Margareta Kušan and Neven Pičuljan

Project 1 - first phase

Dataset and analysis

Dataset

Title of the dataset is Australian Credit Approval. This file concerns credit card applications. All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data. This dataset is interesting because there is a good mix of attributes -- continuous, nominal with small numbers of values, and nominal with larger numbers of values. There are also a few missing values. Number of Instances: 690, number of Attributes: 14 + class attribute. There are 6 numerical and 8 categorical attributes.

Description of Attributes:

- A1: 0,1 CATEGORICAL
- A2: continuous.
- A3: continuous.
- A4: 1,2,3 CATEGORICAL
- A5: 1, 2,3,4,5, 6,7,8,9,10,11,12,13,14 CATEGORICAL
- A6: 1, 2,3, 4,5,6,7,8,9 CATEGORICAL
- A7: continuous.
- A8: 1, 0 CATEGORICAL
- A9: 1, 0 CATEGORICAL
- A10: continuous.
- A11: 1, 0 CATEGORICAL
- A12: 1, 2, 3 CATEGORICAL
- A13: continuous.
- A14: continuous.
- A15: 1,2

37 cases (5%) HAD one or more missing values. The missing values from particular attributes were:

- A1: 12
- A2: 12
- A4: 6
- A5: 6
- A6: 9
- A7: 9
- A14: 13

They were replaced by the mode of the attribute (for categorical) and mean of the attribute (continuous).

Class Distribution:

- +: 307 (44.5%) CLASS 2
- -: 383 (55.5%) CLASS 1

Feature Analysis

We made data analysis using Python. These graphs show the distribution of categorical features:

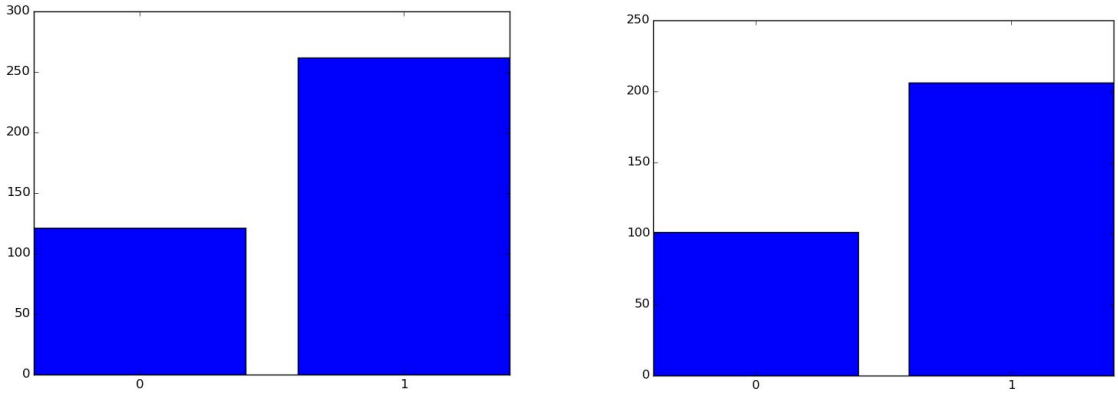


Fig 1. Feature A1 : a) distribution of A1 in class 0, b) distribution of A1 in class 1

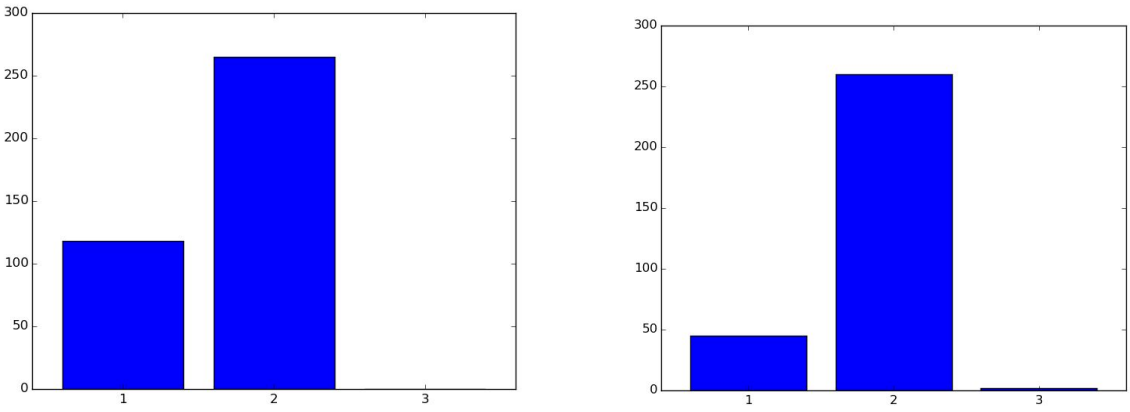


Fig 2. Feature A4 : a) distribution of A4 in class 0, b) distribution of A4 in class 1

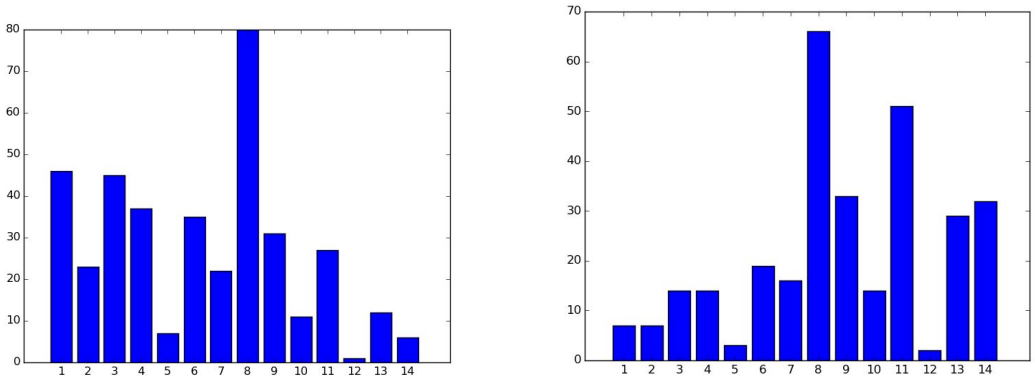


Fig 3. Feature A5 : a) distribution of A5 in class 0 b) distribution of A5 in class 1

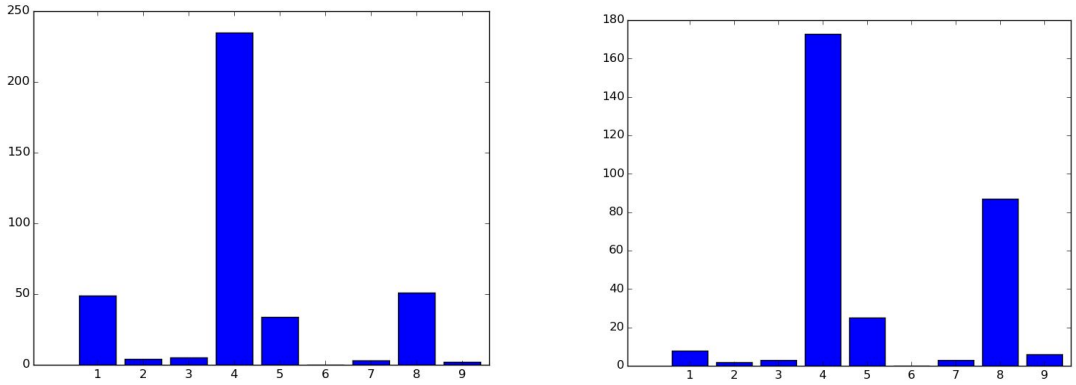


Fig 4. Feature A6 : a) distribution of A6 in class 0 b) distribution of A6 in class 1

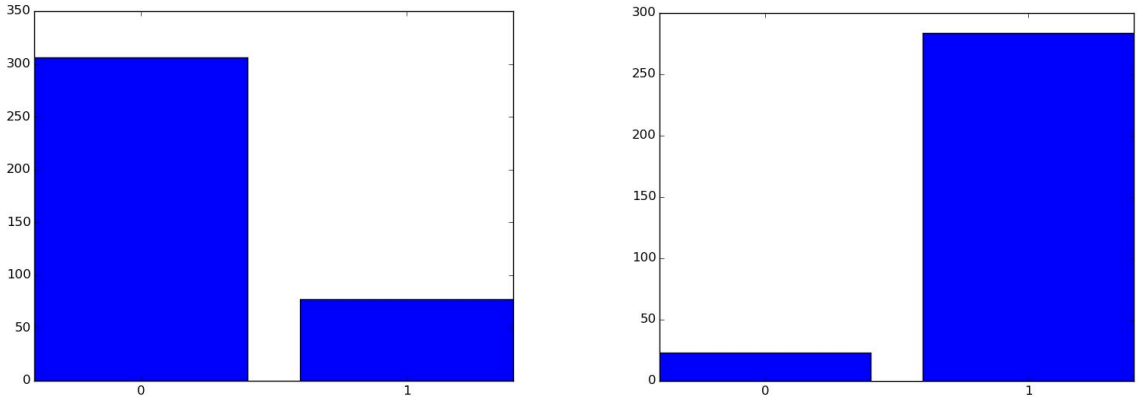


Fig 5. Feature A8 : a) distribution of A8 in class 0, b) distribution of A8 in class 1

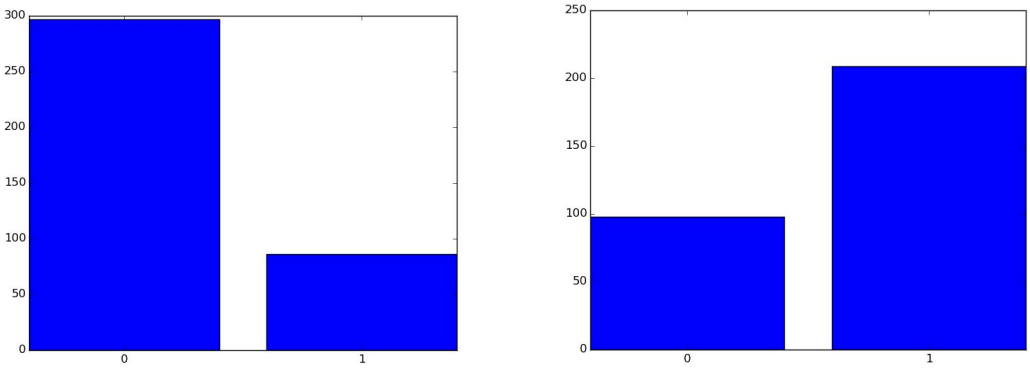


Fig 6. Feature A9 : a) distribution of A9 in class 0, b) distribution of A9 in class 1

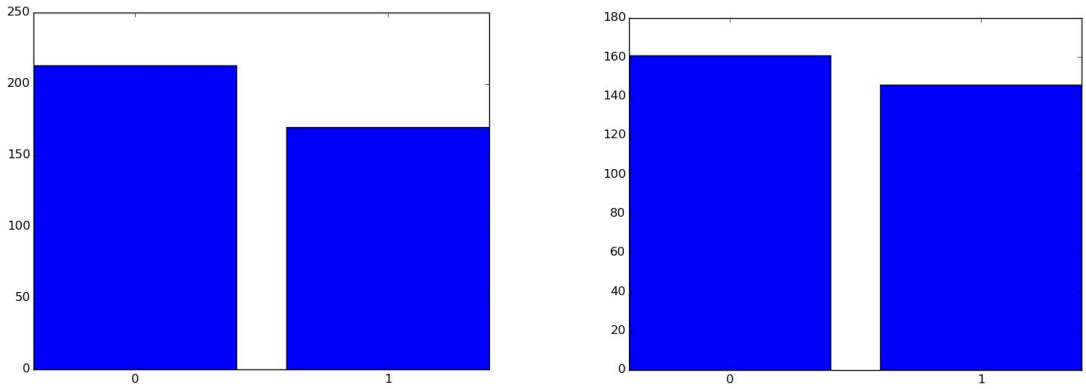


Fig 7. Feature A11 : a) distribution of A11 in class 0, b) distribution of A11 in class 1

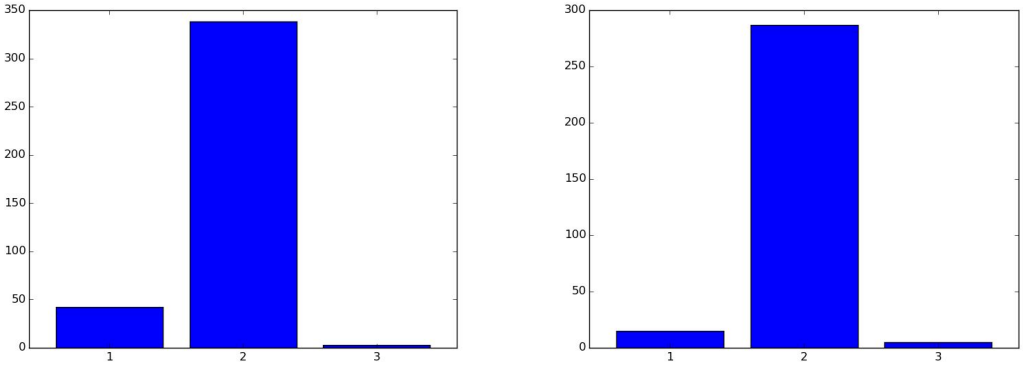


Fig 8. Feature A12 : a) distribution of A12 in class 0, b) distribution of A12 in class 1

These figures show Box and Whisker plots of continuous features:

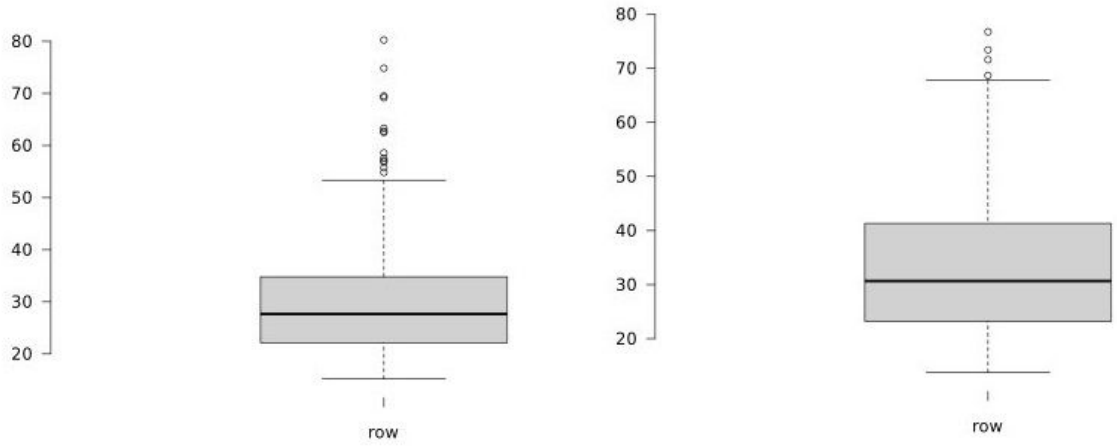


Fig 9. Feature A2: a) Box and Whisker plot for class 0 b) Box and Whisker plot for class 1

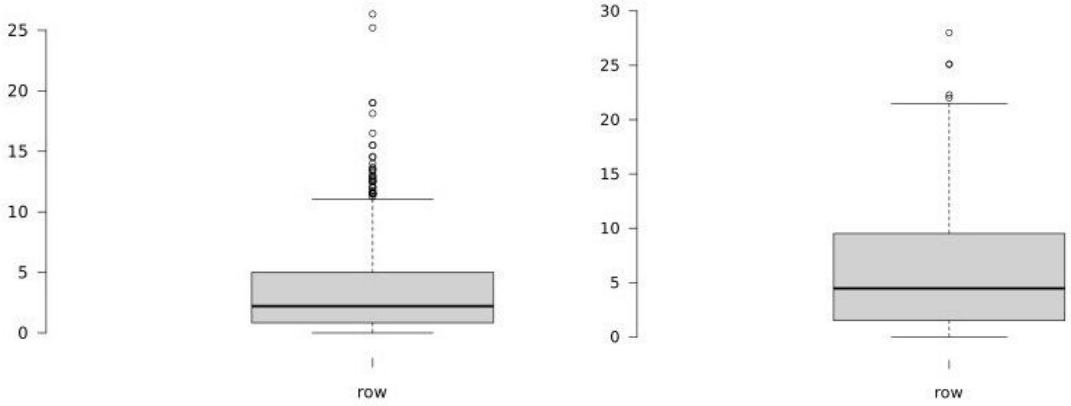


Fig 10. Feature A3: a) Box and Whisker plot for class 0 b) Box and Whisker plot for class 1

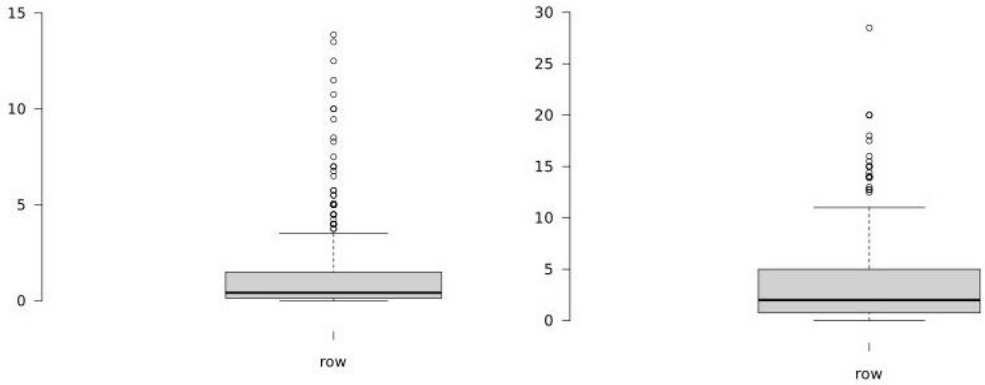


Fig 11. Feature A7: a) Box and Whisker plot for class 0 b) Box and Whisker plot for class 1

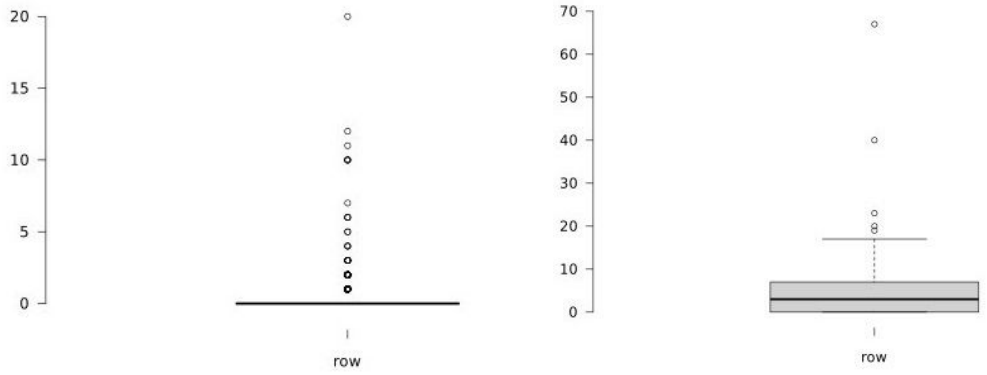


Fig 12. Feature A10: a) Box and Whisker plot for class 0 b) Box and Whisker plot for class 1

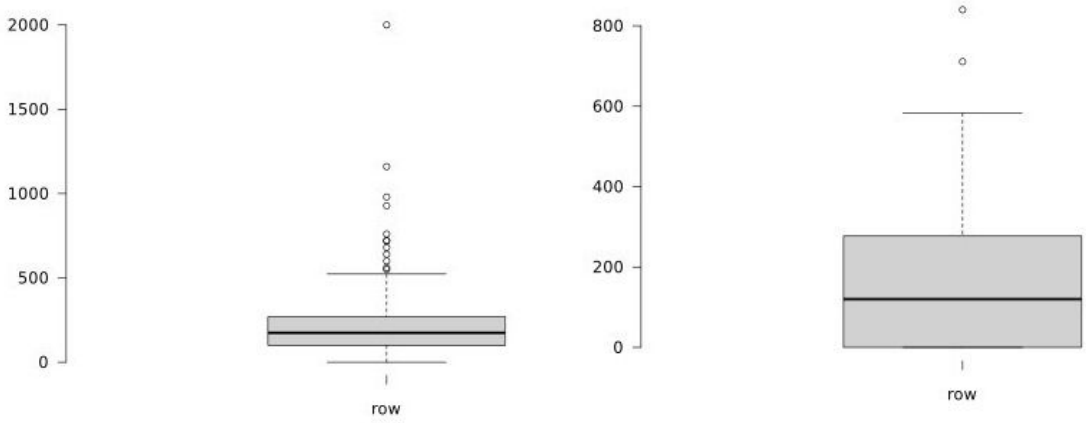


Fig 13. Feature A13: Box and Whisker plot for class 0 b) Box and Whisker plot for class 1

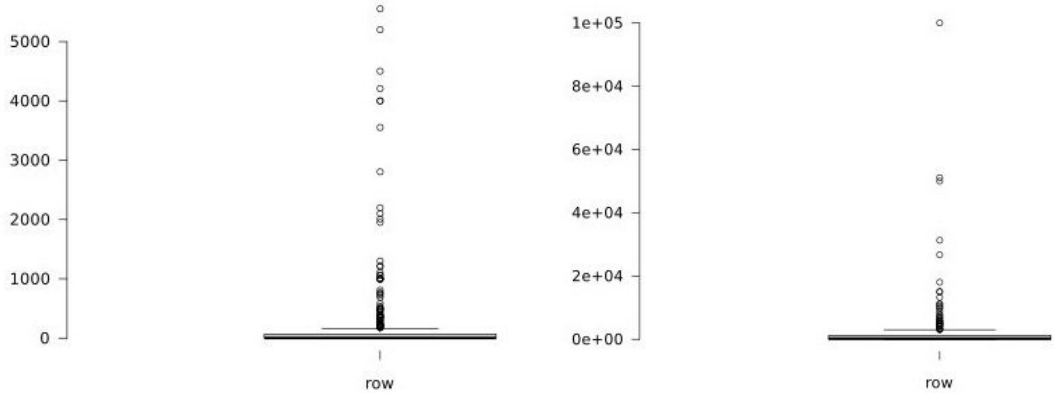


Fig 14. Feature A14: Box and Whisker plot for class 0 b) Box and Whisker plot for class 1

Conclusion of Dataset Analysis

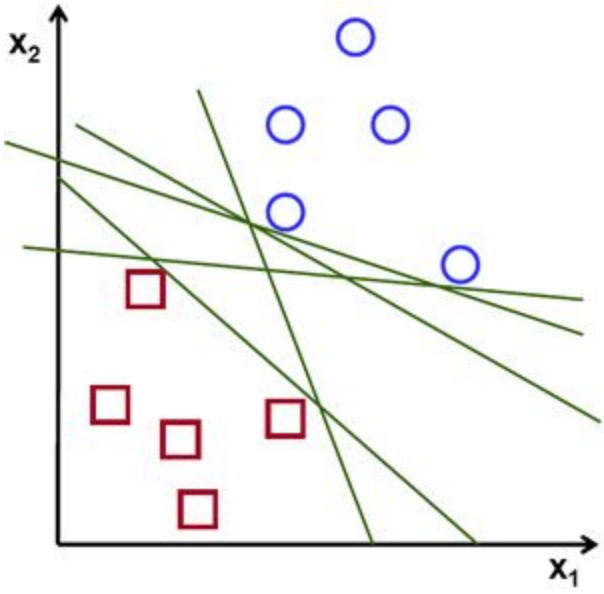
Bar charts are used to analyze categorical features and Box and Whisker plots are used to analyze continuous features. From bar charts, we can clearly see category distributions of each attribute by classes and in Box and Whisker plots data is presented through its quartiles. As much graphs for the same attribute but different classes differ, attributes are better because they are much more discriminatory.

Classificator description

1. SVM

Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. Given labeled training data (*supervised learning*), the algorithm outputs an optimal hyperplane which categorizes new examples.

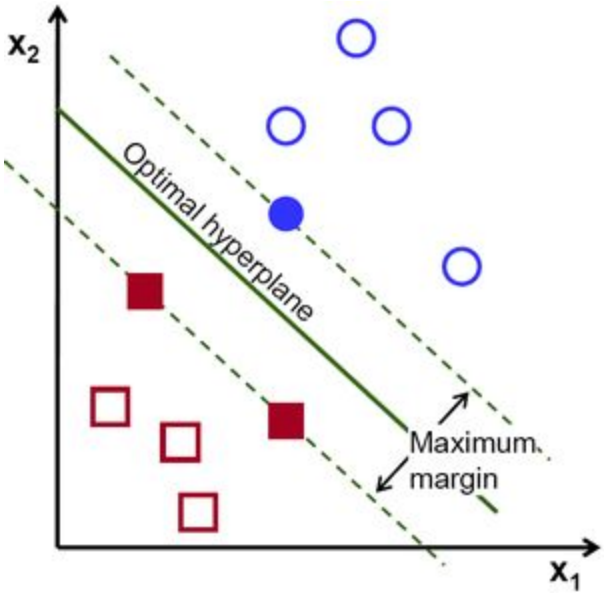
Example: For a linearly separable set of 2D-points which belong to one of two classes, find a separating straight line:



In the above picture you can see that there exists multiple lines that offer a solution to the problem. Is any of them better than the others? We can intuitively define a criterion to estimate the worth of the lines:

A line is bad if it passes too close to the points because it will be noise sensitive and it will not generalize correctly. Therefore, our goal should be to find the line passing as far as possible from all points.

Then, the operation of the SVM algorithm is based on finding the hyperplane that gives the largest minimum distance to the training examples. Twice, this distance receives the important name of **margin** within SVM's theory. Therefore, the optimal separating hyperplane *maximizes* the margin of the training data.



2. Gaussian Naïve Bayes

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes theorem with the "naive" assumption of independence between every pair of features. Given a class variable y and a dependent feature vector x_1 through x_n, Bayes theorem states the following relationship:

$$P(y \mid x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots x_n \mid y)}{P(x_1, \dots, x_n)}$$

Using the naive independence assumption that

$$P(x_i|y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i|y),$$

for all i, this relationship is simplified to

$$P(y \mid x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \dots, x_n)}$$

Since $P(x_1, \dots, x_n)$ is constant given the input, we can use the following classification rule:

$$P(y \mid x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

and we can use Maximum A Posteriori (MAP) estimation to estimate $P(y)$ and $P(x_i \mid y)$; the former is then the relative frequency of class y in the training set.

Result Analysis

Python's library scikit-learn is used. Dataset is divided into two parts: training data (70% of the dataset: 483 samples) and test data (30% of the dataset: 207 samples). Both SVM and Gaussian Naïve Bayes are trained on the same training data and tested on the same test data. Gaussian Naïve Bayes doesn't have hyperparameters to tune, but SVM does. In SVM, rbf kernel is used. Grid search is performed to find optimal gamma (5*10^-6) - kernel coefficient for 'rbf'.

SVM classifier:

Confusion matrix:

	predicted class 0	predicted class 1
actual class 0	98	17
actual class 1	57	35

Accuracy score: 0.642512077295

Precision score: 0.673076923077

Recall score: 0.380434782609

GaussianNB classifier:

Confusion matrix:

	predicted class 0	predicted class 1
actual class 0	103	12
actual class 1	35	57

Accuracy score: 0.772946859903

Precision score: 0.826086956522

Recall score: 0.619565217391

As it can be seen from confusion matrices, accuracy, precision and recall scores, GaussianNB classifier performed better than SVM for this task.