

GitHub Profile Guidelines

Dr. Sajil C. K., Knowledge Officer, ICT Academy of Kerala

This document is to guide you through the important things to be considered when maintaining your GitHub account. As you might know, your GitHub account will be primary profile recruiters reviewing your competency/maturity as a data scientist/machine learning engineer/programmer in general. I am specifically helping you to avoid mistakes that signal out loud to the recruiter that you are just a beginner in this field. Hence it is very important to keep it up to professional standards. I hope that these guidelines will act as reference for you.

Please understand that recruiting mostly works on trust. It is the trust of the recruiter on your abilities to solve business problems for them that gets you hired. Everything we do such as portfolio projects, marks, certifications etc are for building that trust. Because there is no reason for a stranger to trust on your abilities unless you are recommended to them by someone with strong credibility (existing employee/premium institutes like IITs).

1. The following things are important to have on the GitHub profile page.
 - 1.1. Name
 - 1.2. Domain/Designation/Tag Line
 - 1.3. Profile Picture/Avatar: this shows you are a real person and builds trust of the recruiter on you.
 - 1.4. Location and TimeZone
 - 1.5. Email, LinkedIn Profile, and Personal Website (if any)
 - 1.6. Profile Readme (Optional)
 - 1.7. Pinned Repositories: There is a provision to Pin up to six repositories, but only pin those which are complete. It is fine to have only 1 completed repository to be Pinned rather than 6 low quality repositories.
 - 1.8. Each Pinned repository should have the description information filled in two-three sentences.
 - 1.9. It is not the number of repositories but the quality of repositories matter
 - 1.10. Contribution heatmap: make sure you work on some projects or make some updations to your GitHub profile/repositories on a day to day basis. Your contribution heatmap should look dense rather than sparse. Please do not use automated scripts to make heatmap appear dense.

2. Each individual repository should have the following things to be checked and done
 - 2.1. Repository description
 - 2.2. Tags
 - 2.3. Well documented readme. The readme should contain the following
 - 2.3.1. The project title
 - 2.3.2. About the Project/Summary
 - 2.3.3. A GIF image showing the App/Script in action (optional)
 - 2.3.4. Installation/Setup/How to run section: Setup or running instructions should be in code cell blocks
 - 2.4. License
 - 2.5. GitIgnore file to untrack unnecessary and cache files (Do not put things like Pycache to the repository)
 - 2.6. Venv/Conda requirements file/any other dependency management method
 - 2.7. Do not leave a repository only with a Jupyter notebook alone. If you do, it tells you are just a beginner. The repository should be accompanied by additional code/scripts. The Jupyter notebooks can be placed inside a folder named notebooks.
 - 2.8. If you have any reports, it should be inside a folder named reports.
 - 2.9. The folder structure (generic) for a data science project can be as follows.

```

project_name/
|— data/          # Raw and processed data (usually not
committed to Git)
| |— raw/         # Original, immutable data
| |— processed/   # Cleaned and preprocessed data
| |— external/    # Data from third-party sources
| |— interim/     # Intermediate data files
|
|— notebooks/     # Jupyter notebooks for EDA and
experimentation
| |— 01_exploratory.ipynb
| |— 02_feature_engineering.ipynb
| |— 03_model_training.ipynb
| |— 04_model_evaluation.ipynb
|
|— src/           # Source code for the project
| |— __init__.py  # Makes src a Python module
| |— data/        # Scripts for data loading and preprocessing
| | |— load_data.py
| | |— preprocess.py

```

```

├── augment.py
├── features/      # Feature engineering scripts
│   ├── build_features.py
├── models/       # Model training and evaluation scripts
│   ├── train.py
│   ├── predict.py
│   └── evaluate.py
├── visualization/ # Scripts for generating visualizations
│   ├── eda.py
│   └── plot_results.py
├── utils/        # Helper functions
│   ├── config.py
│   ├── logger.py
│   └── metrics.py
├── configs/      # Configuration files (YAML, JSON, INI)
│   ├── config.yaml
│   └── model_params.json
├── tests/        # Unit tests for scripts
│   ├── test_data.py
│   ├── test_models.py
│   └── test_utils.py
├── reports/      # Final reports, figures, and analysis
│   ├── figures/  # Visualizations
│   └── results_summary.md # Summary of findings
├── logs/         # Logs from training, testing, and debugging
├── models/       # Saved model files and artifacts
│   ├── trained_model.pkl
│   └── model_checkpoint.pth
├── scripts/      # Standalone scripts for automation
│   ├── train_model.sh
│   └── run_pipeline.py
├── environment.yml # Conda environment file
├── requirements.txt # Python dependencies
├── setup.py       # Setup script for packaging
└── README.md      # Overview and usage instructions

```

— LICENSE	# License for the project
— .gitignore	# Files to ignore in Git
— .dvc/	# DVC (Data Version Control) files (if used)
— Makefile	# Automate tasks (if needed)