Machine Learning:
Assignment Sheet #3

Due on March 1, 2022 at 10:00

Group HB
Henri Sota, Enis Mustafaj

Problem 3.1

In this exercise, you work with two data sets:

- Statlog (Shuttle) Data Set
- Computer Hardware Data Set

which are both available in the UCI Machine Learning Repository. For each of them, perform the following tasks:

a) Briefly describe the data set and all involved variables in your own words. If some information is missing on the UCI Repository site, do your own search for these details.

The data set of Statlog originated from NASA and concerns the position of radiators within space shuttles. Input samples are 9-dimensional vectors of continuous random variables. Output samples are 1-dimensional vectors of discrete random variable taking values from 1 to 7. The data was divided into a training set and a test set with 43500 samples in training set and 14500 in test set. Approximately 80% of training data belongs to class 1. There are only 6 samples that belongs to class 6.

The data set of Computer Hardware concerns the estimation of the relative performance of CPU based on the properties of the machine. The data set has 10 attributes, 6 predictive attributes describing the specifications for the memory and CPU cycles, 2 non-predictive attributes describing the vendor and the model name. There are 2 more attributes that provide the published relative performance and an estimated relative performance. The latter is not relevant as it is an output of a model that was created to predict the published relative performance.

b) Model the data set via input and output random variables / vectors.

Data representation of Statlog data set:

$$\begin{aligned} &\textbf{Input:} && \mathbf{X} : \Omega \to \mathbb{R}^9 \\ &\textbf{Output:} && \mathbf{Y} : \Omega \to \mathbb{N} \end{aligned}$$

The set of training data is represented as tuples containing the input feature and the output $\{(x_i, g_i)\}_{i=1}^N$

Data representation of Computer Hardware data set:

$$\begin{aligned} &\textbf{Input:} && \mathbf{X} : \Omega \to \mathbb{R}^6 \\ &\textbf{Output:} && \mathbf{Y} : \Omega \to \mathbb{R} \end{aligned}$$

The set of training data is represented as tuples containing the input feature and the output $\{(x_i, y_i)\}_{i=1}^N$

c) Formulate a question that can be solved using machine learning on this data set and give the type of machine learning (supervised / unsupervised / regression / classification) that will allow to answer the question.

The data set of Statlog was generated to extract rules for determining the conditions under which an auto landing would be preferable to manual control of a spacecraft. The task is to decide what type of control of the vessel should be employed. This is a supervised classification task.

The data set of Computer Hardware can be used to predict the relative performance of the machine given the machine specifications. This is a supervised regression task.

Problem 3.2

The Spambase Data Set is a SPAM classification data set that has exactly the 57 input variables that are roughly described in Example 2.9 of the lecture.

a) In Example 2.9, we did not mention the specific words and characters that are used in the features. Use the information of the UCI Repository to make a complete description of these variables, i.e. give all the key words, etc.

The features of the data set are:

word_freq_labs:	continuous r.v	word_freq_make:	continuous r.v
word_freq_telnet:	continuous r.v	word_freq_address:	continuous r.v
word_freq_857:	continuous r.v	word_freq_all:	continuous r.v
word_freq_data:	continuous r.v	word_freq_3d:	continuous r.v
word_freq_415:	continuous r.v	word_freq_our:	continuous r.v
word_freq_85:	continuous r.v	word_freq_over:	continuous r.v
word_freq_technology:	continuous r.v	word_freq_remove:	continuous r.v
word_freq_1999:	continuous r.v	word_freq_internet:	continuous r.v
word_freq_parts:	continuous r.v	word_freq_order:	continuous r.v
word_freq_pm:	continuous r.v	word_freq_mail:	continuous r.v
word_freq_direct:	continuous r.v	word_freq_receive:	continuous r.v
word_freq_cs:	continuous r.v	word_freq_will:	continuous r.v
word_freq_meeting:	continuous r.v	word_freq_people:	continuous r.v
word_freq_original:	continuous r.v	word_freq_report:	continuous r.v
word_freq_project:	continuous r.v	word_freq_addresses:	continuous r.v
word_freq_re:	continuous r.v	word_freq_free:	continuous r.v
word_freq_edu:	continuous r.v	word_freq_business:	continuous r.v
word_freq_table:	continuous r.v	word_freq_email:	continuous r.v
word_freq_conference:	continuous r.v	word_freq_you:	continuous r.v
char_freq_;:	continuous r.v	word_freq_credit:	continuous r.v
char_freq_(:	continuous r.v	word_freq_your:	continuous r.v
char_freq_[:	continuous r.v	word_freq_font:	continuous r.v
char_freq_!:	continuous r.v	word_freq_000:	continuous r.v
char_freq_$:	continuous r.v	word_freq_money:	continuous r.v
char_freq_#:	continuous r.v	word_freq_hp:	continuous r.v
capital_run_length_average:	continuous r.v	word_freq_hpl:	continuous r.v
capital_run_length_longest:	continuous r.v	word_freq_george:	continuous r.v
capital_run_length_total:	continuous r.v	word_freq_650:	continuous r.v
word_freq_lab:	continuous r.v		

The keywords are all of the substrings after the `word_freq_` part of the features that contain `word_freq_` in the beginning. The characters are `;, (, [, , $, #`. The frequencies of these keywords are captured. The remaining three features, `capital_run_length_average`, `capital_run_length_longest`, `capital_run_length_total` capture the average length of capitalized sequences, the length of the longest capitalized sequence, and the number of capitalized letters in the email respectively.

b) Search the web for alternative features that can be used to describe (SPAM) emails. Pick one example feature, cite the source, and describe these features.

Some alternative features to describe SPAM email are :

(a) From name - the name of the entity from which the email comes from

(b) From domain name - the name of the domain from which the email comes from

(c) Blocked IP - whether the email comes from a blocked IP address

(d) Apostrophe in From name - whether there is an apostrophe in the name of the entity from which the email comes from

(e) From address in User's Block list - whether the address from which the email comes from is in User's blocked list

(f) From address in User's White list - whether the address from which the email comes from is in User's white list

(g) Content Type - the type of content being transmitted by email (MIME)

(h) Content Boundary exists - whether the Content Boundary is present in the protocol's message

(i) From address and To address same - whether both the from address and to address of the email are the same

(j) Is subject present - whether the email contains a subject

(k) Subject content has obfuscate words = whether the subject of the email has obfuscated words

(l) Is forwarded message - whether the email has been forwarded

(m) Is reply message - whether the email is a reply to another email

(n) Subject Reply without reference header - whether the subject doesn't refer to a previous email exchange

(o) Sensual message - whether the email contains content considered as "sensual"

(p) Repeated double quotes in body - the number of times the email contains double quotes in its body

(q) Character set includes foreign language - whether the character set used in the email is part of a foreign language

(r) More blank lines in body - whether there are more than usual blank lines in the body of the message (consequent and more than one)

These are a subset of the features that have been considered in the source. [1]

Programming Problem 3.1

Implement a code that reads a text file and extracts the 57 features discussed in the previous task. Apply the feature extractor to the provided three example emails.
The feature extractor has been implemented in the file `programming_exercises.ipynb`. The interface of the class works by simply loading the desired file using its path based off of the current working directory, and then using the `extract` method to retrieve a map containing the output features and their values.

```
featureExtractor = FeatureExtractor()
emails = os.listdir("./emails")

# load the email files from "emails" directory
for email in emails:
    featureExtractor.load("./emails/" + email)
    output_features_map = featureExtractor.extract()
```

References

[1] V.Christina, S.Karpagavalli, and G.Suganya. "Email Spam Filtering using Supervised Machine Learning Techniques". In: *International Journal on Computer Science and Engineering* 2 (Dec. 2010).