Prof. Dr. Oscar Nierstrasz
Pascal Gadient, Pooja Rani

Solution
Assignment 11 — 28/11/2018 – v1.0
Software Data Analytics

Please submit this exercise by mail to sma@list.inf.unibe.ch before 05 December 2018, 10:15am.

Exercise 1: Data in SDA (6 Points)

- What is a confusion matrix? (1pt) **Answer:**

  A confusion matrix is a table as shown in Figure 1 that shows the summary of a classification experiment outcome, i.e., *the classification quality. In other words, it reveals the "class confusion" of the classification model. On the one hand, the table provides column-wise information about the actual class instances used in the experiment,* e.g., *the column "cat" shows that 8 instances of cat have been used in the experiment. On the other hand, the table provides row-wise information about the predicted elements,* e.g., *the row "cat" reveals that five instances have been correctly classified as cat, while 2 were incorrectly classified as dogs.*
  A confusion matrix further supports the straightforward calculation of precision and recall measures, as well as the F1 score. For that, one can simply use the corresponding fields of the table. For instance, the calculation of the cat classification precision could be done by adapting the formula $P = TP/(TP + FP)$ *to the values in the table which would lead to* $P = A1/(A1 + B1) = 5/(5 + 2) = 0.714$.

		actual class: cat	actual class: dog	actual class: rabbit
predicted class	cat	A1 **5**	B1 2	C1 0
	dog	A2 3	B2 **3**	C2 2
	rabbit	A3 0	B3 1	C3 **11**

Figure 1: Traditional confusion matrix regarding the classification of animals

- Which criteria are important when extracting the data? List two different criteria and elaborate on each. (2pts). **Answer:**
  - *Degree of automation. The data extraction process should be automated as much as possible to avoid repetitive mistakes, e.g., a typo in the separation indicator in a CSV file could render data useless and might not be discovered immediately when only a subset of a large dataset is affected.*
  - *Availability of historical data. It is recommended to keep any data, so that experiments can be rerun in case of implementation issues (mistake in the analysis) or for future verification (through different people).*
  - *Noise-level of the data. The level of noise in measurements should be kept minimal. Noise introduces a bias and can even invalidate the results if it is sufficiently strong.*

- Which are the three groups of software data? Elaborate on each. (3pts) **Answer:**
  - *Contextual data. The data related to a product or team. This data group primarily describes project objectives (*e.g., *frequency of releases), the project history (*e.g., *impact of prior features on customers), the organizational structure (*e.g., *experience of developers), or the clients (*e.g., *browser, desktop, mobile, ...).*
  - *Constraint data. The data related to product constraints; the goals that need to be fulfilled. This data group primarily describes non-functional requirements such as quality*[1] *(*e.g., *reliability of code), compatibility (*e.g., *supported platforms), performance (*e.g., *start-up time), or legacy use (*e.g., *percentage of assembly code).*
  - *Development data. The data related to the development process, the verification, and the code readiness. This data group primarily describes code churn (*e.g., *number of changed lines), code velocity (*e.g., *time spent on implementation of a specific feature), code complexity (*e.g., *McCabe's measure), or code dependencies (*e.g., *depth of dependency tree, use of external libraries).*

Exercise 2: Process of SDA (4 Points)

- What is a use case for software data analytics? (1pt) **Answer:**

  There exist countless use cases. The majority of them increase the value for either the users (software customers) or the creators (developers, stakeholders) of a product. The most prevalent use case for SDA in industry is presumably the improvement of a product in terms of user experience. More specifically, SDA provides a helping hand for:

  - *identifying inefficient code in order to apply some optimizations*
  - *finding code hot-spots that are heavily used throughout a software system to harden their resilience against potential attacks which will improve the overall reliability and security of the system*
  - *gathering test coverage for existing code to identify code that needs more testing to reduce the risk of application crashes due to unexpected inputs*

- What are types of software analytics problems? Choose two and explain how they incorporate a problem in software analytics. Provide a concrete example for each. (2pts) **Answer:**
  - *Bug measurements. Most bug measurements are incomplete since humans reported the occasional bugs manually. Nevertheless, the number of bugs is a valuable indicator for the maturity of a project. Therefore, finding an indicator for bugs (*e.g., *the locality of bug prone classes) would greatly help to predict locations which need special attention.*
  - *Development practices. Developer practices might be suboptimal. For instance, one could improve the developer efficiency by analyzing how developers read comments and create guidelines for their optimal use.*

[1] Please note that this quality aspect is not related to any source code metrics like cyclomatic complexity or lines of code.

- *Testing practices. It is impossible to consider every aspect of software testing, still testing is important to keep software maintainable and bug free. Hence, a way to go is to improve the existing testing practices. For instance, one could gather the amount of development time spent on unit tests of certain classes. This would reveal classes that are not sufficiently tested and allow a certain level of test coverage throughout a project.*
- *Evaluating quality. The term "quality" is important, yet not strictly defined in the context of software engineering. Thus, there exists no one-fits-it-all measure to evaluate software quality. In general, it's rather a combination of different metrics that are used for code audits (*e.g.*, LOC, cyclo, ...). Software data analytics can not only help to find reasonable metrics that deliver meaningful results, the metrics themselves can also provide feedback regarding the quality of specific classes. It would be desirable that a lack of quality automatically triggers a code review phase.*
- *Software development lifecycle. The SDL is very complex since many teams are involved (*e.g.*, development, marketing, external service providers). Hence, an optimized SDL would streamline the work for some teams and thus foster substantial savings in monetary resources. For instance, software data analytics could be used to evaluate the issues raised by the different teams with respect to the progress of the project. If found that marketing people are always waiting for product specifications from the developers, then it might be wise to integrate them into the process at a later stage for future projects.*
- *Productivity. Less-productive employees are causing major financial losses. Software data analytics could identify those people and provide valuable feedback to them on how to improve their productivity.*

- What is the recommended five-step guideline for working in software data analytics? (1pt) **Answer:**

  Software data analytics should follow a five-step process:

  1. *Problem identification. Identify the problem and translate the business-side problems into technical questions. Reason about the required data.*
  2. *Collecting data. Reason about the location, the integrity, the structure, and the security demands of the required data. Collect and store the data.*
  3. *Descriptive analysis. Use graphical visualizations or numerical indicators to gather a first glimpse into the collected data (the current state). Countless approaches exist: bar charts, scatter plots, and minimum or maximum values, etc.*
  4. *Predictive analysis. Use machine learning to predict a future state. Approximate the complexity (dimensionality) of the problem and start with simple algorithms.*
  5. *Performance evaluation. Evaluate the quality of the predicted results. Confusion matrices have proven very helpful since they allow the straightforward calculation of several different quality-related parameters such as precision, recall, or the F1 score.*