Data Science

Shahid Beheshti University
Spring 2025
Classification Challenge

Assignment #4

1 Theoretical

Answer the following questions. Justify all answers with clear reasoning or computation.

Q1. (Pattern Recognition Systems)

(a) List and describe the four core components of a supervised pattern recognition system.

(b) Explain how the roles of the "teacher" and "learning algorithm" interact in training.

(c) Illustrate the difference between training mode and classification mode.

Q2. (Classification vs. Regression)

(a) Define the main difference between classification and regression.

(b) For each, identify the type of output variable and give one application from the slides.

Q3. (Feature Vectors and Hidden States)

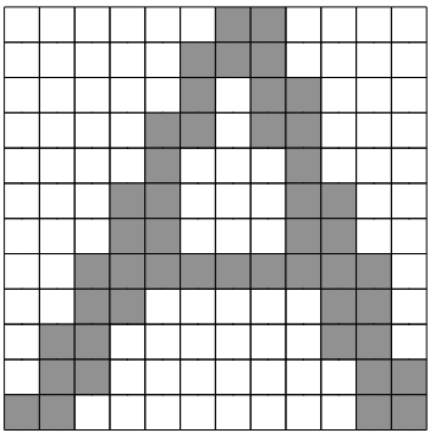
(a) Define a feature vector and explain what it represents in pattern recognition.

(b) What is a hidden state and how is it related to the classification task?

Q4. (Image-Based: Template Matching)

(a) Based on the image below, explain how template matching classifies input patterns using training templates.

(b) Why does template matching struggle when fewer templates are stored?



0	0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0
0	0	0	0	0	1	0	1	1	0	0	0
0	0	0	0	1	1	0	1	1	0	0	0
0	0	0	0	1	0	0	0	1	0	0	0
0	0	0	1	1	0	0	0	1	1	0	0
0	0	0	1	1	0	0	0	1	1	0	0
0	0	1	1	1	1	1	1	1	1	1	0
0	0	1	1	0	0	0	0	0	1	1	0
0	1	1	0	0	0	0	0	0	1	1	0
0	1	1	0	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	0	0	1	1

Q5. (Decision Regions and Boundaries)

(a) What is a decision region in classification?
(b) Define and illustrate the concept of a decision boundary.
(c) How does uncertainty affect the clarity of decision regions?

Q6. (Bayes Classification Rule)

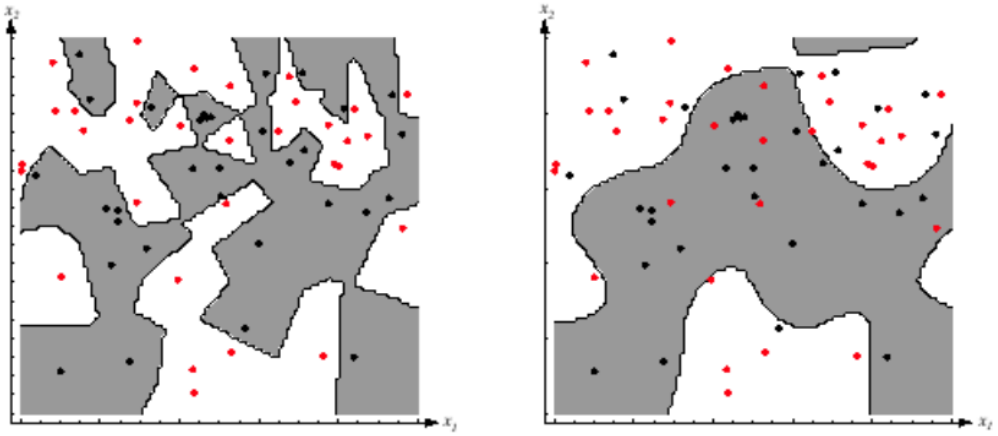
(a) State Bayes' theorem in the context of classification.
(b) Derive the decision rule for assigning an observation to a class.
(c) Explain the terms: prior, likelihood, posterior, and evidence.

Q7. (Probability Density Function Estimation)

(a) Why is estimating the PDF of feature vectors important for Bayes classifiers?
(b) Compare parametric vs. non-parametric estimation, with one example each.

Q8. (Image-Based: Decision Boundaries)

(a) Observe the figure below showing decision regions. Describe how the classifier forms the boundaries.
(b) What happens to the boundaries if classes have overlapping features?



Q9. (1-NN and Similarity-Based Methods)

(a) Explain how the 1-Nearest Neighbor rule works using similarity.

(b) Why is the 1-NN considered implementation-independent?

(c) How does the choice of metric affect classifier accuracy?

Q10. (Parzen and k-NN Density Estimation)

(a) Describe the Parzen window method for estimating a PDF.

(b) How does k-NN PDF estimation differ from Parzen?

(c) What are the challenges in selecting parameters h or k?

Q11. (Discriminant Functions)

(a) What is a discriminant function?

(b) Write the discriminant function for multivariate Gaussian classes.

(c) How does this function relate to decision boundaries?

Q12. (Ensemble Learning and EM Algorithm)

(a) Explain how the EM algorithm is used to estimate parameters in a Gaussian mixture.

(b) Why might ensemble methods outperform a single classifier?

2 Practical

2.1 Introduction

In this assignment you will act as data-scientists for a fictitious data. Your mission is to build a model that predicts a 11 class category from 64 synthesized flags. The features mimic "present / absent" indicators such as abnormal lab values, symptoms, or billing codes, making the task surprisingly subtle despite the modest table size.

2.2 Dataset and Competition Setup

You have three files on This Kaggle competition:

train.csv contains 564 rows, each with an integer `ID`, 64 binary features (`feature0` – `feature63`) and the target `label`. **test.csv** mirrors that structure but omits the label, holding 143 rows that will decide the leaderboard. **sample_submission.csv** is a two-column template (`ID`, `label`). Finally, **y_test.csv** stores the hidden ground truth and is visible only to Kaggle's scoring engine. No feature is missing; every cell is either 0 or 1, and the label is an integer 0–10, giving us eleven classes.

After logging in you may submit up to twenty prediction files per calendar day. Kaggle shows a public score based on 50% of the hidden labels; the private score—used for marking—remains concealed until the deadline.

Your bonus score will be computed by the following formula and added directly to your overall grade!

$$\text{Bonus} \;=\; \frac{2}{1 + 0.5 * (\text{rank} - 1)}$$

2.3 What You Must Deliver

Your work has two facets. First, a *fully reproducible* notebook (Jupyter or Colab) that loads the data, explores it, trains at least one solid model, and writes a submission file. Second, a concise PDF report that we can read without running any code. The report should narrate your decisions, highlight discoveries, and justify the final architecture. When grading we will retrain your best model offline, so keep random seeds fixed and explain any non-deterministic step.

2.4 Suggested Workflow

Begin with a light exploratory analysis: inspect class balance, plot feature frequencies, and calculate simple mutual-information statistics to see which flags hint at certain diseases. Because the inputs are already binary, most classifiers will run straight out of the box—yet you might experiment with alternative encodings or dimensionality reductions such as PCA or UMAP if you think they help.

Next, construct baseline learners. Logistic regression is a welcome starting point, but do not stop there. Try at least two other families: a support-vector machine (linear, RBF, or polynomial kernel), a decision tree pruned for generalisation (sometimes called a "good tree"), or a probabilistic method such as Naïve Bayes or quadratic discriminant analysis. k-nearest-neighbour and Gaussian-process classifiers are also permitted. **Neural networks are off-limits** in this exercise—our aim is to practise classic, interpretable tabular ML.

When you have baselines, venture into ensemble territory. Bagging methods like Random Forests and Extremely Randomised Trees often shine on binary attributes, whereas boosting families—AdaBoost, Gradient Boosting, XGBoost, LightGBM, CatBoost—can squeeze out the final few per-cent if tuned with care. A soft- or hard-voting combiner, or a stacked meta-learner that blends diverse bases, is equally acceptable. Whichever road you take, describe clearly what you tuned (e.g. number of trees, learning rate, maximum depth, SVM C and γ) and how you chose final parameters.

2.5 Model Interpretation

Numbers are not enough. Therefore reserve a slice of your analysis for feature importance. Permutation tests, impurity measures in tree ensembles, or SHAP values all earn credit. Summarise your findings in plain language: *"Indicators 12, 27, and 45 are the strongest red flags for class 7 . . . "*. Finish with two concrete suggestions for how such knowledge could guide data collection or triage.

2.6 Evaluation and Minimum Performance

Kaggle's metric is simple overall accuracy on the leaderboard. To pass the assignment your best private score must reach at least **0.35**. Any solution below that line receives partial credit only. to participate in kaggle competition, you must score at least **0.40**

2.7 Submission

Upload (1) the notebook; (2) the PDF report; and (3) either a compressed folder.