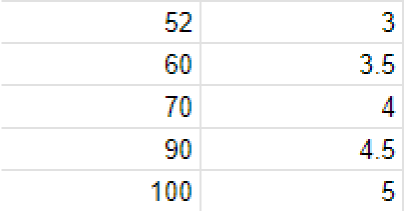
Projekt zaliczeniowy

OMÓWIENIE I CELE

For each point in this project description I add a reference to the folder with appropriate materials that we discussed in the lesson.
Grading:
The total is 100 points. 52 == 3, and is a threshold.



52	3
60	3.5
70	4
90	4.5
100	5

Wymagania do projektu

1. (Can be skipped)Selected problem implemented in the form of a Jupyter notebook and made available online*(2 points)*:
   - Example: Develop a notebook that solves a classification problem using a dataset.
   - Next step: Publish the notebook on a platform like GitHub or Kaggle.

2. Project code refactored to Python modules placed in selected folders*(4 points)*:
(reference: 3. Modularization/2. ML Project)
   - Example: Refactor the code from the Jupyter notebook into modular Python files.
   - Next step: Organize the code into separate folders based on functionality, such as data preprocessing, model training, and evaluation.

3. Project running as an ML pipeline using Kedro*(7 points)*:
(reference: 4. ML Pipeline with Kedro)
   - Example: Use Kedro, an open-source Python framework, to create a scalable pipeline for data processing, model training, and evaluation.
   - Next step: Define the nodes and data dependencies in the Kedro pipeline and run it to ensure it functions correctly.

4. Implemented experiment tracking system*(7 points)*:
(reference: 5. Experiment tracking)
   - Example: Integrate a tool like MLflow or Neptune to track experiments, parameters, and metrics.(W&B is another tool that can be used for this purpose)
   - Next step: Log the experiments, track model versions, and record important metrics during the pipeline execution.

5. Model trained using an AutoML solution*(5 points)*:
(reference: 8. AutoML)
   - Example: Utilize an AutoML platform like H2O.ai or Google Cloud AutoML to automatically search for the best model.
   - Next step: Extract the best model from the AutoML process and save it for deployment.

6. Identified optimal features (feature engineering), algorithms, and their hyperparameters *(5 points)*:
(reference: 6. Detect Data Drift with TFDV/7. Detect Model Drift)
- Example: Perform feature engineering techniques such as one-hot encoding, scaling, or creating new features based on domain knowledge.(Pandas profiling is a great tool for such analysis, you may pick any other suitable tool.)
- Next step: Use techniques like grid search or Bayesian optimization to find the best hyperparameters for the selected algorithms.

7. Model training and deployment pipelines implemented on selected infrastructure*(8 points)*:
(10. Docker Fundamentals/11. Introduction to Kubernetes)
- Example: Set up a cloud-based infrastructure like AWS or GCP to train and deploy models.

OR

- Set up a local server and perform all necessary operation/communications between modules of your project locally.
- Next step: Define the training pipeline to automatically train the model on a regular basis and deploy it to a scalable infrastructure.

8. Implemented a solution enabling monitoring of operation in the production environment (reference: https://grafana.com/docs/phlare/latest/operators-guide/configure-agent/language-support/python/)
(9 points):
- Example: Integrate monitoring tools like Prometheus, Grafana, or ELK stack to track model performance, resource utilization, and anomalies.
- Next step: Configure alerts and notifications for critical events and establish a process for regular monitoring and maintenance.

9. Chosen infrastructure optimization strategy *(4 points)*:
(reference: 9. AI Architecture Design)
- Example: Optimize the infrastructure by considering factors like cost, scalability, and performance. This may involve choosing appropriate instance types, scaling policies, or using serverless computing options.

10. Implemented data versioning OR Implemented a continuous training system *(8 points)*:
(reference: 10. Docker Fundamentals/4. ML Pipeline with Kedro/5. Experiment tracking)
- Example (Data Versioning): Use a data versioning tool like DVC or Pachyderm to track changes and versions of the datasets used for training and evaluation, track changes in the dataset and ensure reproducibility.
- Example (Continuous Training): Set up a system that automatically retrains models with new data as it becomes available, ensuring that the models stay up-to-date and accurate
- Next step: Establish a process for updating the dataset and integrating it into the training pipeline.

11. Deployed Feature Repository OR Implemented Continuous Integration solution (reference: 11. Introduction to Kubernetes)
(9 points):

- Example (Feature Repository): Set up a feature store like Feast or Hopsworks to manage and serve feature data to the model, enabling easy sharing and collaboration among team members.
- Example (Continuous Integration): Implement a CI/CD (Continuous Integration/Continuous Delivery) system using tools like Jenkins, GitLab CI/CD, or GitHub Actions to automate the build, test, and deployment processes.
- Next step: Define the process for updating and serving features to ensure consistent and reliable model inputs.

12. Implemented a model repository OR Implemented a continuous delivery system
(10 points) :
(reference: 12. Kubeflow pipelines)

- Example (Model Repository): Create a model registry using tools like MLflow or Kubeflow to store and version trained models, and associated metadata. This allows easy retrieval, comparison, and deployment of models.(W&B has similar functionality)
- Example (Continuous Delivery): Automate the process of deploying models to production environments using tools like Kubernetes, Docker, or serverless computing platforms.(GitHub Actions has ability to run such tasks)
- Next step: Set up a process for deploying new model versions to production, including testing, validation, and rollback mechanisms.

13. Integrated and developed project documentation and architecture visualization *(4 points)*:

- Example: Create documentation that describes the project's purpose, architecture, data flow, key components, and dependencies.
- Next step: Visualize the project's architecture using tools like diagrams, flowcharts, or interactive dashboards.(Lucidchart, Diagrams.net, or Graphviz.)

14. Final presentations *(10 points)*:

- Example: Prepare a presentation summarizing the project's objectives, methodology, results, and key findings. Create visualizations, performance metrics, and any insights gained during the project, include future steps.
- Next step(beyond course): Deliver the presentation to stakeholders, including team members, management, or clients, to showcase the project's outcomes and solicit feedback.🙂

15. Overall quality of the project.*(10 points)* :

- It includes style, organization of repo, reliability and reusability of a code and project parts. Will be graded on the basis of the materials that you submit as your final project.

Remember that this checklist is a general description, and the specific steps and tools may vary depending on the project aim and your team’s preferences.