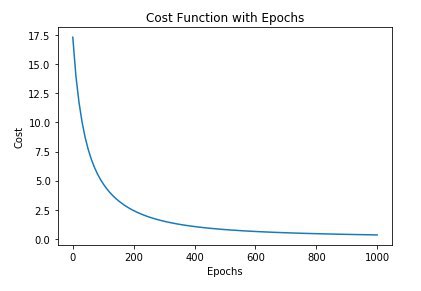
# Linear Regression

Implement a gradient descent algorithm for a linear regression problem. Use [RMSE](https://en.wikipedia.org/wiki/Root-mean-square_deviation) as error function. Test your solution at the [prices dataset](https://drive.google.com/open?id=1uyFYsVKFa7X5qYbQwxP8isAytnqTO4yc). Plot the dependence of the error function on the number of iterations of the gradient descent. Compare your result with [implementation](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html) of linear regression from the [*scikit-learn*](https://scikit-learn.org/stable/index.html) library. Use [MinMaxScaler](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html) or [StandardScaler](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html) pre-processing to make the data more consistent but don’t change target value. Repeat your experiment 3 times:

1. With no pre-processed data.
2. With min-max scaled data.
3. With standardized scaled data.



# Support Vector Machine

Study the [SVM classifier](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html) class from the [*scikit-learn*](https://scikit-learn.org/stable/index.html) library. Select multiple kernels for it. For each selected kernel try to find the best parameters and plot how your classifier classify the whole space. You can use matplotlib [pyplot.contour](https://matplotlib.org/3.1.0/api/_as_gen/matplotlib.pyplot.contour.html) to do it.

