MSc in Data Science

Machine Learning

Academic Year: 2020-2021

Exercise 1: Regression and Classification

Delivery Date: **13/12/2020**

You are provided with a dataset, about rental bikes between years 2011 and 2012. The identity of the dataset can be found in the following link:

- Bike Sharing in Washington D.C. Dataset: https://www.kaggle.com/marklvl/bike-sharing-dataset/home
- The dataset can be found in the following link: https://www.kaggle.com/marklvl/bike-sharing-dataset

The dataset is available also in the GitHub repository of the course, at:

https://github.com/MSc-in-Data-Science/class_material/tree/master/semester_1/Machine_Learning/datasets/bike-sharing-dataset

For this exercise, **we will use only the dataset contained in file "day.csv":** https://github.com/MSc-in-Data-Science/class_material/blob/master/semester_1/Machine_Learning/datasets/bike-sharing-dataset/day.csv

Using this dataset, you are requested to learn a set of models according to the following restrictions:

a) **Classification**
Treating the problem as a classification problem[1], use a decision tree to learn a classification model that predicts the number of persons that used a bicycle (column "cnt") based on the available features. Ensuring that overfitting has not occurred, use the learned model to identify the two most prominent features.

b) **Linear regression**
Treating the learning task as a regression problem, develop a linear regression object that predicts the number of persons that used a bicycle from all the available **numeric** features. Perform the experiment 3 times, each time with a different learning rate a, and plot the loss with respect to the training epochs required for the model to converge[2]. Which value of a has been more suitable and why? For one of the two prominent features selected in step a), and by using only a single instance from the training set, plot the loss with respect to $(y - \hat{y})$[3]. Finally, describe your processing workflow for modelling the data.

---

[1] You can also convert the values of the "`cnt`" feature into a set of categorical values (i.e. 10 categories), with `pandas.cut()`.

[2] If you use scikit-learn, you can get the loss in each epoch using a similar approach to the one shown in: https://datascience.stackexchange.com/questions/28411/how-to-plot-cost-versus-number-of-iterations-in-scikit-learn

[3] You can use `sklearn.linear_model.SGDRegressor()` with `max_iter=1`, and its `partial_fit()` method, to simulate a single step of gradient descent.

c) **Logistic regression**

Treating the problem as a **binary classification problem**[4], apply logistic regression that predicts the if "few" or "many" persons have used a bicycle from all the available **numeric** features. For one of the two prominent features selected in step a), plot the loss with respect to $(y - \hat{y})$.

In case you want to implement gradient descent from scratch in python, the following resources (among others that you can find online) may help you towards this direction:

- Gradient descent with Python: https://www.pyimagesearch.com/2016/10/10/gradient-descent-with-python/
- Gradient Descent Example for Linear Regression: https://github.com/mattnedrich/GradientDescentExample
- Gradient Descent implemented in Python using numpy: https://gist.github.com/ajmaradiaga/118f55ef4999318d6640232a73a735bd https://gist.github.com/ajmaradiaga
- (Batch) gradient descent algorithm: http://www.bogotobogo.com/python/python_numpy_batch_gradient_descent_algorithm.php
- Python Tutorial on Linear Regression with Batch Gradient Descent: http://ozzieliu.com/2016/02/09/gradient-descent-tutorial/
- How to Implement Linear Regression with Stochastic Gradient Descent from Scratch with Python: https://machinelearningmastery.com/implement-linear-regression-stochastic-gradient-descent-scratch-python/

[4] You can also convert the values of the "`cnt`" feature into a set of categorical values (i.e. 2 categories), with `pandas.cut()`.