SCIPER: ________________ **NAME:** ____

Machine Learning for Behavioral Data

School of Computer and Communication Sciences

Professor: Tanja Käser

CS-421

Do not turn the page before the start of the exam. This document is double-sided and contains 18 pages.

- The exam consists of two parts: conceptual questions (worth 100 exam points) and coding questions (worth 100 exam points) for a total of 200 points.
- The **conceptual part of the exam (this booklet) is due at 10h15**, when we will collect your exam sheets.
- The coding part is ready in your Noto directory and stays available for the whole duration of the exam. You will **only be allowed to use Noto** to solve the exam. You will turn in your notebook on Moodle in the assignment called *MLBD Final Exam: Coding Questions.*
- The exam is **open book**. You are **allowed** to use all resources on the internet (including Moodle, lecture materials, previous Stackoverflow posts, blogs, GitHub, and your notes) on both the conceptual and coding parts of the exam.
- However, you are strictly **NOT allowed** to **communicate** with your peers or outside entities (including models like **ChatGPT**, **CoPilot**, etc.), post the questions online, or post on this forum during the exam.
- Please do not cheat. We will report any such occurrence to the legal service of EPFL.

Time: 9h15 - 10h15 July, 1$^{\text{st}}$ 2023 Maximum Mark: 200

Streaming Service Shenanigans

You are a data scientist at *NetPrime*, a popular movie and series streaming company. It's very important for your business model that your servers are up at all times, but recently, lots of users have been reporting that they cannot access some movies. You look at the logs, and you notice that lots of your machines are crashing. Therefore, you are modeling the time it takes for a computer system to crash after start-up (`time_to_crash`).

Task 1 (10 points)

(a) Given that `time_to_crash` is numerical, you model it using linear regression. You use your model to predict `time_to_crash` for a new computer system given the brand (`brand_computer_system`) and the age (`age_of_computer`). The model predicts that the time it will take to crash after start-up is -60 seconds. How is this possible?

(b) Fix the problem, i.e. make sure that your model only predicts `time_to_crash >= 0`. Provide your regression equation and explain your solution.

(c) Your team is studying the mutual information (MI) between the features `time_to_crash` and `age_of_computer`. Using the same data, one colleague says the MI between these variables is 0.6, while another one says that it is 0.8. Why do they have different values?

Task 2 (12 points)

Because of these outages, a considerable number of users have canceled their subscriptions. You want to build a classification model. Using the information from the previous month, you want to predict if a user will cancel their subscription in the next month. You have the following features (from the previous month): the number of days in which the user was active, the median hours watched, and the number of different series the user watched.

You want to build a decision tree as a baseline classifier. You take a random sample of 10 points:

number of active days	**median hours watched**	**number of series watched**	**subscription cancelled**
5	3	1	True
5	2	1	False
7	2	1	True
5	3	2	True
5	3	2	False
7	1	1	True
7	2	2	False
5	3	1	False
7	1	2	True
7	1	1	False

(a) Using only the 10 points sample, on which attribute (name the attribute) should you do the first split and why?

(b) Having built the decision tree, you ask your interns to find the best hyperparameters for the model. They want to tune the following parameters: `min sample split` (2, 3, 4) and `max depth` (2, 3, 4). The use 5-fold cross validation to tune the hyperparameters and assess the generalization error of the model. They obtain a `min_sample_split` of 2 and `max_depth` of 2 as optimal hyperparameters. Here are their results per fold:

- Accuracy Fold 1: 0.90
- Accuracy Fold 2: 0.80
- Accuracy Fold 3: 0.85
- Accuracy Fold 4: 0.80
- Accuracy Fold 5: 0.90

They report that the generalization error of the model is 0.85.

Explain what is wrong with their evaluation procedure and how you would improve their work.

MOOC Mania

You decide to quit your previous job as a streaming service data scientist because you realize that your life calling is in teaching data science instead! You teach a MOOC about data science on the platform CoursEdX.

Your class is wildly popular with thousands of students enrolled. However, you notice that there are a number of bots pretending to take your class and posting random things on the discussion forums. You decide to address this problem by building a spam filter.

Task 3 (7 points)

You continue labeling discussion posts in your course to identify spam. You see that the dataset is severely imbalanced, with 80% of posts with no-spam and 20% of posts with spam.

You consider the following three models:

- *MAJORITY* always predict class no-spam
- *UNIFORM* predicts class no-spam 50% of the time and class spam otherwise
- *PROP* predicts class no-spam 80% of the time and class spam otherwise

(a) Which of these models will have the highest accuracy on the dataset? Explain.

(b) Which of these models will have the highest balanced accuracy on the dataset? Explain.

Task 4 (2 points)

To deal with the class imbalance of your discussion post data, you decide to use resampling methods instead of a single model with a train-test-val split. It's been a few years and you're a bit confused about the details of the different approaches. Which of the following is true regarding evaluating these models?

Circle the correct answer, no justification required.

(a) Out-of-Bag error estimate uses a separate validation set.

(b) K-fold cross-validation uses a separate validation set.

(c) Out-of-Bag error estimation is more computationally expensive than K-fold cross-validation.

(d) Out-of-Bag error estimation provides an unbiased estimate of generalization error.

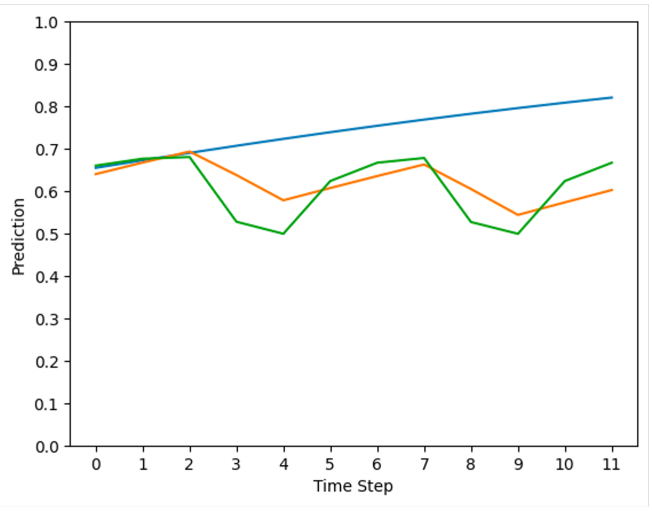
(e) K-fold cross-validation provides an unbiased estimate of generalization error.

Now that you are sufficiently happy with bots not taking over your discussion forums, you decide to focus on improving students' performance. You model your student performance data and analyze how they learn skills.

Task 5 (9 points)

You train an AFM, a PFA, and a BKT model on your students' data (their correct or wrong answers on skills taught in the course). You obtain the following model parameters from the trained models:

- BKT: $p_o = 0.6$, $p_s = 0.1$, $p_g = 0.3$, $p_l = 0.3$, $p_f = 0.3$
- AFM: $\theta = 0.2$, $\beta = 0.44$, $\gamma = 0.08$
- PFA: $\theta = 0.3654$, $\beta = 0.21$, $\gamma = 0.12$, $\rho = -0.25$



You use these three models to predict the probability that a student has learned the skill s at time step t over the first 12 time steps. The graph shows three models' predictions for a student with observed question interactions [1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1], where 1 represents a correct answer and 0 represents incorrect.

Which model generates each line? Why?

Blue:

Orange:

Green:

Task 6 (15 points)

You realize that your student performance dataset actually has three labels: correct, partially correct, wrong. Since BKT only has correct and wrong options for student answers, you decide to adjust the BKT model to be able to handle these three different outcomes.

To do so, you introduce two new parameters: p_{hg}, the probability of a partial guess (getting a question partially correct despite not having mastered the skill) and p_{hs}, the probability of a partial slip (getting a question partially correct despite having already mastered the skill). The BKT emission probability table is modified as follows:

o_t	s_t	$p(o_t\|s_t)$
wrong	not mastered	$1 - p_{hg} - p_g$
partially correct	not mastered	p_{hg}
correct	not mastered	p_g
wrong	mastered	p_s
partially correct	mastered	p_{hs}
correct	mastered	$1 - p_s - p_{hs}$

$p_0 = 0.7$

$p_S = 0.1$

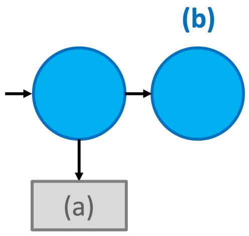
$p_{HS} = 0.2$

$p_G = 0.1$

$p_{HG} = 0.2$

$p_L = 0.1$

$p_F = 0.2$



After training your model on the dataset, you obtain the following parameters: p_0 = 0.7, p_s= 0.1, p_{hs} = 0.2, p_g = 0.1, p_{hg} = 0.2, p_l = 0.1, p_f =0.2.

(a) What is the predicted probability p(o_0 = correct) that a student solves the first question correctly?

(b) What is the predicted probability p(s_1 = mastered | o_0 = correct) that a student has mastered the skill before the second attempt, given that they got their first attempt correct?

Task 7 (10 points)

Now that your BKT model is finalized, you want to design interventions for your failing students to get them back on track. You have last year's data from a related class with some features describing each assignment, student performance on each assignment, and their overall performance in the class. Using these, you design a pass-fail prediction LSTM. Before deploying it for all of your students, you want to carefully analyze the model.

Specifically, you are interested in finding which assignments in the course are most predictive of student success. In this way, you can identify which assignments to spend time improving with your limited TA support and which ones are less important. Using your model, how can you design an explainability analysis to help with this? Which method(s) would you use? Justify your decisions.

Task 8 (10 points)

Your pass-fail model seems to be working well, but you now want to design a time-series model with a separate quiz score prediction for each week to model your students' changing performance and design interventions at the right time in the course. You already have working code for a traditional pass-fail student success prediction LSTM network. You now want to modify it to a time-series prediction of student quiz performance each week.

State all the changes you need to make to the model architecture below and explain why.

```
def create_model(nb_features, nb_skills, params):

inputs = tf.keras.Input(shape=(None, nb_features), name='inputs')

x = tf.keras.layers.Masking(mask_value=params['mask_value'])(inputs)

x = tf.keras.layers.LSTM(params['recurrent_units'],
                      return_sequences=False,
                      dropout=params['dropout_rate'])(x)

dense = tf.keras.layers.Dense(nb_skills, activation='sigmoid')
outputs = dense(x)
model = tf.keras.models.Model(inputs=inputs, outputs=outputs, name='TimeSeries')

model.compile(loss=tf.keras.losses.binary_crossentropy,
             optimizer=params['optimizer'],
             metrics=[tf.keras.metrics.AUC(), 'binary_accuracy'])
return model
```

TrioLingo: Leisurely Language Learning

Now that things are going smoothly with your MOOC course, you decide to pick up language learning as a hobby. You find a very cool app – TrioLingo, and decide to involve all of your friends.

Task 9 (8 points)

You, Paola and Vinitra are trying to master Italian by the beginning of August so you can go on a summer vacation together. During the month of July, your first exercise was 5 points, second exercise was 5 points and 6,7,7,8,6,7,8 points respectively on the following days over 9 total sessions. Likewise, Paola scored 5,6,7,8,8,9 points over 6 sessions and Vinitra scored 6,6,7,6,7,7,8,8 points over 8 sessions. Using dynamic time warping (minimizing L1 distance), calculate the distance (minimum distance path) between the three TrioLingo participants (You, Paola and Vinitra). Which two TrioLingo learners had the closest study performance and what was their distance? Show your work.

Task 10 (8 points)

You convince five more of your friends to also learn Italian on TrioLingo with you. You would like to create subgroups based on their Italian level. Matrix A shows the Italian-level similarity between the five of them: Alex, Bastien, Camille, Dominique and Evan (in that order). For example, Alex and Bastien are 43% similar, and Dominique and Bastien are 65% similar. Draw the resulting similarity graph after applying the mutual 1-nearest graph algorithm. Who will be grouped together? Show your work.

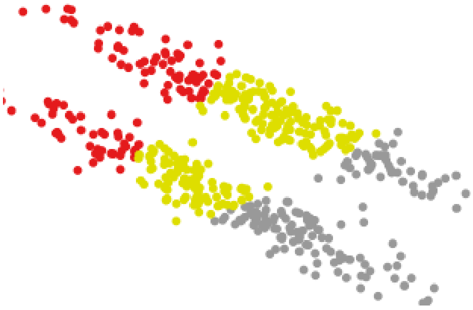
$$A = \begin{bmatrix} 1.00 & 0.43 & 0.16 & 0.75 & 0.32 \\ 0.43 & 1.00 & 0.98 & 0.65 & 0.27 \\ 0.16 & 0.98 & 1.00 & 0.47 & 0.68 \\ 0.75 & 0.65 & 0.47 & 1.00 & 0.91 \\ 0.32 & 0.27 & 0.68 & 0.91 & 1.00 \end{bmatrix}$$

Task 11 (6 points)

Your friend, an aspiring data scientist, used the following method to cluster the TrioLingo users:

```
cluster.KMeans(n_clusters=3,init='random',n_init=1)
```

He shows you the following results:



What three pieces of advice would you give him to improve his solution and why?

Task 12 (3 points)

Learning a language is harder than expected and you're getting a bit discouraged. You want to see how your peers are performing on TrioLingo to see if you really have a chance of learning Italian in time for the trip. To collect data on EPFL students' use of Triolingo, you stand outside near the Esplanade and ask 200 students for their usage statistics. Afterwards, you design a model on this data and you note that the predictions are unfair for certain cases. What is the most likely kind of bias you identify in your subset? Explain why.

Additional Space

Additional Space

Additional Space

Additional Space