

## ExamUppgift#2

Individual exam.

Publish : 2024-02-22

Due date : 2023-03-09 kl 23:59

### About the exam:

We need to create a python desktop app that makes our work to find the best **regressor** or best **classifier** for the data we get from our customers as **simple** and **informative** as possible.

The project consists of these parts:

- 1- Read in from the user what type of supervised ML they need to find is it **regressor** or **classifier**. (3 points).
  - 2- Read in from the user the path or/and .csv file name. (3 points).
  - 3- Read in from the user dependent target (y) by reading column name after printing all columns name to the user. (3 points).
    - a. Validate at their choice is right by checking if dependent value is **Continuous** or **Categorical**. (5 points).
  - 4- then the app needs to do:
    - a. Check if the data is ready for the machine learning process or not.
    - b. **In case it is not** we need to print report to terminal at **data is not ready and exit**. (3 points).
      - i. **Or : In case it is not**: we need to **report why it is not ready** and give hints about why it is int ready like: we have missing data in these columns .... Fill in them and rerun the app.
- data has Categorical / String data and need to create dummies, do u like to create / convert them? (yes or no) .... Etc... (5 points).

- c. In case data is ready: and it is **regressor** we need to:  
(25 points).
- i. Program in python the needed to create ML model of each **regressor** we learned:
  - ii. LiR,
  - iii. Lasso,
  - iv. Ridge,
  - v. Elastic Net,
  - vi. SVR,
  - vii. And print a report per each **regressor**.
    1. best parameter after applying GridSearchCV, CV =10 if needed.
    2. MAE,
    3. RMSE,
    4. R2 Score.
    5. Print to terminal your feedback/conclusion: which is the best for this data!
- d. In case data is ready: and it is **classifier** we need to find:  
(45 points).
- i. Program in python the needed to create ML model of each **classifier** we learned.
  - ii. LoR,
  - iii. KNN,
  - iv. SVC,
  - v. And print a report for each **Classifier**.
    1. best parameter after applying GridSearchCV, CV =10 if needed.
    2. Plot the confusion matrix per each.
    3. Print classification report per each.
    4. Print to terminal Your feedback/conclusion: which is the best for this data!

- e. At the end we need to ask the user if they agree with your feedback about the best ML model. and if yes what the name of the model to dump it out. (8 points).

Points and betyg

Ttl point 100

Point >70 => G

Points > 85 => VG