1. Introduction

1.1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Food Recommendation System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

1.2. Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

2 Dataset:

2.1 **Introduction - Data Provided:**

The data used in this report was taken from the UCI repository (Mice Protein Expression Dataset). The following description of the studied dataset is transcribed from the previous mentioned source as is:

``The data set consists of the expression levels of 77 proteins/protein modifications that produced detectable signals in the nuclear fraction of cortex. There are 38 control mice and 34 trisomic mice (Down syndrome), for a total of 72 mice. In the experiments, 15 measurements were registered of each protein per sample/mouse.

Therefore, for control mice, there are 38x15, or 570 measurements, and for trisomic mice, there are 34x15, or 510 measurements. The dataset contains a total of 1080 measurements per protein. Each measurement can be considered as an independent sample/mouse.

The eight classes of mice are described based on features such as genotype, behavior and treatment. According to genotype, mice can be control or trisomic. According to behavior, some mice have been stimulated to learn (context-shock) and others have not (shockcontext) and in order to assess the effect of the drug memantine in recovering the ability to learn in trisomic mice, some mice have been injected with the drug and others have not.

2.2 Categories in data set(dependent Feature):

c-CS-s: control mice, stimulated to learn, injected with saline

c-CS-m: control mice, stimulated to learn, injected with memantine

c-SC-s: control mice, not stimulated to learn, injected with saline

c-SC-m: control mice, not stimulated to learn, injected with memantine

t-CS-s: trisomy mice, stimulated to learn, injected with saline

t-CS-m: trisomy mice, stimulated to learn, injected with memantine

t-SC-s: trisomy mice, not stimulated to learn, injected with saline

t-SC-m: trisomy mice, not stimulated to learn, injected with memantine

3 Methodology:

Data analysis of this project consisted of 4 steps:

1) data pre-processing

2) data exploration

3) data modelling

4) test the model.

1. **Data pre-processing**

This dataset in each protein contain missing values. To deal with these, the mice will receive the mean value of the mice in their class, rather than eliminating the entire mouse from the data, or by using the mean of all of the mice which have very different genetic and environmental factors in

COSC2670 their development.

The second thing that needs to be dealt with is the high variance of measurement values. For example, some proteins have exploration value range from 0 to 1 while other proteins ranged from 0 to 8. With this condition, proteins with higher value will have more influence in this classification outcome and make the model unable to learn correctly. Therefore, after making sure no null values in the measurements, all value of measurements are standardized with zero means and unit variance.

**2. Data Exploration**

The data exploration was done on the data following the replacement of nulls with the means, but before it was normalised.

1. **Data Modelling**

**Model Background**

Decision tree and random forest were used to model and predict the mice class and predict which proteins were critical for each class. These are both examples of supervised machine learning methods with the goal of creating model that predicts the value of a target variable based on several

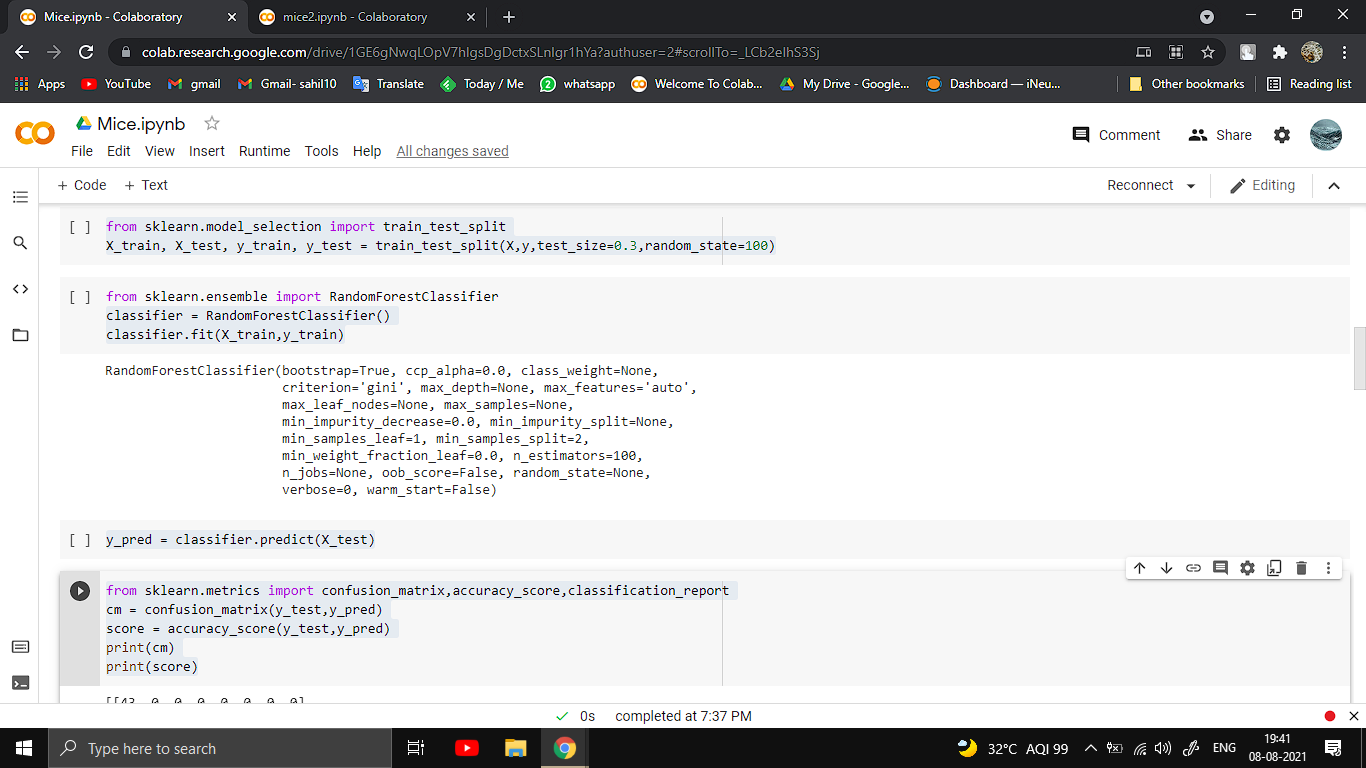
input variables.

Decision tree algorithm tries to solve the problem by using tree representation of a series of decisions. A tree can be "learned" by splitting the source set into subsets based on an attribute value test. The splitting creates two types of nodes: decision nodes, and leaves. The starting node is called

the root. If the root is a leaf then the decision tree is degenerate and the same classification is made for all data. A single variable is being examined for each decision nodes, and move to another node based on the outcome of a comparison. This is repeated until a leaf node is reached. At a leaf node

the decision is being made: whether the training data routed to the leaf node as a classification decision, or return the mean-value of outcomes as a regression estimate.

Similar with decision tree, random forest also use tree representation to solve the problem. The difference is instead of using one tree, random forest averaging multiple deep decision trees and trained on different parts of the same training set, with the goal of reducing the variance and overfittig.



4.Results

In this project, we built and used 3 models. First, decision tree model with “class” variable as a compound of three binary variables (8 in total). Second, random forest model with also “class” variable. Third, we created our own classifier that first predicted the 3 binary variables (“Genotype”,

“Treatment”, and “Behavior”), before passing these variables in as binary features to another Random Forest classifier.

