CS-3410-1: Introduction to Machine Learning

Assignment 1

Datasets

You are required to work with **three datasets of increasing complexity**. All three are **regression tasks**.

1. Life Expectancy (WHO)

Type: Regression Task
Description: Includes health, economic, and demographic data from WHO member states.

Training Data: Uploaded on Classroom

Target Feature: `Life expectancy` (years)
Test Data: Will not be provided to students.

---

2. Laptop Price Prediction

Type: Regression Task
Description: Includes laptop specifications such as brand, CPU, RAM, storage, GPU, operating system, and other features.

Training Data: Uploaded on Classroom

Target Feature: `Price` of the laptop (continuous)
Test Data: Will not be provided to students.

---

3. Large Retail Dataset

Type: Regression Task
Description: Retail transaction dataset with 70+ customer, product, and transaction-related variables.

Training Data: Uploaded on Classroom

Target Feature: `avg_purchase_value` per customer
Test Data: Will not be provided to students.

Instructions

1. Implement Algorithms from Scratch

- **No Machine Learning Libraries**: You must code all algorithms manually. The use of machine learning libraries such as scikit-learn, TensorFlow, Keras, PyTorch, etc., is strictly prohibited.

- **Allowed Libraries**: You may use libraries for basic data handling and mathematical computations such as NumPy and Pandas. Visualization libraries (Matplotlib/Seaborn) are also permitted.

---

2. Project Structure and Version Control

Directory Structure

Organize your project using the following directory structure for each task:

```
FirstName_LastName_A1/
├── report.pdf
├── life_expectancy_task/
|   ├── data/
|   |   └── train_data.csv
|   ├── notebooks/
|   |   └── data_exploration.ipynb
|   ├── models/
|   |   ├── regression_model1.pkl
|   |   ├── regression_model2.pkl
|   |   ├── regression_model3.pkl
|   |   └── regression_model_final.pkl
|   ├── src/
|   |   ├── data_preprocessing.py
|   |   ├── train_model.py
|   |   └── predict.py
|   ├── results/
```

```
|   |   ├── train_metrics.txt
|   |   └── train_predictions.csv
|   ├── requirements.txt
|   ├── .git/
|   └── .gitignore
├── laptop_price_task/
|   ├── data/
|   |   └── train_data.csv
|   ├── notebooks/
|   |   └── data_exploration.ipynb
|   ├── models/
|   |   ├── regression_model1.pkl
|   |   ├── regression_model2.pkl
|   |   ├── regression_model3.pkl
|   |   └── regression_model_final.pkl
|   ├── src/
|   |   ├── data_preprocessing.py
|   |   ├── train_model.py
|   |   └── predict.py
|   ├── results/
|   |   ├── train_metrics.txt
|   |   └── train_predictions.csv
|   ├── requirements.txt
|   ├── .git/
|   └── .gitignore
└── retail_task/
    ├── data/
    |   └── train_data.csv
    ├── notebooks/
    |   └── data_exploration.ipynb
    ├── models/
    |   ├── regression_model1.pkl
    |   ├── regression_model2.pkl
    |   ├── regression_model3.pkl
    |   └── regression_model_final.pkl
    ├── src/
    |   ├── data_preprocessing.py
    |   ├── train_model.py
    |   └── predict.py
    ├── results/
    |   ├── train_metrics.txt
    |   └── train_predictions.csv
```

```
├── requirements.txt
├── .git/
└── .gitignore
```

Parent Directories: The three main tasks are separated into `life_expectancy_task`, `laptop_price_task`, and `retail_task`, each with the same internal structure.

Model Saving:

- Save your trained models in the `models/` directory.
- Models should be saved using Python's `pickle` module.

Version Control with Git

- **Separate Repositories:** Initialize a separate Git repository inside each task directory.
- **Final Code:** The final, polished code should be in the main branch of each repository.
- **Commit Messages:** Write clear and descriptive commit messages that reflect the changes made.
- **Commit History:** We will review your commit history to assess time and effort spent on experimentation.
- **Gitignore**: Make sure to add `data` to `.gitignore`. Do NOT commit the `data` folder to git.

---

3. Model Training, Saving, and Evaluation

Data Preprocessing

- Handle missing values, encode categorical variables, and perform any necessary preprocessing.
- Document these steps in your code and report.

Algorithm Implementation

- All three tasks are **regression tasks**.
- You may implement multiple regression models (e.g., Linear Regression, Polynomial Regression, Ridge/Lasso). Save the best as `regression_model_final.pkl` and the others as `regression_model1.pkl`, `regression_model2.pkl` and so on.

Training

- Train your models using the training data provided.
- Save the trained models in the specified format within the `models/` directory.

---

5. Evaluation Script (`predict.py`) and Standard Output

Purpose

- Loads the saved model and evaluates it on a dataset.
- Generates predictions and outputs metrics in a standardized format.

Usage

```
python src/predict.py \
  --model_path models/regression_model_final.pkl \
  --data_path data/train_data.csv \
  --metrics_output_path results/train_metrics.txt \
  --predictions_output_path results/train_predictions.csv
```

Arguments

- `--model_path`: Path to the saved model file.
- `--data_path`: Path to the data CSV file that includes features and true labels.
- `--metrics_output_path`: Path where evaluation metrics will be saved.
- `--predictions_output_path`: Path where predictions will be saved.

Standard Structure for `train_predictions.csv`

- Single column CSV file with predictions.
- No header.
- First row = first prediction.

Standard Structure for `train_metrics.txt`

For all regression tasks, use the exact following format:

```
Regression Metrics:
Mean Squared Error (MSE): <value>
Root Mean Squared Error (RMSE): <value>
R-squared (R²) Score: <value>
```

- Round values to **two decimal places**.
- Follow the format strictly (labels, order, spacing).

---

6. Report (`report.pdf`)

Your report must include the following sections:

1. **Introduction** – Briefly describe the dataset and problem.
2. **Methodology** – Algorithms and preprocessing techniques used.
3. **Experimentation** – Different model variations and hyperparameters tested.
4. **Results** – Metrics, tables, and graphs.
5. **Explanation** – Insights / Explanation of the model in terms of relation between predictor variables and the predicted variable. This will be a function of regularization.
6. **Challenges** – Difficulties faced and how you addressed them.

7. **Conclusion** – Summary and possible improvements.
8. **References** – Any references used.

---

7. Grading Criteria

- Algorithm implementation correctness.
- Code quality: modularity, readability, documentation.
- Model performance on regression tasks.
- Project structure adherence.
- Report quality: clarity, analysis, insights.
- Git usage: frequency and clarity of commits.
- Compliance: following restrictions (no ML libraries).

---

8. Submission Guidelines

- Submit a `.zip` file named:

```
FirstName_LastName_A1.zip
```

- Do **not** include raw CSV files in submission i.e. make sure to remove the files from `data` before zipping.
- Any runtime error will result in loss of marks.
- Code similarity and AI detection tests will be applied.
- Use of AI tools (ChatGPT, Claude, etc.) is strictly prohibited.