MAD 2023/2024 Exam

Bulat Ibragimov, Sune Darkner

Exam period: 15.01.2024 – 22.1.2024

This is the exam questions for the take-home exam on the course Modeling and Analysis of Data (MAD). The exam must be solved **individually**, i.e., you are **not allowed to work in teams or to discuss the exam questions with other students**. You have to submit your solution of the exam electronically via the **Digital Exam**[1] system. The submission deadline is **22 January 2024 at 10:00**.
Your solution should consist of

1. a pdf file `answers.pdf` containing your answers, and
2. a `code.zip` file containing the associated code (python files and / or Python Jupyter notebooks).

The grading will be done anonymously; **hence you should not mention your name anywhere in the answers or code**. Instead, you must **add your exam number** at the beginning of your `answers.pdf`.

WARNING: The goal of this exam is to evaluate your individual skills. We have to report any suspicion of cheating, in particular collaboration with other students, to the head of studies. Note that, if proven guilty, you may be expelled from the university. Do not put yourself and your fellow students at risk.

You are allowed to ask questions via the Discussions board in Absalon, but make sure that you do not reveal any significant parts of the solution. In doubt, just contact Bulat or Sune directly via e-mail! Any additional hints given by us will be made available to all of you via Absalon. Some further comments:

1. You **are allowed** to reuse the code that was made available to you via Absalon as well as the code you have developed in the course of the assignments. If you reuse code from the lectures or from the assignments, make sure to put a reference to this in your code, and if your code was developed as part of an assignment, in collaboration with a fellow student, add a corresponding comment to your answers, although keeping anonymity (i.e., just mention which parts stem from team work). In case you reuse code snippets you have found on the internet, please make sure that you provide a reference to this external source as well.
2. In case you notice any inaccuracies in the problem descriptions below, please let us know. If needed, we will provide updates and additional comments via Absalon. Thus, make sure that you check Absalon for announcements and discussions regularly!
3. For the coding tasks, you are given Python files/Jupyter notebook templates. You are supposed to complete these files and notebooks, but you are allowed to convert a Jupyter notebook to a python file or the opposite. Note that you are allowed to import additional Python packages and to make use of the functions provided by, e.g., the Numpy package. However, you should not use built-in functions if we ask you to implement a specific algorithm without using existing implementations (e.g., a single `kmeans` function from some other package that implements the K-means clustering approach). If in doubt, please ask us via the discussion board in Absalon!
4. All code templates and data can be found in the files `code.zip` and `data.zip`.
5. The deadline is hard (late submissions are not allowed), so make sure to submit in good time before the deadline. Up until the deadline it is possible to upload new versions of your solution several times. In the unlikely event that the Digital Exam system fails when you submit just at the deadline, then immediately send an e-mail to `uddannelse@diku.dk` and `bulat@di.ku.dk` with an explanation of what happened and attach your solution (the pdf and zip files) to the exam. Your solution will be assessed if you have a valid excuse and submitted on time.
6. Good luck! :-)

[1] `https://eksamen.ku.dk/`

Statistics

In this part, we will test your knowledge and skills in performing statistical analysis.

Question 1 (**Maximum Likelihood Estimation, 1 points).** Let X be a continuous random variable with the probability density function given below and having a parameter $\theta \in \mathbb{R}_+$ (the set of positive real numbers),

$$f(x;\theta) = \begin{cases} \frac{\theta}{x} \cdot \exp\left(-\frac{\theta}{x^2}\right) & \text{for } x \in \mathbb{R}_+ \\ 0 & \text{otherwise.} \end{cases}$$

Prove that the maximum likelihood estimate $\hat{\theta}$ for the parameter θ is given by

$$\hat{\theta} = \frac{N}{\frac{1}{x_1^2} + \frac{1}{x_2^2} + \ldots + \frac{1}{x_N^2}} \ .$$

Deliverables. Provide the essential steps and argumentation in your proof.

Question 2 (**Hypothesis testing, 2 points).** Let X be a normal distributed random variable with $X \sim \mathcal{N}(\mu, \sigma^2 = 0.1)$ where μ is unknown. Further assume that we are given this data set of samples from X,

$$\{8.1, 7.5, 8.7, 8.3, 8.5, 8.2, 8.9, 8.2, 8.7, 7.6, 8.5\} \ .$$

Perform hypothesis test at 5%-significance level of the null hypothesis $H_0 : \mu \geq 8.5$ against the alternative hypothesis $H_A : \mu < 8.5$.

Deliverables. Provide the steps you follow in order to perform the test as well as your conclusion.

Principal component analysis

Question 3 (**Principal Component Analysis, 3 points).** The aim of this task is to test your understanding of the principal component analysis algorithm. You are given the following set of $N = 6$ 2-dimensional points. The first dimension x is the following:

Points						
coordinate x	0.5	1.1	-0.7	1.5	-1.2	0.9

The second dimension y is defined as $y = \frac{2x}{4-x}$. Please compute y and round all values to the 2 decimal places. For example, 2.345 should be rounded to 2.35. Your task is to compute the principal component decomposition for these points. You can use either N or $N-1$ in the denominator during calculation of the covariance matrix.

Deliverables. Step-by-step calculation of the principal components. Your report should include mathematical derivations of the following:

1. Mean point of the data.
2. Two eigenvalues.
3. Two eigenvectors.

kNN

Question 4 (**Regression, 5 points).** The aim of this task is to test your understanding of methods for regression. You are provided with a database of patients with and without a heart disease. The patients are separated into training `heart_simplified_train.csv` and testing parts `heart_simplified_test.csv`. Each patient is characterized with six features: **Age, RestingBP, RestingECG, Cholesterol, MaxHR, Sex, ChestPainType**. Four features **Age, RestingBP, Cholesterol, MaxHR** are numerical features, while features **Sex, ChestPainType** are categorical. The last column in the database files named **HeartDisease** is the binary label representing the heart disease. Your task is to predict manifestation of the heart disease using patient features.

The problem can be solved using nearest neighbor classifier. More precisely, for a vector $\mathbf{x}$, the prediction is given via

$$f(\mathbf{x}) = round(\frac{1}{k} \sum_{\mathbf{x}_n \in N_k(\mathbf{x})} t_n)$$

where $N_k(\mathbf{x})$ denotes the set of the $k \geq 1$ nearest neighbors of $\mathbf{x}$ in the set $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ of training points.

1. (2 points) Extend the Jupyter notebook `kNN_students_2023.py` and implement nearest neighbor classification in Python for arbitrary numbers $k \geq 1$ of nearest neighbors. Consider only numerical features and use of the sum of absolute differences $d(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{D} |p_i - q_i|$ for data points $\mathbf{p}, \mathbf{q}$. Use the training set to train the model using $k = 1, 3, 5, 7, 9$ neighbors. Afterwards, apply the model to the instances given in the test set. Compute the RMSE and the prediction accuracy between the predictions made by the model for the test instances and the corresponding test labels. The accuracy is the number of correctly predicted languages divided to the total number of samples (accuracy lies between 0 to 1).

2. (3 points)Further extend the Jupiter notebook `kNN_students_2023.py` to handle both numerical and categorical features. Your kNN should be able to take a list of features where some are numerical, while some are categorical with textual values. For a numerical feature i, the distance is calculated as $d_{num}(p_i, q_i) = |p_i - q_i|$. For a categorical feature j, the distance is calculated as an indicator function $d_{cat}(p_j, q_j) = 1_{p_j \neq q_j}$. The combined distance is calculated as:

$$D(\mathbf{p}, \mathbf{q}) = w_{num} \sum_{\text{I} \in I} d_{num}(p_i, q_i) + w_{cat} \sum_{\text{J} \in J} d_{cat}(p_j, q_j),$$

where w_{num} and w_{cat} are the weight factors multiplied to numerical and categorical features. Please use fixed $k = 5$ and $w_{num} = 1$. Try different $w_{cat} = 0.025$, $w_{cat} = 0.05$ and $w_{cat} = 0.1$. Report the RSME and classification accuracy for these values of w_{cat}.

Comments: You are supposed to implement both parts on your own, i.e., you are not allowed to use, e.g., corresponding class/functions from the *Scikit-Learn* package. Note that you are allowed to make use of the *Numpy* package. In doubt, please get in touch with us.

Deliverables. All your source code used to solve this exercise (as a Jupyter notebook or Python script file(s)). Add to your report:

1. RMSE and accuracy value for the optimal k and scatter plot with the dependency between the accuracy and k.

2. RMSE and accuracy value for the w_{cat} values of interest mentioned in the task.

Classification

In this part, we will test your knowledge and skills in performing classification of data.

Question 5 (Random Forests, 2 points). This question tests your understanding of random forest construction, in particular the selection of the optimal threshold value. Let's say you have a training set of the following samples:

Points												
feature	10	20	30	40	50	60	60	70	80	90	100	110
label	1	0	0	2	1	1	0	0	0	0	1	0

Each data sample is defined with one feature and has a binary label. Your aim is to compute the optimal threshold for separating this data into two subsets using a selected metric (Please use information gain). You are asked to calculate all the steps manually and give explanations in words when appropriate:

a) Correct use of information gain based on entropy
b) Correct identification of potential thresholds to consider
c) Calculating the metric values for the potential thresholds and finding the optimal threshold.

Deliverables. For a) and c), include your calculations. For b), write an explanation of which thresholds are worth considering. Do not forget for each threshold to explicitly write the metric values.

Question 6 (Classification & Validation, 5 points). This task aims to test your understanding on the training/validation data separation and classifier parameter tuning. Use training data from `heart_simplified_train.csv` and validation data from `heart_simplified_validation.csv`. Use the implementation of Random forests from the `sklearn` Python package to do the following:

a) (1 points) Convert categorical features into numerical. The following is an example how you can move from categorical to numerical features. If a categorical feature can take only two values: A and B; it can be turned into numerical by simply replacing A with 1 and B with 0. Suppose that a categorical feature i can take three values: A, B and C. Remove feature i from the list of features but add three more features: feature j that equals to 1 if i was equal to A (and 0 otherwise) for patient p; feature k that equals to 1 if i was equal to B (and 0 otherwise) for patient p; feature l that equals to 1 if i was equal to C (and 0 otherwise) for patient p.

b) (1 point) Implement random forest training using the data from `heart_simplified_train.csv` having all categorical features converted to numerical. Report the accuracy of the random forest model on the training data.

c) (2 points) Use the validation data to find the optimal set of random forest classifier parameters. The parameters to test: 1) criterion (entropy or gini); 2) maximal tree depth (values to test 2, 5, 7, 10, 15); 3) the number of features to consider for each split ($sqrt$ and log_2 of the total number of features). To find the optimal set of parameters, try 15 randomly generated options for criterion, tree depth and split rule using lists 1), 2) and 3). Calculate and print two metrics during the optimal parameter search. The first metric is the number of correctly classified validation samples. The second metric is the average probability assigned to the correct class for all validation samples. Use the second metric to select the optimal set of parameters. Suppose, that random forests assigns probability [0.7, 0.3] of sample with label 1. So it is correctly classified, because 0.7 ¿ 0.3, but the probability is not 1. Add into report the code fragment that calculates both metrics and updates the optimal parameter set if these metrics are superior to the metrics for the previously found optimal set.

d) (1 point) Add into the report the accuracy improvements during the optimal parameter search. Every time you find a more optimal set of parameters, print a statement "criterion = ? ; max_depth = ? ; max_features = ? ; accuracy on validation data = ? ; number of correctly classified validation samples = ?". Replace question marks with the appropriate values.

Deliverables. a) Provide a code snippet in the report of the essential steps, b) code snippet in the report of the essential steps, c) provide the obtained results in the report as well as your reflections on these.

Clustering

In this part, we will test your knowledge and skills in performing clustering of data.

Question 7 (7 points). This task aims to test your understanding of clustering and principal component analysis. The task includes the following:

a) (1 point) Read and normalize data from the comma-separated data file `housing.csv`. The dataset consist of 999 samples. Each line in the file represent one sample. Each sample is defined with 9 features **MedInc, HouseAge, AveRooms, AveBedrms, Population, AveOccup, Latitude, Longitude, MedHouseVal**. Normalize the data using the minimum and maximum values computed for each feature. That is, for the i'th sample, the normalized **HouseAge** feature value is $HouseAge_i^{norm} = (HouseAge_i - HouseAge_{min})/(HouseAge_{max} - HouseAge_{min})$, where $HouseAge_i$ is the original **HouseAge** feature value for the i'th sample.

b) (3 points) Implement hierarchical K-means clustering without using any existing implementation. Test your K-means implementation using the normalized data from a) and set number of clusters to $K_0 = 3$ on the first level and $K_1 = 2$ on the second level and $K_3 = 2$ on the third level. Run K-means 5 times using random samples from the dataset as the initial cluster centers. Select the solution with the smallest intra-cluster distance. Compute and print the number of samples in each cluster in the final solution you obtained in b).

c) (1 point) Compute the principal components of the data. You are allowed to use existing implementations of the principal component analysis.

d) (2 points) Transform the data using the two largest components, i.e. eigenvectors with the largest eigenvalues. Plot the clustering results including the transformed cluster centers and transformed data. Your plot should contain 999 colored dots representing data samples and 2*2*3 black dots representing the centers of the clusters. The dot associated with i'th sample should have coordinates $(y, z) = PCA_2(MedInc_i, HouseAge_i, ..., MedHouseVal_i)$, where PCA_2 is the transformation defined by projection onto the two largest components, and the color defined by the cluster label of i'th sample obtained in b). In Figure 1, you find an example of such a plot with only three clusters. Note that you should run clustering on the original 9-dimensional data, save the clustering labels in some array, transform the data to 2-dimensional space and then plot the data samples colored according to the saved clustering labels.

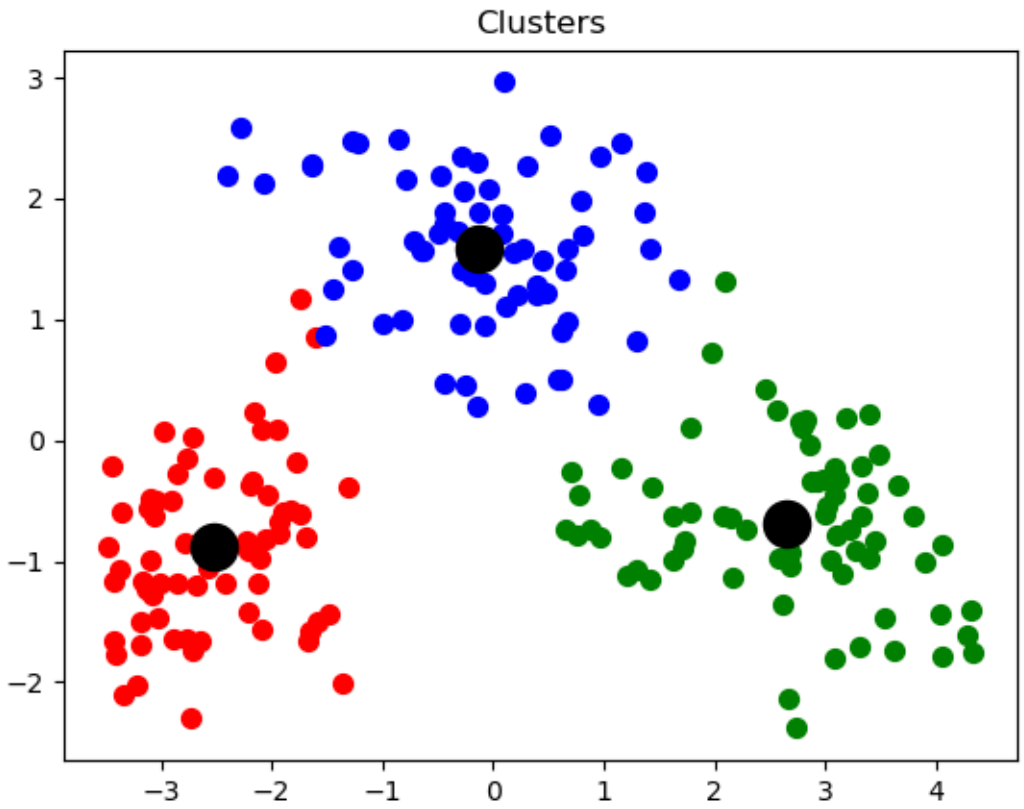

Figure 1: Example of plot we ask for in question 7(e). The black dots represents the cluster centers.

Deliverables. a) Provide a code snippet in the report showing how you perform the normalization, b) provide a code snippet in the report showing your K-means implementation, c) include the number of samples in each cluster in the report, d) provide a code snippet and explanation of what you did in the report, e) include a code snippet, the plot and your reflections on the results in the report.