COMP3308 Assignment 1
Predicting Diabetes

Woo Hyun Jung 310250811
Khanh Cao Quoc Nguyen 311253865

1 Aim

The aim of our study is to predict whether a new patient will test positive for diabetes (class 1). The study is important because blah blah dicks

2 Data

2.1 Dataset

The data set we used was the Pima Indians Diabetes Database found at `http://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/`.
There are nine attributes:

1. Number of times pregnant
2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/(height in m)^2)
7. Diabetes pedigree function
8. Age (years)
9. Class variable (0 or 1)

and two classes:

1. class1 (testing positive for diabetes)
2. class0 (testing negative for diabetes)

2.2 Data preparation

2.2.1 Manual preprocessing

The raw `.data` file was preprocessed manually in multiple ways. Firstly we had to come up with simple names for the nine attributes:

1. num_pregnant
2. plasma_glucose_concentration
3. diastolic_blood_pressure
4. tricep_skin_fold_thickness
5. 2h_serum_insulin
6. bmi
7. diabetes_pedigree_function
8. age

9. class

Using these names, we added a header row the `.data` file. We then changed the values of the class column to class0 and class1 instead of 0 and 1. This file was then saved a `.csv` file for later convenience.

2.2.2 Missing Values

We wrote a script `csv_scripts\missing_values.py` that dealt with the missing values in the following attributes:

1. plasma_glucose_concentration
2. diastolic_blood_pressure
3. tricep_skin_fold_thickness
4. 2h_serum_insulin
5. bmi
6. diabetes_pedigree_function

We determined that these attributes consisted of missing values because 0 is a impossible or unrealistic value for it.

This script outputs two `.csv` files, one with missing values taking the calculated average of its respective attribute(`pima-indians-diabetes-avg.csv`) and the other with any instances with missing values omitted(`pima-indians-diabetes-omit.csv`).

We noticed that if we omit the instances with missing values we would be left with 392 instances out of the original 768. So we decided to use the averaged output file for the rest of the assignment.

2.2.3 Normalisation

Once this new file was loaded into Weka, each of the attributes were normalised in the range $[0, 1]$. The final output file was saved as `pima.csv`

2.3 Attribute selection

With the `pima.csv` loaded into Weka, we used the `CfsSubsetEval` attribute evaluator with the `BestFirst` search method available on the `Select attributes` tab to determine the best subset of features based on how good the individual feature are at predicting the class and how much they correlate with the other features. The selected attributes were:

1. plasma_glucose_concentration
2. 2h_serum_insulin
3. bmi
4. diabetes_pedigree_function
5. age

We then proceed to use Weka's preprocessing features to remove the attributes that were not selected and saved the file as `pima-CFS.csv`.

3 Results and Discussion

	ZeroR	1R	1-NN	5-NN	NB	DT	MLP
No feature selection	65.1042	70.8333	67.7083	74.7396	74.7396	72.2656	75.3906
Correlation-based feature selection	65.1042	70.8333	69.0104	74.6094	76.4323	73.9583	75.7813

Table 1: Results from Weka

	My1-NN	My5-NN	MyNB
No feature selection	68.7645	75.2546	75.3828
Correlation-based feature selection	68.4979	76.0509	76.6883

Table 2: Results of implementations of kNN and NB

discuss here blah blah blah bleh

4 Conclusions

5 Reflection

6 Instructions

Our implementation of K Nearest Neighbour, Naive Bayes and 10fold stratified cross validation was written in Python 2.7.5.

The `ai_main.py` file runs 10-fold stratified cross validation on K Nearest Neighbour($k = 1$ and $k = 5$) and Naive Bayes on a given `.csv` file.

To run our implementation, you must `cd` into the `ai_code` directory and ensure that `pima.csv` and `pima-CFS.csv` are present in the directory. Then run:

```
python ai_main.py pima.csv
python ai_main.py pima–CFS.csv
```

This should output `pima-folds.csv` and `pima-CFS-folds.csv` respectively, which are the stratified folds.