Assignment OO Programming Sem 2 2025

The aim of this assignment is to build a java application. The application will be a predictor, based on a theme that will be assigned to you. Effectively, you will be building a supervised machine learning classifier based on the naïve bayes classifier.

This assignment is worth 60% of your Sem 2 CA (and therefore, 30% of your OVERALL CA mark.

Deadline

Submit by Tuesday, 29th April 10pm.

First Step

Note the theme assigned to you under the "assignment themes". Each person will build a different theme

---

.

Assignment criteria

- All of your assignment to be written in Java.
- Your code should demonstrate the use of OO concepts where appropriate (separate classes, encapsulation, composition, inheritance , interfaces, etc).
- Code should follow java code standards
- Use BitBucket or GIT to manage your source code. Your code and video link will be collected from your whatever code repository you use (more on this). If you haven't used GIT, we can cover in class. Please make sure you commit to your code source throughout the project.
- If you make use of any code directly from another source (online, genAI, book), **you must show this in the comments.** You will be marked on code that is your own code.

What you submit:

- **Code** (either as zip file or repository link);
- **Video:** *3 minutes or less*, in which you demo all of your app. Explain how the application works, what you implemented, the data you are entering or what you are clicking, any error checking etc. as if you were demo'ing in person. Just show your screen with your voice - as opposed to you!
- **Readme file**: Include a readme file to explain your project. This should list:
    - The frequency table for your dataset (I need this to be able to verify your code, no matter how what Level of code you deliver)
    - List of java classes, with a description

Assignment OO Programming Sem 2 2025

- List of functionality you included;
- Explain what you would add if you had more time.

Machine learning model

This assignment allows you to learn about machine learning, including building a supervised predictive/classification model.

The purpose of the assignment is to build a **GUI prediction tool** that predicts an outcome, based on theme you have been given. For example, if your theme was LoanApproval - the predictor will be about predicting whether a user of the application will be approved for the loan (and the probability), based on enter their data (i.e. the value of the "features" the model is trained on).

As per the lecture on "A peek at machine learning" Week 7 , the predictor will be trained by your dataset. Your dataset contains previous "examples" that your predictor will learn a set of rules or probabilities from. This is done by calculating the various permutational frequencies as done in the lecture.

For your dataset, generate 200 rows of data as per the exercise of Lab week 8: 5 columns (4 features and 1 label), save the file down as a CSV file. The last column (the label) has values "yes" or "no" **. Suggest suitable predictive columns title and values for the first four columns. All data should be categorical, with two possible values per column. The values should be meaningful (e.g. not a,b,c,) The 200 rows should include all permutations of values across the 5 columns.

You will need to build a GUI to take the new input (feature values) that you want to predict about.

The levels of functionality is listed below. See the rubric for the level of marking per level:

Level 1 GUI and predictor functionality where the predictive rules are driven by values that you calculated for your dataset offline using a frequency table and hard coded in your code as variables. The user can enter a set of (feature) values and get a prediction.

Level 2 :GUI and predictor functionality where the predictive rules are driven by values that are calculated dynamically from the training dataset in your code. This should include a button on your GUI to train your classifier (i.e. read in your dataset and calculate your rule).

Level 3 : You can also use the GUI to enter in new rows to the dataset with feature values and known label values (i.e. expand your training dataset). This should then recalculate the various permutations totals for the rules (i.e. recalculate the classifier, based on pressing a button).

Level 4 : Your application is also able to calculate the accuracy of the predictor. This will involve training your predictor on 150 rows of the data – and testing your data on 50 rows (where each of these rows is automatically put through the rules, and the predictive output automatically matched with the actual label). Note that your training and test data should be stratified via your code (have the same proportion of the yes/nos of the label classes in each of training/ testing. So if overall dataset had 60% yes, 40% no,

Assignment OO Programming Sem 2 2025

you could accidentally train on 60% yes, and test on all the nos, which would not work. Instead you want your 60/40 balance of rows in your training, and 60/40 in your testing.

Marking scheme

The marking scheme will be:

- Up to 10% video *(and read me file)* explaining your assignment – available via YouTube or any video hosting service
- Up to 10% GUI and application quality, scaled for level delivered.
- Up to 10% code and class design quality, scaled for level delivered.
- Up to 70% Functionality delivered (working:
    - Level 1 out of 20% / Level 2 out of 15%
    - Level 3 out of 15%
    - Level 4 out of 20%

- **Note: Plagiarism means attempting to pass off someone else's work or GenAI outputs as your own or deliberately allowing another student to copy your work**
- **IF you plagiarise, your assignment, you will get zero marks and it will be recorded on your record.**