Machine Learning

Linear Regression

Tasks:
- loading data +
- visualization +
- cost function +
- gradient descent implementation (from scratch) +
- testing +

- Solve linear regression by Normal equation +
- Instead of linear function use polynomial which over performed linear hypothesis accuracy +

Dataset. In this programming homework, we will use a Turbo.az dataset which is a collection of cars (Mercedes C-class). It contains information about 1328 cars with features of (Sheher, Marka, Model, Buraxilish ili,Ban novu,Reng,Muherrikin hecmi,Muherrikin gucu,Yanacaq novu,Yurush,Suretler qutusu,Oturucu,Yeni, Qiymet, Extra Information, Seller's comment).

Homework is divided in several parts:

Part 1 - Linear Regression from scratch - 80% of homework

80% is divided into{

10% - loading data

10% - visualization

20% - cost function

40% - gradient descent implementation (from scratch)

20% - testing}

Part 2 - Linear Regression using library - 20% of homework.

You are free to use any programming language, although Python is highly recommended. Also, consider using Jupyter Notebook for your own convenience.

Description.

Loading data. You will need to read the data from data file (turboaz.csv) and extract only 3 columns for your model - Yurush, Buraxilish ili and Qiymet. (Using pandas library is recommended)

Visualization part 1.

You will need to provide 3 visualizations of data.

1st - Yurush vs Qiymet;

2nd - Buraxilish ili vs Qiymet;

3rd - 3d plot of all three values (Yurush, Buraxilish ili and Qiymet) (Using matplotlib library (scatter, Axes3D) is recommended.)

Cost function. Implement a function which returns cost given true y values, x values and coefficients (theta).

Squared error cost function: $\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$

(□

Gradient descent from scratch. Implement a function which finds best fit parameters for linear regression using gradient descent from scratch. You should be able to change training step and number of iterations through the input to the function. In addition to that, you should collect cost at each iteration and plot it in the next sub-task. (It is recommended to normalize your data before using the gradient descent for best results).

Learning rate: alpha=0.001, you can change it in different experiments

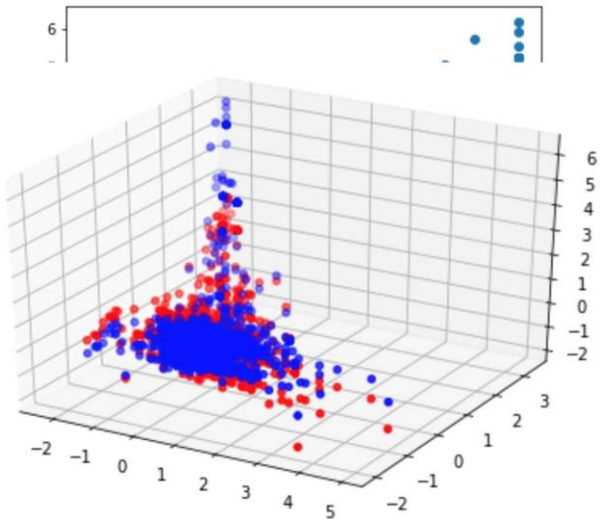
Number of iteration: 10000 or you can stop it when two sequential values are too close.

Visualization part 2.

You should plot:

1. Array of costs at each iteration (Cost vs Iteration).

2. Plot points of Buraxilish ili vs Qiymet and draw a line of predictions made with parameters you got from gradient descent. It should look similar to this:



3. Do the same for Yurush vs Qiymet.

4. Plot 3d graph of points of Yurush, Buraxilish ili, Qiymet and predicted Qiymet using the same Yurush and Buraxilish ili. It should look similar to this (blue points are true values, red points are predicted values):

Testing.

Here are given two new cars which are not in the dataset:

Car 1 { Yurush: 240000, Buraxilish ili: 2000, Qiymet: 11500}

Car 2 { Yurush: 415558, Buraxilish ili: 1996, Qiymet: 8800}

Predict Qiymet for these cars by using your parameters and see how close are your predictions of Qiymet and actual Qiymet.

Linear Regression using library (20% of grade). Use a library to fit Linear Regression to the data. You should use the same features (Yurush and Buraxilish ili) as input to this model and perform the same testing as above. (Using scikit-learn library is recommended)

Extra tasks:

1. Solve linear regression by Normal equation (20 %)
2. Instead of linear function use polynomial which over performed linear hypothesis accuracy (20%)

Note. Please be informed that your assignment on coding will be checked through Safe Assign in Blackboard. In case of two same (similar) assignment submissions, both students will be scored as a zero. Furthermore, any student whose solution may arise a question or, will be asked for some explanations as well.