Các giải thuật hồi quy

1. Hồi quy tuyến tính

```
from sklearn import linear_model
import pandas as pd
import numpy as np
from pandas import ExcelWriter
from pandas import ExcelFile
import math
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
from sklearn.metrics import r2_score
import seaborn as sns
sns.set_style('whitegrid')
% matplotlib inline

ILPD = pd.read_excel('data.xlsx', sheet_name='Sheet1')
ILPD.head()

#X = ILPD[:,1:11]
Y = ILPD['DIEMTS'].as_matrix()
DIEM = ILPD.drop(['HO TEN', 'DIEMTS', 'KQ'], axis=1)
X =  DIEM.as_matrix()

model = linear_model.LinearRegression()
model.fit(X, Y)
print(pd.DataFrame({"Diem: ":DIEM.columns, "Coefficients":model.coef_}).sor
t_values(by='Coefficients') )
# Sai số
print(model.intercept_)
y_pred=model.predict(X)
trainScore = math.sqrt(mean_squared_error(y_pred, Y))
print(trainScore)
print(r2_score(Y, y_pred))
x_ax=range(886)
plt.scatter(x_ax, Y, s=5, color="blue", label="Điểm thực")
plt.plot(x_ax, y_pred, lw=1.5, color="red", label="Điểm dự đoán")
plt.legend()
plt.show()

plt.scatter(DIEM['TOAN'].values, Y, color = 'blue')
plt.plot(DIEM['TOAN'].values, y_pred, color = 'red')
plt.title('Hồi quy tuyến tính')
plt.xlabel('Điểm môn Toán')
plt.ylabel('Điểm tuyển sinh')
plt.show()

plt.scatter(DIEM['ANH'].values, Y, color = 'blue')
plt.plot(DIEM['ANH'].values, y_pred, color = 'red')
```

```
    plt.title('Hồi quy tuyến tính')
    plt.xlabel('Điểm môn Anh')
    plt.ylabel('Điểm tuyển sinh')
    plt.show()

    plt.scatter(DIEM['VAN'].values, Y, color = 'blue')
    plt.plot(DIEM['VAN'].values, y_pred, color = 'red')
    plt.title('Hồi quy tuyến tính')
    plt.xlabel('Điểm môn Văn')
    plt.ylabel('Điểm tuyển sinh')
    plt.show()
```

2. Hồi quy SVM

```
    import random
    import math
    import numpy as np
    import matplotlib.pyplot as plt
    from sklearn.svm import SVR
    from sklearn.metrics import mean_squared_error
    from sklearn.metrics import r2_score
    from pandas import ExcelWriter
    from pandas import ExcelFile
    import pandas as pd
    import seaborn as sns
    sns.set_style('whitegrid')
    % matplotlib inline
    from sklearn.model_selection import train_test_split

    ILPD = pd.read_excel('data.xlsx', sheet_name='Sheet1')
    Y = ILPD['DIEMTS'].as_matrix()
    DIEM = ILPD.drop(['HO TEN', 'DIEMTS', 'KQ'], axis=1)
    X =  DIEM.as_matrix()
    x_training, x_test, y_training, y_test = train_test_split(X,Y,test_size=0.1
0, random_state=0,shuffle=True)
    cache__size = len(x_training)
    model=SVR(C=1.0, cache_size=cache__size, coef0=0.0, degree=3, epsilon=0.9,
gamma='auto', kernel='poly', max_iter=-
1, shrinking=True, tol=0.001, verbose=False)
    model.fit(x_training,y_training)

    #Huấn luyện
    pred_y = model.predict(x_training)

    x_ax=range(len(x_training))
    plt.scatter(x_ax, y_training, s=5, color="blue", label="Điểm thực")
    plt.plot(x_ax, pred_y, lw=1.5, color="red", label="Điểm dự đoán")
    plt.legend()
    plt.show()
    mse = mean_squared_error(y_training, pred_y)
    print("Mean Squared Error:",mse)
    rmse = math.sqrt(mse)
```

```
print("Root Mean Squared Error:", rmse)
print(r2_score(y_training, pred_y))

#Dự đoán
pred_y_test = model.predict(x_test)

x_ax=range(len(x_test))
plt.scatter(x_ax, y_test, s=5, color="blue", label="Điểm thực")
plt.plot(x_ax, pred_y_test, lw=1.5, color="red", label="Điểm dự đoán")
plt.legend()
plt.show()
mse = mean_squared_error(y_test, pred_y_test)
print("Mean Squared Error:",mse)
rmse = math.sqrt(mse)
print("Root Mean Squared Error:", rmse)
print(r2_score(y_test, pred_y_test))
from sklearn.metrics import mean_absolute_error
print(mean_absolute_error(y_test, pred_y_test))

# Dự đoán kết quả
ILPD = pd.read_excel('dudoan2019.xlsx', sheet_name='Sheet1')
DIEM = ILPD.drop(['HO TEN'], axis=1)
X_new =  DIEM.as_matrix()
DIEM_DD = model.predict(X_new)
print(DIEM_DD)
```

3. Hồi quy Đa thức

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('whitegrid')
% matplotlib inline
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from pandas import ExcelWriter
from pandas import ExcelFile
import math
from sklearn.metrics import mean_squared_error

ILPD = pd.read_excel('data.xlsx', sheet_name='Sheet1')
Y = ILPD['DIEMTS'].values
DIEM = ILPD.drop(['HO TEN', 'DIEMTS', 'KQ'], axis=1)
X =  DIEM.values

from sklearn.preprocessing import PolynomialFeatures
poly_reg = PolynomialFeatures(degree=2)
X_poly = poly_reg.fit_transform(X)

lin_reg_2 = LinearRegression()
lin_reg_2.fit(X_poly, Y)
```

```
print(len(lin_reg_2.coef_))

y_pred = lin_reg_2.predict(X_poly)
trainScore = math.sqrt(mean_squared_error(y_pred, Y))
x_ax=range(886)
plt.scatter(x_ax, Y, s=5, color="blue", label="Điểm thực")
plt.plot(x_ax, y_pred, lw=1.5, color="red", label="Điểm dự đoán")
plt.legend()
plt.show()
print(trainScore)
print(r2_score(Y, y_pred))

plt.scatter(DIEM['TOAN'].values, Y, color = 'blue')
plt.plot(DIEM['TOAN'].values, y_pred, color = 'red')
plt.title('Hồi quy tuyến tính')
plt.xlabel('Điểm môn Toán')
plt.ylabel('Điểm tuyển sinh')
plt.show()

plt.scatter(DIEM['ANH'].values, Y, color = 'blue')
plt.plot(DIEM['ANH'].values, y_pred, color = 'red')
plt.title('Hồi quy tuyến tính')
plt.xlabel('Điểm môn Anh')
plt.ylabel('Điểm tuyển sinh')
plt.show()

plt.scatter(DIEM['VAN'].values, Y, color = 'blue')
plt.plot(DIEM['VAN'].values, y_pred, color = 'red')
plt.title('Hồi quy tuyến tính')
plt.xlabel('Điểm môn Văn')
plt.ylabel('Điểm tuyển sinh')
plt.show()
```

4. Cây quyết định hồi quy

```
import random
import math
import numpy as np
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from pandas import ExcelWriter
from pandas import ExcelFile
import pandas as pd
import seaborn as sns
sns.set_style('whitegrid')
% matplotlib inline
from sklearn.model_selection import train_test_split

ILPD = pd.read_excel('data.xlsx', sheet_name='Sheet1')
Y = ILPD['DIEMTS'].as_matrix()
```

```
DIEM = ILPD.drop(['HO TEN', 'DIEMTS', 'KQ'], axis=1)
X =  DIEM.as_matrix()
x_training, x_test, y_training, y_test = train_test_split(X,Y,test_size=0.1
0, random_state=0,shuffle=True)

model = DecisionTreeRegressor(random_state = 0)
model.fit(x_training,y_training)

#Huấn luyện
pred_y = model.predict(x_training)
x_ax=range(len(x_training))
plt.scatter(x_ax, y_training, s=5, color="blue", label="Điểm thực")
plt.plot(x_ax, pred_y, lw=1.5, color="red", label="Điểm dự đoán")
plt.legend()
plt.show()
mse = mean_squared_error(y_training, pred_y)
print("Mean Squared Error:",mse)
rmse = math.sqrt(mse)
print("Root Mean Squared Error:", rmse)
print(r2_score(y_training, pred_y))

#Huấn luyện
pred_y_test = model.predict(x_test)
x_ax=range(len(x_test))
plt.scatter(x_ax, y_test, s=5, color="blue", label="Điểm thực")
plt.plot(x_ax, pred_y_test, lw=1.5, color="red", label="Điểm dự đoán")
plt.legend()
plt.show()
mse = mean_squared_error(y_test, pred_y_test)
print("Mean Squared Error:",mse)
rmse = math.sqrt(mse)
print("Root Mean Squared Error:", rmse)
print(r2_score(y_test, pred_y_test))
```

5. Hồi quy Rừng ngẫu nhiên

```
import random
import math
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from pandas import ExcelWriter
from pandas import ExcelFile
import pandas as pd
import seaborn as sns
sns.set_style('whitegrid')
% matplotlib inline
from sklearn.model_selection import train_test_split
```

```
    ILPD = pd.read_excel('data.xlsx', sheet_name='Sheet1')
    Y = ILPD['DIEMTS'].as_matrix()
    DIEM = ILPD.drop(['HO TEN', 'DIEMTS', 'KQ'], axis=1)
    X =  DIEM.as_matrix()
    x_training, x_test, y_training, y_test = train_test_split(X,Y,test_size=0.1
0, random_state=0,shuffle=True)
    print(len(x_training))

    model = RandomForestRegressor(n_estimators=20, random_state=0)

    model.fit(x_training,y_training)
    #huấn luyện
    pred_y = model.predict(x_training)

    x_ax=range(len(x_training))
    plt.scatter(x_ax, y_training, s=5, color="blue", label="Điểm thực")
    plt.plot(x_ax, pred_y, lw=1.5, color="red", label="Điểm dự đoán")
    plt.legend()
    plt.show()
    mse = mean_squared_error(y_training, pred_y)
    print("Mean Squared Error:",mse)
    rmse = math.sqrt(mse)
    print("Root Mean Squared Error:", rmse)
    print(r2_score(y_training, pred_y))

    # Dự đoán
    pred_y_test = model.predict(x_test)
    x_ax=range(len(x_test))
    plt.scatter(x_ax, y_test, s=5, color="blue", label="Điểm thực")
    plt.plot(x_ax, pred_y_test, lw=1.5, color="red", label="Điểm dự đoán")
    plt.legend()
    plt.show()
    mse = mean_squared_error(y_test, pred_y_test)
    print("Mean Squared Error:",mse)
    rmse = math.sqrt(mse)
    print("Root Mean Squared Error:", rmse)
    print(r2_score(y_test, pred_y_test))
```