Software Analytics - LTAT.05.008

Homework HW6: evaluation and interpretation

Submission deadline: Homework solutions must be submitted within seven days, i.e., not later than on the following Monday, at 23:59 hours.

Late submission policy:

- 50% of the total marks deducted for submission up to 24 hours late
- 100% of the total marks deducted for submission more than 24 hours late

Maximum amount of points is ten (10).

Knowledge Discovery

During three lab sessions, you will walk through the standard data mining workflow and learn how to apply it to real problems in the context of software analytics.

We have chosen four real problems that are still interesting for the Software Engineering research community in order to show how the knowledge discovery process can help. These problems are:

- Estimating story points from issue reports [1, 4]
- Assignment of developers to tasks
- Exploring the GitHub social network [3]

To provide a better understanding of how to solve these problems, we have broken the solutions down into three steps that correspond to three labs sessions:

- KD1: data selection and pre-processing
- KD2: data transformation and data mining
- KD3: evaluation and interpretation

KD3: evaluation and interpretation

During the first two labs (KD1 and KD2) you walked through some common tasks related to data extraction, transformation and data mining. At the end of the lab, it is expected you will able to evaluate and interpret common problems in the context of software analytics.

Preparation

- Background information on the necessary topics for this homework can be found in the lecture slides.
- Familiarize yourself with Python, IPython Notebook, and Pandas.
- In addition, please consider following the Python Tutorial under Additional Links.

Homework
Task 1 Evaluating the effort predictions (5 points)

The main goal of this task is to apply a prediction model to get Story Points estimates for the issues used in the Spring XD project. We can treat this problem as a classification task, in which you will try to predict the estimates (in Story Points) for each issue. To train the classifier, you have used features extracted from JIRA. In this homework, you will evaluate and interpret the results of your predictions.

Task 1.1 (0.5 points)
Evaluate the accuracy of the model that you have created during HW5

1. Split the dataset about issue reports "issues-xd.csv" into train/test sets.
2. Train the classifier as you have done in HW5 using the training set.
3. Calculate the accuracy of the model on the test dataset.

In [ ]:

Task 1.2 (0.5 points)
Perform a sanity check about the predictions obtained

1. Calculate the accuracy of a model that makes random predictions.
2. Calculate the accuracy of a model that makes predictions based on the most common value.

In [ ]:

Task 1.3 (0.5 points)
Compute problem-specific metrics

1. Calculate the MAE of the prediction model created in HW5.
2. Calculate the SA of the prediction model created in HW5.

Task 1.4 (1.5 points)
Perform hyperparameter tuning

1. Choose a k-fold strategy or a holdout strategy. Using that strategy, learn the best parameters for the classification model and evaluate the model in terms of MAE and SA.
**Task 1.5 (2 points)**

1. Compare the performance in terms of accuracy, MAE, and SA of the model that uses the best parameters with the performance of your first model (without hyperparameter tuning). **Which model have shown the best performance?**

**Task 2 Evaluating the recommender system (5 points)**

In this homework, you will evaluate the performance of the recommender system you have created during HW5.

**Task 2.1 (2 points)**

1. Choose a k-fold strategy or a holdout strategy. Using that strategy, learn the best parameters for the recommender system and evaluate the model as follows:
2. Set N in order to propose a list of N items that a user is expected to find interesting (top-N recommendation).
3. Calculate the PR-curve of the model for different values of N. To build a PR-curve, you will have to:
   - Calculate the Recall of the system
     \[
     Recall = \frac{\text{#liked and shown}}{\text{#liked}}
     \]
   - Calculate the Precision of the system
     \[
     Precision = \frac{\text{#liked and shown}}{\text{#shown}}
     \]

**Task 2.2 (3 points)**

Perform a sanity check of the recommender system you have created.

1. Compare the results in terms of PR-curve of the recommender with the results of a Random recommender.
2. Compare the results in terms of PR-curve of the recommender with a recommender using k=3
3. According to your results, please answer the following question: **How can you improve the performance of your recommender system?**
Submission

The solution to this homework must be submitted in a zip file via the course web-page selecting Lab 5. The zip file must contain:

- Your name
- The code scripts for all the tasks. Ideally, you will complete the IPython Notebook provided. If you use another language, you must submit the code/scripts used. In this case, each piece of code must clearly indicate by a comment which tasks is addressed.

Additional links

- Python tutorial - [https://docs.python.org/2/tutorial/](https://docs.python.org/2/tutorial/)
- Jupyter installation guide for IPython Notebooks - [https://jupyter.org/install](https://jupyter.org/install)
- Pandas 10 min tutorial - [https://pandas.pydata.org/pandas-docs/stable/10min.html](https://pandas.pydata.org/pandas-docs/stable/10min.html)

References


Copyright (c) 2018, Ezequiel Scott All rights reserved.