

ETN_EliasSteyaert-stage

Elias Steyaert

2024-10-28

Electronic traineeship notebook (ETN) - Elias Steyaert

A logbook that describes what happened on a daily basis.

Link to my github:

<https://github.com/EliasSteyaert/Amsterdam-UMC.git>

Week 1

Monday 28/10/2024

Introduction workplace

Exploring the main project for the following weeks:

- Mil zijn project analyseren/laten werken
- ENCORE article lezen/screenen naar informatie
- Tutorial Python Machine Learning https://www.w3schools.com/python/python_ml_getting_started.asp
- Practical example Python Machine Learning: <https://machinelearningmastery.com/machine-learning-in-python-step-by-step/>
- Research Scikit-learn and the option to use “pipeline”

Preparation for the project:

- Github in orde maken
- Brainstormen over hoe alles verloopt en het aan te pakken

Tuesday 29/10/2024

Exploring the main project:

- Downloading the newest version of htmltools
- Mil zijn project grondig analyseren

Brainstormen hoe data altijd hetzelfde te krijgen:

- CPM_B0_2013_G1_Yes_Male vs B0-2013_G1
- two files vs one file
- which columns are the needed ones

Meeting bijwonen waar project voorgesteld word (Chronische lymfatische leukemie) en ge-brainstormd wordt hoe de gelijkenissen tussen methodes bepaald kan worden (ex-vivo en 3D)

Preparation for the project:

- <https://link.springer.com/article/10.1186/1471-2164-9-S1-S13> A comparative study of different machine learning methods on microarray gene expression data
- <https://bura.brunel.ac.uk/bitstream/2438/3013/1/TanGilbertNZ2003.pdf> <https://bura.brunel.ac.uk/bitstream/2438/3013/1/TanGilbertNZ2003.pdf>
- <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9693112> A Survey of Machine Learning Approaches Applied to Gene Expression Analysis for Cancer Prediction

Wednesday 30/10/2024

Preparation for the project:

- Kijken tutorial “Machine learning for Cancer Prediction with Gene Expression Data | Random Forests | Python Tutorial”
- <https://medium.com/@michaelanekson/bioinformatics-20-myelodysplastic-syndromes-gene-expression-prediction-using-machine-learning-52fb2f9da2f2>
- <https://github.com/vappiah/Machine-Learning-Tutorials/blob/main/notebooks/projects/Cancer%20Prediction%20using%20Gene%20Expression%20Data.ipynb>

Data goed krijgen zodat er machine learning op uitgevoerd kan worden en vlot lukt in de toekomst toe:

- Data in dictionaries zetten om dan later te converteren naar een dataframe
- De juiste kolommen indexeren
- Data van de kolommen met ‘count per million’ uit de dataframe halen, transposen en mergen met de “targets” table.
- De variable “data” bekomen zodat er preprocessing steps op kunnen gebeuren

Thursday 31/10/2024

Preparation for the project:

- Fixing my jupyter notebook, which doesn’t want to work and shows a lot of errors
- opening the notebook through a link in the terminal with its own <https://> url
- <https://ernest-bonat.medium.com/rna-seq-gene-expression-classification-using-machine-learning-algorithms-de862e60bfd0>
- <https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/>
- The challenge of working with imbalanced datasets is that most machine learning techniques will ignore, and in turn have poor performance on, the minority class, although typically it is performance on the minority class that is most important
- Choosing an oversampling technique: SMOTE is primarily made for binary classification, as this is my case, this will be chosen.

Coding the preprocessing part:

- Distribution graph
- Creating X and y and splitting into traing/test sets
- Label Encoder
- Feature Selection
- Min Max Scaler

- PCA
- SMOTE

Friday 1/11/2024 Classification part of the machine learning (Logistic Regression (11)):

- Completing the machine learning with coding a classification technique (which is logistic regression (L1) in this case)
- Classification models testing and looking which parameters make the F1-score go up
- <https://blog.damavis.com/en/machine-learning-with-python-practical-examples/>
- https://www.w3schools.com/python/python_ml_logistic_regression.aspe

Week 2

Monday 04/11/2024

Classification models testing and looking which parameters make the F1-score go up:

- Concluding that the F1 is low (but also the precision and the accuracy) because of the data that is used, not because of the machine learning steps and options.

Cancer Radar seminar

Building a Machine Learning App in Python with Streamlit:

- <https://vitalflux.com/pca-explained-variance-concept-python-example/#more-8433>
- <https://www.youtube.com/watch?v=LJ6DcLGQ4vY>
- <https://discuss.streamlit.io/t/how-to-use-multiple-files-loaded-with-file-uploader/25051>

Tuesday 05/11/2024

Start of making the interactive app:

- What I hardcoded last week for the two tables to transpose and merge together to have one table, now code it so through questions these will be converted to 'data' automatically
- <https://blog.streamlit.io/generative-ai-and-streamlit-a-perfect-match/>

committing and pushing everything to my GitHub

Wednesday 06/11/2024

Reading the data properly and the data cleaning part:

- Trying to convert all the data from two files into one big dataframe called "data"
- Finding 2 new datasets with all the data in 1 file (samples vertical and horizontal)
- Coding so when data of only one file is needed, the machine knows which column is which

Thursday 07/11/2024

Reading the data properly and the data cleaning part:

- Perfecting the code so every possible data type the user uploads can be worked with for the machine learning automatically.
- Creating a data distribution graph

Brainstorming on how to do the following steps as good as possible without backlash

Friday 08/11/2024

Converting the machine learning bit from past week to streamlit friendly code:

- Starting to write the machine learning bit in streamlit
- <https://docs.streamlit.io/develop/api-reference/widgets/st.slider>
- <https://docs.streamlit.io/develop/api-reference/layout/st.sidebar>
- Coding the preprocessing steps (some with radio buttons for when they are not always needed) and adding an error if there goes something wrong

Week 3

Monday 11/11/2024

- Remembrance day

Tuesday 12/11/2024

Finetuning the preprocessing step:

- <https://docs.streamlit.io/develop/api-reference/text/st.divider>
- making sure using SMOTE is an option instead of hardcoding it

Coding both Random Forest as Logistic Regression (L1)

Looking for more options of machine learning bits to expand the information I can give to the user:

- https://github.com/leosimoes/Streamlit-Logistic-Regressor/blob/master/streamlit_app.py
- <https://docs.streamlit.io/develop/api-reference/text/st.markdown>
- <https://www.geeksforgeeks.org/ways-to-visualize-individual-decision-trees-in-a-random-forest/>

Wednesday 13/11/2024

Trying extra visualising techniques:

- trying to add a heatmap and a feature importances table but these were useless

Finding out how the user can add his data to the script to predict the outcome

Thursday 14/11/2024

Having a conversation with the internship mentor:

- Reviewing what I have done so far
- Talking about problems occurred so far
- Debating future plans and what can be done when I don't know what to do next
- Elaboration was given about parts I use/need but don't understand fully (PCA and feature selection)

Expanding the interactive app so the user can also upload data that has to be predicted:

- coding so the 'testing_data' is always the same, whatever data type the user uses
- Performing the same steps as previously did on the machine learning data

Friday 15/11/2024

Encountering a harder problem to solve:

- ValueError: The feature names should match those that were passed during fit. Feature names unseen at fit time
- Trying to solve with numerous solutions
- Getting to know more about the data scaling step so it's easier to debug this issue
- <https://medium.com/@hhuseyincosgun/which-data-scaling-technique-should-i-use-a1615292061e>
- <https://medium.com/@sjacks/feature-transformation-21282d1a3215>
- Min-Max Scaler: Rescales to predetermined range [0–1] Doesn't change distribution's center (doesn't correct skewness) Sensitive to outliers

Week 4

Monday 18/11/2024

Fixing schoolstuff that still had to be done:

- Learning outcome
- TDP
- ETN
- Weekly Annexes

Presentatie over statistische bio-informatica bijwonen

Fixing the error in connection with the MinMaxScaler:

- <https://stackoverflow.com/questions/46062679/right-order-of-doing-feature-selection-pca-and-normalization>
- Using the same Feature selection genes as the training data: didn't work
- The columns of the training data and the columns of the test data wasn't completely aligned, what worked was the following code snippet:

```
selected_features = X_train_selected.columns  
test_data_selected = X_test[selected_features]
```

Instead of

```
test_data_selected = X_test.iloc[:, selected_scores_indices]
```

Looking how to use multiple sites to reduce loading times:

- <https://docs.streamlit.io/get-started/tutorials/create-a-multipage-app>

Tuesday 19/11/2024

Trying to create the multipage app from my interactive app:

- using the following site/tutorial/example to create a multipage app <https://docs.streamlit.io/get-started/tutorials/create-a-multipage-app>
- “Yes, using multiple pages instead of tabs in your Streamlit app improves performance by avoiding repeated computations, data loading, and rendering that occur with tabs”
- <https://discuss.streamlit.io/t/passing-variable-from-one-page-to-other/3127>
- <https://blog.streamlit.io/how-to-build-an-interconnected-multi-page-streamlit-app/>
- <https://discuss.streamlit.io/t/pass-variable-from-one-page-to-another-after-introduction-of-multi-page-apps/31482>
- <https://discuss.streamlit.io/t/multipage-apps-can-the-pages-share-data/60964>

Meeting about the most efficient way to classify biomarkers

Searching how to add an informative button that shows extra information when hovered over:

- <https://docs.streamlit.io/develop/api-reference/widgets/st.button>
- <https://docs.streamlit.io/develop/concepts/design/buttons>

Struggles to push the git branch to the main branch:

- `git pull --no-rebase`

Wednesday 20/11/2024

Creating a volcano plot and dropping data based on the outcome:

- <https://discuss.streamlit.io/t/how-to-display-matplotlib-graphs-in-streamlit-application/35383>
- https://www.reneshbedre.com/blog/volcano.html?utm_content=cmp-true
- <https://thecodingbiologist.com/posts/Making-volcano-plots-in-python-in-Google-Colab>
- Running into an error where my data doesn't recognise the `gene_ids` list to drop after the volcano plot

```
data = data.drop(genes_inside_threshold, axis=1, errors='ignore')
```

Thursday 21/11/2024

Gesprek met stagementor

Fixing the volcano plot:

- slider to adjust volcano plot's thresholds dynamically
- Solving issue that occurs when performing the `'data = data.drop(genes_inside_threshold)'` twice, it couldn't find the matching genes and gave an error
- <https://stackoverflow.com/questions/74968179/session-state-is-reset-in-streamlit-multipage-app>

Friday 22/11/2024

Trying to add an interactive PCA vs PCA plot:

- <https://bioturing.medium.com/principal-component-analysis-explained-simply-894e8f6f4bfb>
- PCA vs PCA plot making it interactive so the user can choose which two PCA's he wants to plot against each other automated colloring, grouping and legend
- <https://machinelearningmastery.com/principal-component-analysis-for-visualization/>
- <https://plotly.com/python/pca-visualization/>
- <https://stackoverflow.com/questions/60049074/how-to-calculate-principal-component-analysis-to-plot-graph-showing-pc1-vs-pc2-u>

Week 5

Monday 25/11/2024

Printing tests in the 4th app to see why it doesn't run my script:

- Fixing the problem which was a typo in a variable

Gathering extra information:

- <https://www.youtube.com/watch?v=wIvxFEMQVwg> Differential expression in Python with pyDESeq2

Making an easy visualization:

- ROC/AUC for Random Forest Classifier
- plotting 'Visualizing Decision Tree' Deciding if it is useless or not

Tuesday 26/11/2024

Plotting Coefficients Visualization:

- Trying to get the gene names under each bar, turned out it represented PCA's.
- Mapping the PCA's back to the genes first

Plotting "Logistic Regression Coefficients Mapped to genes":

- This was way too much information on one graph

Looking for visually more pleasing plots with not way too much information on one graph:

- ROC/AUC for Logistic Regression (L1)

Running into an error:

- IndexError: index 5219 is out of bounds for axis 0 with size 300

Traceback:

```
File "/media/sf_SF/Stage AMC/Amsterdam-UMC/pages/4_Visualizing _classifications.py", line 151, in <lambda>
    aligned_gene_names = [gene_names[i] for i in selected_scores_indices[:coef_length]]
                        ~~~~~~^~^
```

Solved

Wednesday 27/11/2024

Performing the app with the horizontal (multiclass) data:

- seeing some flaws
- creating a ROC/AUC for multiclass data and automating this
- when trying the prediction page for the multiclass data:

```
KeyError: "None of [Index(['ENSG00000240864', 'ENSG00000277301', 'ENSG00000259719',\n 'ENSG0000027596
```

- fixed by correcting the columns that needs to be taken from the new data to match the columns of the training data set

Thursday 28/11/2024

One Vs Rest Visualization:

- <https://stackoverflow.com/questions/56090541/how-to-plot-precision-and-recall-of-multiclass-classifier>
- ROC/AUC, recall-precision curve, looking at the best options and how to fix my ugly outcome of the curves

Googling a way to get an interactive 'i':

- Would be used after all my multiple choices select boxes so users can know what the best settings are for them specifically
- <https://discuss.streamlit.io/t/tooltip-on-st-write-and-st-markdown/41701>
- <https://docs.streamlit.io/develop/api-reference/text/st.markdown>

- <https://discuss.streamlit.io/t/possible-to-show-tooltips-on-hover/3110/4>

Adding the volcano plot code:

- trying to add a feature that plotting, the users sees how many features would be left after removing the insignificant features.
- Mistake in the code: The genes that were predicted to be kept and the amount of genes that were kept weren't even closely the same.
- Debugging: It was clear that the Original system had flaws (it only filtered on P value, not on the logFc).
- Rewriting the code: Originally, I put all the genes that were in the threshold to be put in a list, and drop the data where the columnnames in the data table were corresponding to the genes in the list. New code:

```
genes_outside_threshold = vulcano_plot_data[vulcano_plot_data['outside_threshold']]['Gene Names'].t
    # Identify the first and last columns
    first_column = data.columns[0]
    last_column = data.columns[-1]

    # Retain the first and last columns and the ones in 'genes_inside_threshold'
    columns_to_keep = [first_column] + \
        [col for col in data.columns if col in genes_outside_threshold] + \
        [last_column]

    # Filter the DataFrame to retain only the desired columns
    data = data.loc[:, data.columns.isin(columns_to_keep)]
```

Adjustments for user friendliness:

- making the predicting data page more user friendly
- user is now able to download the results to either CSV or Excel

Friday 29/11/2024

Looking at researchgate articles of logistic regression and there visualization to look for inspiration

styles.py:

- Writing css/html code to be able to make toolboxes

Writing the first toolbox:

- Writing information about the logFc slider and the P-value slider for the user

Week 6

Monday 2/12/2024

Making the layout of the tooltip better:

- https://www.w3schools.com/css/css_margin.asp
- <https://docs.streamlit.io/develop/api-reference/text/st.markdown>
- tooltips bij de radiobuttons proberen krijgen Trying some options, but none of them felt good so I gave up and wrote the explanation of the technique as a “st.write” after the user chooses it.

Adding an extra oversampling technique:

- <https://github.com/stavskal/ADASYN>
- writing an extra oversampling technique: ADASYN

Tuesday 3/12/2024

Toolboxtip for the n_estimators:

- When you lower the amount of trees, you make the model faster to train but may reduce accuracy and stability since fewer votes determine the output.
- When you higher the amount of trees, you increase accuracy and reduce variance by averaging more predictions, but they also make the model slower to train and require more memory.

Extra information about the different type of classification types when clicked upon one:

- <https://stackoverflow.com/questions/2595176/which-machine-learning-classifier-to-choose-in-general>

Searching for more information on SVM and XGBoost:

- <https://www.datacamp.com/tutorial/svm-classification-scikit-learn-python>

Wednesday 4/12/2024

Adding XGBoost and it's own visualizations:

- https://shap.readthedocs.io/en/latest/example_notebooks/overviews/An%20introduction%20to%20explainable%20AI%20with%20Shapley%20values.html

Making a scree plot for the PCA's:

- <https://sanchitamangale12.medium.com/scree-plot-733ed72c8608>
- <https://plotly.com/python/pca-visualization/>

Thursday 5/12/2024

Adjusting the visualization of logistic regression:

- added a 'top gene contributions after PCA'
- When plotting the contribution value of the genes of a chosen PCA, the gene name is shown upon hovering on the bar of the barplot
- <https://github.com/ezgisubasi/breast-cancer-gene-expression>

gesprek met stagementor

Googling more databases and information on PCA:

- https://github.com/shameigirls/Machine-Learning-on-Gene-Expression-Data-for-Staging-Classification-of-Breast-Cancer/blob/master/TCGA-BRCA_classificationML.ipynb
- <https://github.com/gksr15/Detecting-Presence-of-Huntington-Disease-Using-Machine-Learning-Models-on-Gene-Expression-Data/blob/main/README.md>
- https://www.youtube.com/watch?v=HMOI_lkzW08 PCA explained in 5 minutes
- <https://www.youtube.com/watch?v=FgakZw6K1QQ> StatQuest: Principal Component Analysis (PCA), Step-by-Step

Friday 6/12/2024

Reorganising the pages to maximise the loading speed:

- Making a page for preprocessing on its own
- Making the machine learning and visualization one big page as this loads faster than preprocessing + machine learning

Trying to get a next and previous button in streamlit:

- The first code I found was to go next and previous on the same page: <https://discuss.streamlit.io/t/next-and-previous-buttons/54353/2>
- The next problem was that when using html to go through the next page, the whole app reloads and so loses the `st.session_state` variables <https://discuss.streamlit.io/t/navigate-multipage-app-with-buttons-instead-of-sidebar/27986>
- Button is there now but fails to go from one page to the other, it just stays on the same page.

Week 7

Monday 9/12/2024

Last attempt to make the “Next” button work based on a new thread I found:

- <https://discuss.streamlit.io/t/navigate-multipage-app-with-buttons-instead-of-sidebar/27986/6> from `streamlit_extras.switch_page_button` import `switch_page`

Presentation about the importance of adding ADA (antidrug antibodies) by anti-TNF-alpha therapy

Making the plot of PCA and the right amount of PCA's for the analysis better so the elbow point is shown

Looking into how to add cross validation into my app

Tuesday 10/12/2024

Going over the code so that I'm positive that there aren't any mistakes there

```
MI=mutual_info_classif(X_train,y_train)
selected_scores_indices=np.argsort(MI)[:,-1][0:n_features]
X_train_selected=X_train.iloc[:, selected_scores_indices]
X_test_selected=X_test.iloc[:, selected_scores_indices]
# Evaluate with cross-validation
model = RandomForestClassifier(random_state=42)
scores = cross_val_score(model, X_train_selected, y_train, cv=5, scoring='f1_weighted')
st.write(f"Mean F1 Score with top 300 features: {np.mean(scores):.4f}")
- Tooltip for the cross-validation of the Mutual information step
- Cleaning up the visual part of the tooltips
- Make columns out of the extra information given when having to choose with radio buttons
```

Asserting cross-validation for the Mutual information step:

Wednesday 11/12/2024

Improving the last page:

- Selectboxes in the data predicting instead of text input boxes
- adding labels/classes into the downloaded data

Improving the overall lay-out:

- trying to get more line breaks in the app so it's cleaner for the user
- Cleaning up the uploading part, volcano part and the machine learning part so its way more user friendly and cleaner to look at

Thursday 12/12/2024

Trying to figure out how to get the visualization better:

- Adjusting the logistic regression
- Start of the adjustment of Random Forest

Gesprek met stagementor (Aldo)

Attend a PhD defense and reception

Friday 13/12/2024

Logistic regression:

- When having multiclass data, letting the user choose it's class to show which genes contributes the most
- <https://medium.com/@penpencil.blr/data-imbalance-how-is-adasyndifferent-from-smote-f4eba54867ab>

Week 8

Monday 16/12/2024

Random Forest optimization:

- Trying to get the Random forest visualization work better
- Let the user ask which class he wants to inspect to see the highest feature importance scores

Kerstdrink

Tuesday 17/12/2024

Trying to end the lose ends:

- Trying to finish the random forest visualization
- Trying to get the same for XGBoost but keeping on failing
- Doing it right for the OneVsRest Classifier

Wednesday 18/12/2024

Trying to end the lose ends:

- Trying to have the multiclass visualization of Random Forest good so the user can choose a class and it shows the most important features for that class, but failing in it
- Trying to get the same for XGBoost but keeping on failing

Thursday 19/12/2024

Trying to end the lose ends:

- Trying to have the multiclass visualization of Random Forest good so the user can choose a class and it shows the most important features for that class, but failing in it
- Deleting the not working feature and concluding that when someone wants to see it for each option, he needs to use the One Vs Rest classifier
- Trying to get the same for XGBoost but keeping so also deleting the feature here

Friday 20/12/2024

Adding a good home page text

Converting my github into the ENCORE guidelines

- https://github.com/EliasSteyaert/Multipage_App_Machine_Learning/tree/main
- Having a README file, my computer specs, adding a requirements.txt file

Finishing school obligations

Trainee Documentation Plan (TDP)

Tentative Planning

Tasks to Be Performed During the Traineeship The main task of this internship is to create a fully functional interactive application. This app will be developed using Python and is intended to perform machine learning on patient cancer data. The key steps include:

1. Data Identification:

Regardless of the data the user uploads, the script must identify:

- Columns containing sample IDs.
- Columns indicating illness status.
- Columns containing gene expressions.

2. Data Preprocessing:

Perform various preprocessing steps, such as:

- Data scaling.
- Oversampling smaller data groups to balance distributions.
- Feature selection.
- Principal Component Analysis (PCA).

3. Model Creation:

Enable the creation of a machine learning model using various classification techniques. Users can select their preferred classification method.

4. Prediction on New Data:

Allow users to upload new patient data. The model should predict whether the patient(s) are affected by the types of cancer it has been trained on.

Data Management

FAIR Principles The FAIR principles guide the storage and structuring of data. Below is how each principle is applied:

- **Findability:**
 - The data used for creating the app was provided by Amsterdam UMC and cannot be found online (yet). It consists of:
 - * A table with patient information.
 - * A table with gene information and gene expression data for each patient.
 - Related documentation and code are available on GitHub: Amsterdam UMC GitHub Repository.
 - Data intended for use with the app must be in `.txt` or `.csv` format and must include:
 - * Sample IDs.
 - * Gene IDs.
 - * Gene expressions.
 - * Illness status.
- **Accessibility:**
 - The original data is not publicly accessible. It is intended for use by doctors/researchers with their own gene expression data.
 - The app's code and a README explaining its use are available on GitHub. Users can clone the repository and run the app with a single CLI command.
- **Interoperability:**
 - The app accepts data in `.csv` or `.txt` format.
 - It prompts users to specify:
 - * File type.
 - * Separator used.
 - * Decimal delimiter.
 - * Presence of headers.
- **Reproducibility:**
 - The app can be reproduced using its GitHub repository.
 - Users can clone the repository into their virtual environment, use it as-is, or modify it as needed.

Traceability of Steps and Methods

Documentation of Project Steps

- An RMarkdown file in the GitHub repository serves as the **Electronic Traineeship Notebook (ETN)**.
- The ETN includes descriptions of daily actions and progress.

Version Control of Code

Git Repository

- The code is stored in a dedicated GitHub repository for this project: Multipage App for Machine Learning Repository.

Access for Supervisors

- The repository is set to **public** visibility, ensuring both internal and external supervisors can access the code.

The Learning Outcome (LO)

Development Goal Trying to learn/become better at using the coding language ‘Python’.

We want to achieve this goal through coding an interactive application that predicts cancer outcomes based on machine learning.

The SMART Principles Can Be Applied Here **Specific:** Develop an interactive application using Python and Streamlit to facilitate machine learning on gene expression data. The app will include functionalities for data preprocessing, model training, evaluation, and visualization.

Measurable: By the end of the internship, the application should operate without errors. It must deliver clean, user-friendly visualizations, and produce reliable results with high accuracy when provided with appropriate data.

Achievable: Given that I work with Python daily as part of my internship, this goal is realistic and within reach with consistent effort and focus.

Relevant: In the context of the traineeship, this learning outcome is relevant. This project is also very relevant when you look at the education I follow: ‘Bioinformatics’.

Time-bound: The app must be fully functional and error-free before the conclusion of my internship. To ensure enough time for testing and troubleshooting, an internal deadline will be set well in advance of the final due date.

Development Activity I will be working with Python to code an interactive app that performs machine learning on cancer data and gives the user an easy interface to choose what steps they want to be performed on their data. Python will also be used to let the user upload their data files, and through the interface, the Python script will transcribe the data into a usable dataframe to perform the machine learning on.

Desired Results Being able to make a fully automated interactive app where the user does not have to do a lot of manual work, except for uploading their files and choosing which options they want to use for its machine learning. The application should be able to run completely and perform all the necessary tasks without running into errors.

Schedule Throughout the whole internship, I will work with Python and create/perfect the interactive app daily. The application will be improved daily and tested to ensure that the adjustments have the desired effect.

Necessary Support and Facilities Normally, all this should be done within the traineeship and its 250 hours. Nothing should be bought for this goal to be achieved. There is, for example, no need for cloud space or an instance.