# Python Machine Learning Labs - End-to-End Machine Learning Pipeline

**Course Information:**

- **Course:** Python Machine Learning Labs

- **Instructor:** Assan Sanogo

- **Professor:** Hanna Abi Akl

- **Students:** Yaakoub El Guezdar, Kateryna Draganova

**Dataset Source:**

https://www.transferxl.com/download/08vSFcz3B7fXPr

**Overview:**

This README provides a step-by-step guide to the development of an end-to-end Machine Learning Pipeline, conducted as part of the Python Machine Learning Labs course. The pipeline includes data extraction, exploratory data analysis (EDA), feature engineering, and the implementation and evaluation of various machine learning models.

1. Data Extraction and Exploratory Data Analysis (EDA):

- Data Extraction:

- Extracted data from the provided source and loaded it into a pandas DataFrame.

- Exploratory Data Analysis (EDA):

- Conducted a detailed exploration of the dataset, addressing NaN values and filtering essays by type.

- Identified key features for model training, including:

1. Essays Length (Number of Words)

2. Extracted Tags and Tag Frequency

3. Ratio of Sentences to Paragraphs

4. Sentence Length Variation

5. Transition Words Usage

6. Corrected Type Token Ratio (CTTR)

7. Number of Sentences

8. Word Frequencies and Most Repeated Words

9. Flesch-reading Ease Score

10. Gunning Fog Index

11. Misspelling Rate

12. Frequency of Adjectives, Adverbs, Nouns, and Verbs

2. Feature Engineering:

- Developed code for extensive feature engineering using natural language processing tools such as spaCy.

- Enriched the dataset with calculated features, creating the `Valid\_Set` DataFrame.

3. Data Preprocessing:

- Utilized the `MinMaxScaler` from scikit-learn to preprocess features in the `Valid\_Set` DataFrame.

- Normalized predictive features, ensuring consistent and comparable magnitudes.

4. Multi-Output Regression Model:

- Implemented a multi-output regression model using scikit-learn to predict multiple target scores.

- Normalized features using Min-Max scaling and split the dataset into training and testing sets.

- Initialized a `MultiOutputRegressor` with a linear regression model as the base estimator.

- Trained the model, made predictions, and evaluated its performance.

5. Evaluation of Regression Models:

- Employed various regression models, including Linear Regression, Random Forest Regressor, Support Vector Regression (SVR), and Multi-layer Perceptron (MLP).

- Incorporated pipelines with StandardScaler for SVR and MLP models to ensure optimal performance.

- Trained models, made predictions, and calculated Mean Squared Error (MSE) and R-squared (R²) scores for evaluation.

6. Model Assessment and Comparison:

- Assessed each model's performance based on MSE and R² scores.

- Concluded that RandomForestRegressor appeared to be the best model among the three, considering positive R² and a relatively low MSE.

7. Conclusion:

- Highlighted the importance of model selection based on evaluation metrics.

- Acknowledged potential issues, such as negative R² indicating model fit problems.

This README serves as a comprehensive guide to the development and evaluation of the Machine Learning Pipeline, providing insights into the process and outcomes of the project.