**The laboratory work (Linear regression)**

In this laboratory work you will learn how to use the linear regression for arrays of data, and then you will see how to use the scikit-learn python library to simplify the calculation.

|  |
| --- |
| The gradient descent |
| **import** numpy **as** np **import** pandas **as** pd **import** seaborn **as** sns **import** plotly **import** plotly.plotly **as** py **import** matplotlib.pyplot **as** plt **from** matplotlib **import** style  df = pd.read\_csv(**"E:\Study materials\Educational process\Regression\housingdata.csv"**) housing\_colnames = [**'CRIM'**, **'ZN'**, **'INDUS'**, **'CHAS'**, **'NOX'**, **'RM'**, **'AGE'**, **'DIS'**, **'RAD'**, **'TAX'**, **'PTRATIO'**, **'B'**, **'LSTAT'**, **'MEDV'**] df.columns = housing\_colnames  **def** predictPrice(x,theta):  **return** np.dot(x,theta)  **def** abline(x, theta, Y):  *"""Plot a line from slope and intercept"""* y\_vals = predictPrice(x, theta)  plt.xlim(0, 20)  plt.ylim(-10, 60)  plt.xlabel(**'No. of Rooms in the house'**)  plt.ylabel(**'Price of house'**)  plt.gca().set\_aspect(0.1, adjustable=**'datalim'**)  plt.plot(x, Y, **'.'**, x, y\_vals, **'-'**)  plt.show()  **def** gradientDescentLinearRegression(alpha=0.045, iter=3000):  theta0 = 0  theta1 = 0  predictor = df[**"RM"**]  x = np.column\_stack((np.ones(len(predictor)), predictor))  Y = df[**"MEDV"**]  theta = np.zeros(2)  **for** i **in** range(iter):  pred = predictPrice(x, theta)  t0 = theta[0] - alpha \* (pred - Y).mean()  t1 = theta[1] - alpha \* ((pred - Y) \* x[:, 1]).mean()  theta = np.array([t0, t1])   abline(x, theta, Y)  gradientDescentLinearRegression() |

Scikit-learn library

|  |
| --- |
| **import** numpy **as** np **import** matplotlib.pyplot **as** plot **import** pandas **as** pd **from** sklearn.model\_selection **import** train\_test\_split **from** sklearn.linear\_model **import** LinearRegression  df = pd.read\_csv(**"E:\Study materials\Educational process\Regression\housingdata.csv"**) housing\_colnames = [**'CRIM'**, **'ZN'**, **'INDUS'**, **'CHAS'**, **'NOX'**, **'RM'**, **'AGE'**, **'DIS'**, **'RAD'**, **'TAX'**, **'PTRATIO'**, **'B'**, **'LSTAT'**, **'MEDV'**] df.columns = housing\_colnames  x = df[**"RM"**] y = df[**"MEDV"**] x = x.values.reshape(-1,1) y = y.values.reshape(-1,1) x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.3, random\_state = 0)  regr = LinearRegression() regr.fit(x\_train, y\_train) y\_predictor = regr.predict(x\_test) plot.xlabel(**'No. of Rooms in the house'**) plot.ylabel(**'Price of house'**) plot.scatter(x\_test, y\_test, color = **'red'**) plot.plot(x\_train, regr.predict(x\_train), color = **'blue'**) plot.show() |