DATA-448
Exam II

Exam Instructions

1. **Show all your work** and write complete and coherent answers.
2. Show all of the steps that you used to get your final answer. If you do not show your work I can not give partial credit in the case of incorrect answers.
3. **Please strive for clarity and organization.**
4. **You are not allowed to discuss any of the exercises in exam 2 with others.**
5. **Late submission will not be accepted, regardless of the circumstances.**

Take the time to carefully read all the questions on the exam. GOOD LUCK!

1. (20 points) **True or False**.
   (a) `GridSearchCV` and `Optuna` will always produce the same results. (**True** **False**)
   (b) Support vector machine is an ensemble learning technique. (**True** **False**)
   (c) If the classifier produce a 80% recall on test dataset (using default values for hyper-parameters), there is no need to tune the classifier hyper-parameters. (**True** **False**)
   (d) `RandomSearchCV` and `Optuna` will always produce the same results. (**True** **False**)
   (e) Increasing the value of `max_depth` in tree-based models prevents overfitting. (**True** **False**)
   (f) Scaling the input variables in tree-based models can speed up the hyper-parameter tuning process. (**True** **False**)
   (g) In `GridSearchCV`, the number of trials can be specified by the practitioner. (**True** **False**)
   (h) XGBoost always outperforms random forest. (**True** **False**)
   (i) XGBoost is faster than regular gradient boosting algorithms because it trains the the base-learners in parallel. (**True** **False**)
   (j) In order to avoid overfitting, it is recommended to tune the learning rate hyper-parameter in random forest models. (**True** **False**)

2. (6 points) In what scenarios, `RandomSearchCV` would be preferred instead of `GridSearchCV`? Be specific.

3. (6 points) How does stacking work? Be specific.

4. (6 points) If you have trained three different models on the exact same training dataset, and they all achieve a 93% F1-score on the exact same dataset. Is there any chance that you can combine these three models to get better results? If so, how? If not, why? Be specific.

5. (4 points) If your XGBoost model overfits the data (probably because the number of boosted trees is large), what hyper-parameter you need to tune? Be specific.

6. (4 points) Which of the following algorithm is **NOT** an ensemble learning algorithm?
   (a) support vector machine
   (b) random forest
   (c) AdaBoost
   (d) XGBoost
   (e) None of the above

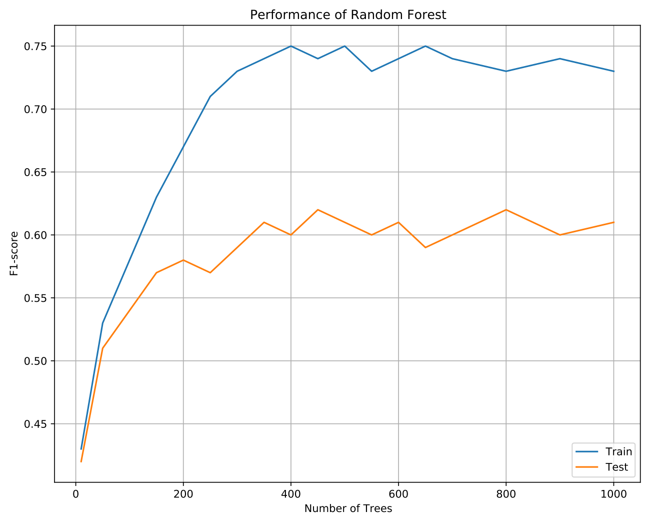
7. (6 points) How does XGBoost with random forest as base learner work? Be specific.

8. (4 points) How do we select the best hyper-parameter combination for a given model?
   (a) Performance on the training dataset
   (b) Performance on the validation dataset

(c) (a) and (b)

(d) None of the above

9. The below chart shows the performance of a random forest models (with different number of trees).



Using the above chart, answer parts (a)-(b).

(a) (4 points) If a data scientist wants to tune the number of trees based on the F1-score, how many trees would he select? Be specific.

(b) (6 points) What is the reason there is big difference in the F1-score performance between the `Train` and `Test` datasets? Be specific.

10. (6 points) What is the difference between homogeneous and heterogeneous ensembles? Be specific.

11. (4 points) Random forest is an example of

(a) Homogeneous ensemble

(b) Heterogeneous ensemble

(c) Bagging

(c) (a) and (b)

(d) (a) and (c)

(e) (b) and (c)

(f) None of the above

12. (8 points) A junior data scientist is working on building a classification framework that can help a small local credit union to identify fraudulent transactions. The data scientist

collects a very small dataset with 100 observations, from which 4 observations are labelled as fraudulent, to start building and tuning the model. As part of the tuning process, the data scientist considers using the `GridSearch` function from scikit-learn with `cv = 5`. Explain why this approach is a good idea or not. Be specific.

13. The typical hyper-parameters in a XGBoost model that data science practitioners tune are:

    - `n_estimators`
    - `learning_rate`
    - `max_depth`
    - `min_child_weight`
    - `subsample`
    - `colsample_bytree`

    (a) (4 points) What is/are the hyper-parameters that may cause overfitting if we increase their values while holding the values of the hyper-parameters constant? Be specific.

    (b) (4 points) What is/are the hyper-parameters that prevents overfitting if we increase their values while holding the values of the hyper-parameters constant? Be specific.

14. (6 points) In tree-based models such as random forest, gradient boosting, and XGBoost, how does a data scientist optimize the number of trees in a manually fashion? Be specific.

15. A Machine Learning Specialist is assigned to a team that is responsible for tuning an XG-Boost model to ensure that it performs correctly on test data. However, when dealing with unknown data, this does not operate as planned. The following table summarizes the existing hyper-parameters:

```
XGBoost_params = {'n_estimators': 2000,
                  'max_depth': 30,
                  'min_child_weight': 3,
                  'subsample': 0.9,
                  'objective': 'reg:squarederror'}
```

    (a) (3 points) What type of task the machine learning specialist is working on? Classification or regression? Be specific.

    (b) (5 points) Given the existing XGBoost hyper-parameters, what may be the reason that the XGBoost model is not performing as expected? Be specific.

16. (8 points) How can overfitting and underfitting be addressed through hyper-parameter tuning in random forest? That is, what hyper-parameters would you tune to address overfitting and underfitting in random forest? Be specific.

17. Consider the `Customers.csv` data file. This file contains basic information on customer data. The owner of a shop gets information about Customers through membership cards. This dataset consists of 2000 records and 8 columns:

    - `Customer ID`: customer ID.

- `Gender`: customer's gender.
- `AGE`: customer's age in years.
- `Annual Income`: customer's annual income.
- `Spending Score`: Score assigned by the shop, based on customer behavior and spending nature.
- `Profession`: customer's profession.
- `Work Experience`: customer's work experience in years.
- `Family Size`: number of family members.

The goal is to predict `Spending Score`; that is, this a regression task. This is a continuation from Exercise 16 in Exam 1. **In Python**, answer the following:

(a) (4 points) Using the pandas library, read the `Customers.csv` data file and create two data-frames, called it `customers`.

(b) (3 points) Drop the `Customer ID` variable.

(c) (5 points) Split the `customers` data-frame into: `train` (80%) and `test` (20%).

(d) (20 points) Using results from Exercise 16 part (e) in Exam 1, use the top 7 variables, build a model on the `train` data-frame, and tune this model using the `Optuna` framework. Notice that you need to tune the model based on the root mean squared error. Then, use the optimal model to make predictions on the `test` data-frame. Finally, compute the root mean squared error of the model.

(e) (20 points) Using results from Exercise 16 part (e) in Exam 1, use the top 7 variables, build another model (different from the one in part (d)) on the `train` data-frame, and tune this model using the `Optuna` framework. Notice that you need to tune the model based on the root mean squared error. Then, use the optimal model to make predictions on the `test` data-frame. Finally, compute the root mean squared error of the model.

(f) (20 points) Using results from Exercise 16 part (e) in Exam 1, use the top 7 variables, build another model (different from the one in parts (d)-(e)) on the `train` data-frame, and tune this model using the `Optuna` framework. Notice that you need to tune the model based on the root mean squared error. Then, use the optimal model to make predictions on the `test` data-frame. Finally, compute the root mean squared error of the model.

(g) (20 points) Using the predictions from parts (d), (e), and (f), ensemble them by taking the average of those predictions. Finally, compute the root mean squared error of this ensemble.

(h) (5 points), Using the results from parts (d) to (g), what model/approach would use to predict `Spending Score`. Be specific.