Examination:
Programming for Data Science (ID2214)

Course code: ID2214
Course name: Programming for data science
Literature and tools:
The following rules apply to both parts of the examination:
Literature, other documents, including lecture slides, notes, etc. are not allowed.
Computers, tablets, phones, etc. are not allowed.
Date and time: Jan. 7, 2019, 08:00-12:00
Examiner: Henrik Boström

15 points are required to pass.
On part I, keep the text short and to the point.
The answers (including blank ones) should be numbered and ordered.
Unreadable answers will be ignored.
Good luck!

Part I (Theory, 10 points)

1a. Methodology, 2 points

Assume that we are organizing a Kaggle competition and have received 100 contributions (by independent teams), in the form of predictive models generated from 1000 training instances. We have now evaluated the models on a test set of the same size, which has been hidden from the teams, and found that the best performing model received an accuracy of 90.01%, which is just above the 90% threshold, which was a requirement for receiving an award of $100 000. Should we expect the best performing model to reach this level of performance when tested on a second test set of the same size, assuming it has been sampled from the same underlying distribution as the first test set? Explain your reasoning.

1b. Data preparation, 2 points

Assume that we want to employ one-hot-encoding in order to allow for a k-nearest neighbor classifier to be used on a dataset with categorical features only. What actions are required to handle any missing feature values before one-hot-encoding can be employed? Explain your reasoning.

1c. Naïve Bayes, 2 points

Assume that we want to use a naïve Bayes classifier on a binary classification task, with the class labels being $c1$ and $c2$ and involving the binary features $f1$ and $f2$. Moreover, asume a uniform class prior, i.e, $P(c1) = P(c2)$ and that the class conditional probabilities include $P(f1 = 0|c1) = 0$ and $P(f2 = 0|c2) = 1$. What class label $c \in \{c1, c2\}$ maximizes $P(c|f1 = 0 \& f2 = 1)$? Explain your reasoning.

1d. Performance metrics, 2 points

Assume that we have a binary classification model M that has been evaluated on four test instances. Is it possible that M receives an area under the ROC curve (AUC) of 1.0 and at the same time (for the same test set) an accuracy of only 50%? Explain your reasoning using an example.

1e. Clustering, 2 points

What are the relative strengths and weaknesses of k-means clustering compared to agglomerative clustering? Indicate clearly for which approach a property is considered a relative strength or weakness and for what reason.

Part II (Programming, 20 points)

2a. Data preparation, 10 points

Your task is to define the following Python function that converts a list of text paragraphs (lists of words) into an array using random projection:

```
random_projection(word_lists,rand_proj,dim)
```

which takes as input a list `word_lists` of lists of words (strings), a dictionary `rand_proj` which is a mapping from each possible word to a NumPy (random) vector of floats, and where all vectors have the same dimensionality `dim`. For example, the word `"Python"` could by `rand_proj` be mapped to `array([0.0, 0.0, 1.0, -1.0, 0.0])` and `"Julia"` could be mapped to `array([0.0, -1.0, -1.0, 0.0, 1.0])` (and hence `dim` should be 5).
The function should return a NumPy array of the shape `(p,dim)`, where `p` is the number of paragraphs (elements) in `word_lists`, and each row `r` in the array should be the vector sum of the corresponding random vectors for the words in the rth element of `word_list`, assuming that the empty list corresponds to a vector of zeros (of length `dim`).
For example, assuming that `rand_proj` includes the above mappings:

```
random_projection([["Python"],["Python,"Julia"]],rand_proj,5)
```

should return the following array:

```
array([[ 0.,  0.,  1., -1.,  0.],
       [ 0., -1.,  0., -1.,  1.]])
```

2b. Combining models, 10 points

Assume that you have been provided with a definition of the following Python function:

`decision_tree(df,min_leaf)`

which given a pandas dataframe `df`, where the columns correspond to features (except for a column named `CLASS` or `REGRESSION`, which contains the target values), the rows correspond to instances, and the integer `min_leaf` (which defaults to 5) specifies the minimum number of instances that may appear in a leaf node, returns a decision tree (represented by a NumPy array).

Your task is to define the following function:

`decision_forest(df,min_leaf,no_of_trees,no_of_features)`

which given a pandas dataframe `df` and an integer `min_leaf` (as specified above) returns a decision forest (a NumPy array) of `no_of_trees` (classification or regression) trees, where each tree is generated from a *bootstrap replicate* of the dataframe and a subset of `no_of_features` features that are randomly sampled (without replacement) prior to generating each tree, using the above `decision_tree` function.

Hint: You may consider using the function:

`np.random.choice(values,no_selected,replace)`

which given an array or list of `values`, randomly selects `no_selected` of them, with or without replacement, if `replace=True` or `replace=False`, respectively.