phân loại dữ liệu.

```
# Nạp các gói thư viện cần thiết
import pandas as pd
from sklearn import tree
from sklearn.ensemble import BaggingClassifier
import numpy as np

# Đọc dữ liệu iris từ UCI (https://archive.ics.uci.edu/ml/datasets/Iris)
# hoặc từ thư viện scikit-learn
# Tham khảo https://scikit-
learn.org/stable/auto_examples/datasets/plot_iris_dataset.html
from sklearn import datasets
from sklearn.model_selection import train_test_split
iris = datasets.load_iris()
columns=["Petal Length","Petal Width","Sepal Length","Sepal Width"];
X = pd.DataFrame(iris.data, columns=columns)
y = iris.target
print(X.describe())

# Sử dụng nghi thức kiểm tra hold-out
# Chia dữ liệu ngẫu nhiên thành 2 tập dữ liệu con:
# training set và test set theo tỷ lệ 70/30
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
#print(X_train.shape, y_train.shape)
#print(X_test.shape, y_test.shape)
```

```
#print(X_train.head())

# Xây dựng bagging cua 50 cay quyet dinh
model =
BaggingClassifier(base_estimator=tree.DecisionTreeClassifier(),n_estimators=5
0)
model.fit(X_train, y_train)

# Dự đoán nhãn tập kiểm tra
prediction = model.predict(X_test)
#print(prediction)

# Tính độ chính xác
print("Do chinh xác cua mo hinh voi nghi thuc kiem tra hold-out: %.3f" %
model.score(X_test, y_test))

# Sử dụng nghi thức kiểm tra chéo k-fold
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score

# Thực hiện nghi thức kiểm tra 5 fold
nFold = 5;
model =
BaggingClassifier(base_estimator=tree.DecisionTreeClassifier(),n_estimators=5
0)
scores = cross_val_score(model, X, y, cv=nFold)
print("Do chinh xac cua mo hinh voi nghi thuc kiem tra {:d}-fold: {:0.2f}
(+/- {:0.2f})".format(nFold, scores.mean(), scores.std()))
```

2. Ví dụ minh họa sử dụng giải thuật Boosting của cây quyết định.

Trong ví dụ này, học viên làm quen với:

- Sử dụng giải thuật Boosting của cây quyết định để phân loại dữ liệu.
- Đánh giá hiệu quả bằng chỉ số RMSE (Root Mean Squared Error)

```
# Nạp các gói thư viện cần thiết
import pandas as pd
from sklearn import tree
from sklearn.ensemble import AdaBoostClassifier
import numpy as np

# Đọc dữ liệu iris từ UCI (https://archive.ics.uci.edu/ml/datasets/Iris)
# hoặc từ thư viện scikit-learn
# Tham khảo https://scikit-
learn.org/stable/auto_examples/datasets/plot_iris_dataset.html
from sklearn import datasets
from sklearn.model_selection import train_test_split
iris = datasets.load_iris()
columns=["Petal Length","Petal Width","Sepal Length","Sepal Width"];
X = pd.DataFrame(iris.data, columns=columns)
```

```
y = iris.target
print(X.describe())

# Sử dụng nghi thức kiểm tra hold-out
# Chia dữ liệu ngẫu nhiên thành 2 tập dữ liệu con:
# training set và test set theo tỷ lệ 70/30
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)

# Xây dựng boosting của 50 cây quyết định, cây có độ sâu tối đa là 3
model =
AdaBoostClassifier(base_estimator=tree.DecisionTreeClassifier(max_depth=1),n_
estimators=50)
model.fit(X_train, y_train)

# Dự đoán nhãn tập kiểm tra
y_pred = model.predict(X_test)

# Đánh giá mô hình bằng chỉ số RMSE (Root Mean Squared Error)
# Tham khảo: https://en.wikipedia.org/wiki/Root-mean-square_deviation

from sklearn import metrics
print('Gia tri Mean Absolute Error: %.3f' %
metrics.mean_absolute_error(y_test, y_pred))
print('Gia tri Mean Squared Error: %.3f' % metrics.mean_squared_error(y_test,
y_pred))
print('Gia tri Root Mean Squared Error: %.3f' %
np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

3. Ví dụ minh họa sử dụng giải thuật Rừng ngẫu nhiên (Random Forest).

Trong ví dụ này, học viên làm quen với:

- Sử dụng giải thuật Rừng ngẫu nhiên (Random Forest) để phân loại dữ liệu.
- Tìm các thuộc tính quan trọng

```
# Nạp các gói thư viện cần thiết
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
import numpy as np

# Đọc dữ liệu iris từ UCI (https://archive.ics.uci.edu/ml/datasets/Iris)
# hoặc từ thư viện scikit-learn
# Tham khảo https://scikit-
learn.org/stable/auto_examples/datasets/plot_iris_dataset.html
from sklearn import datasets
from sklearn.model_selection import train_test_split
iris = datasets.load_iris()
columns=["Petal length","Petal Width","Sepal Length","Sepal Width"];
df = pd.DataFrame(iris.data, columns=columns)
y = iris.target
print(df.describe())

# Sử dụng nghi thức kiểm tra hold-out
```

```
# Chia dữ liệu ngẫu nhiên thành 2 tập dữ liệu con:
# training set và test set theo tỷ lệ 70/30
X_train, X_test, y_train, y_test = train_test_split(df, y, test_size=0.3)
#print(X_train.shape, y_train.shape)
#print(X_test.shape, y_test.shape)
#print(X_train.head())

# Xây dựng mô hình rừng ngẫu nhiên với 100 cây quyết định
model = RandomForestClassifier(n_estimators=100)
model.fit(X_train, y_train)

# Dự đoán nhãn tập kiểm tra
y_pred = model.predict(X_test)
#print(prediction)

# Tính độ chính xác
from sklearn import metrics
print("Do chinh xác cua mo hinh voi nghi thuc kiem tra hold-out: %.3f" %
metrics.accuracy_score(y_test, y_pred))

# Tìm các thuộc tính quan trọng
feature_importances =
pd.Series(model.feature_importances_,index=iris.feature_names).sort_values(as
cending=False)

# Hiển thị tầm quan trọng của các thuộc tính

import matplotlib.pyplot as plt
import seaborn as sns

# Vẽ biểu đồ
sns.barplot(x=feature_importances, y=feature_importances.index)

# Gán nhãn trục tung và trục hoành
plt.xlabel('Feature Importance Score')
plt.ylabel('Features')
plt.title("Visualizing Important Features")
plt.legend()
plt.show()
```

4. Hãy viết chương trình phân loại các tập dữ liệu sau với giải thuật Bagging của cây quyết định: Breast Cancer Wisconsin, Wine, Optical recognition of handwritten digits dataset. Khi chạy cần tuân theo nghi thức kiểm tra chéo k-fold, chỉ cần xây dựng 50 cây quyết định ở mỗi lần học. Ghi nhận kết quả.

5. Hãy viết chương trình phân loại các tập dữ liệu sau với giải thuật Boosting của cây quyết định: Breast Cancer Wisconsin, Wine, Optical recognition of handwritten digits dataset. Khi chạy cần tuân theo nghi thức kiểm tra chéo k-fold, chỉ cần xây dựng 50 cây quyết định ở mỗi lần học, có thể thay đổi tham số độ sâu của cây. Ghi nhận kết quả.

6. Hãy viết chương trình phân loại các tập dữ liệu sau với giải thuật Rừng ngẫu nhiên (Random Forest): Breast Cancer Wisconsin, Wine, Optical recognition of handwritten digits dataset. Vẽ biểu đồ tầm quan trọng của các thuộc tính trong các tập dữ liệu này.