Giraldo Nainggolan

https://portofoliokreatif.blogspot.com/

Portofolio

2025

Information Systems

Giraldo Nainggolan

IINNOVATIVE IT PROGRAMMER

About Me

I am an Information Systems student with an interest in technology and innovation. Experienced in website development, UI/UX, mobile application, data analysis, and project management, I focus on creating digital solutions that are secure, efficient, and impactful.

Skills

- Python
- Javascript
- Java / Kotlin
- Wordpress
- PHP
- Laravel
- SQL

Contact

082383077087

https://giraldonainggolan.blogspot.com/

PERUM TGR BLOCK I 47/16

Education

senior high school
SMAKN SAMOSIR 2018 - 2021

Information Systems Students
Trunojoyo University 2022 - 2026

During my studies, I have been actively involved in various IT trainings and projects, including software redesign, website and Android application debugging, and user convenience implementation. I am active in the Information Systems Association organization, Google Developer Campus, ITC, and several other additional activities.

Experience

Project Management - Surabaya
Reka Village April 2024 - June 2024
Explore my Real Estate Website project built with WordPress CMS. Featuring responsive design, SEO optimization, and advanced property search for seamless browsing.

Product Developmen - Surabaya
UPN East Java June 2024 - Agustus 2024
Discover my FarmBot project: an innovative solution combining automation and precision farming to revolutionize sustainable agriculture and boost efficiency.

Software Staff
Bangkalan Legislative Council Sep 2024 - Nov 2024
Tasked with designing a cross-platform application for the Bangkalan Regional People's Representative Council (DPRD) to facilitate efficient correspondence and communication.

Audit & Corporate Governance
PT Bukit Darmo.Tbk Oct 2024 - Dec 2024
Data entry in the IT governance audit using COBIT 19 involves collecting, verifying, and recording information related to IT Processes, GAMO, and corporate compliance over a three-month period.

Production Planning and Inventory Control
PT Sebastian Jaya Metal Jan 2025 - July 2025
In this role, I am responsible for production planning, inventory control, as well as ensuring a smooth production process in accordance with the demand and availability of raw materials.

2025

PROJECT

DATA SPECIALIST

Data Science, Analytics, and Computer Vision for Advanced Insights and AI Solutions

Disusun oleh
Giraldo Naiggolan

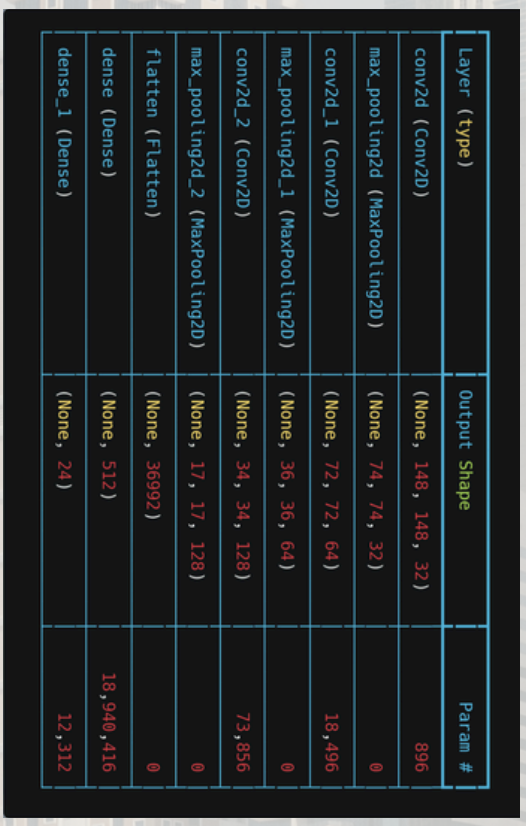
Fruit Classification using CNN and Image Processin

Dataset properties:

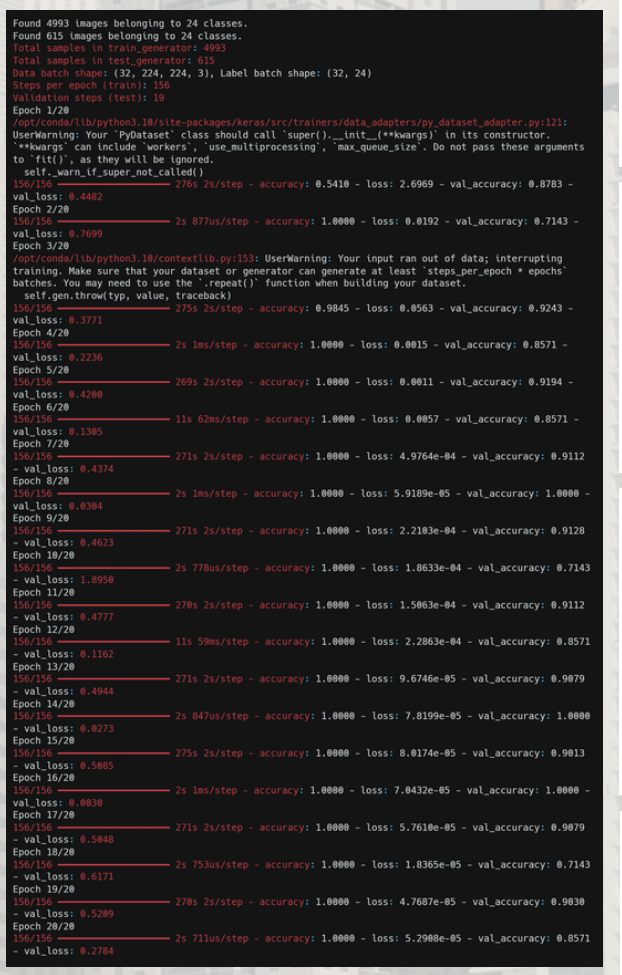
The total number of images: 94110. Only images scaled to 100x100 pixels are counted here. Training set size: 70491 images (one object per image). Test set size: 23619 images (one object per image). The number of classes: 141 (fruits and vegetables). Image size in the largest dataset version: 100x100 pixels. Input (1.01 GB)

Technologies Used:

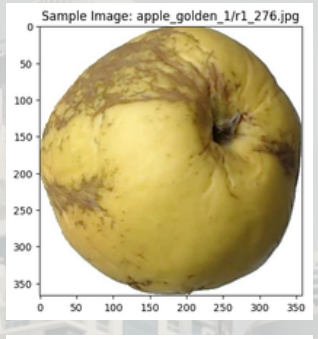
os, cv2 (OpenCV), matplotlib, seaborn, numpy, pandas, torch (PyTorch), torchvision, sklearn.model_selection (train_test_split), tensorflow.keras, tensorflow.keras.optimizers (Adam), tensorflow.keras.callbacks (EarlyStopping), sklearn.metrics (classification_report, confusion_matrix), mpl_toolkits.mplot3d.Axes3D, sklearn.preprocessing.StandardScaler

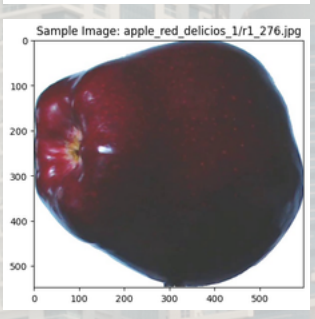


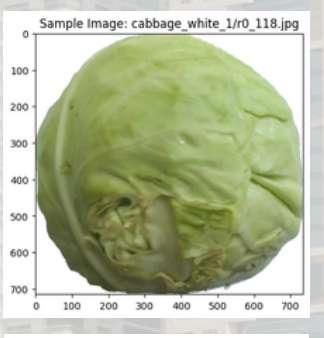
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 72, 72, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_2 (Conv2D)	(None, 34, 34, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 128)	0
flatten (Flatten)	(None, 36992)	0
dense (Dense)	(None, 512)	18,940,416
dense_1 (Dense)	(None, 24)	12,312

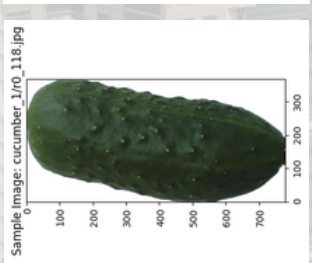


```
Found 4993 images belonging to 24 classes.
Found 615 images belonging to 24 classes.
Total samples in train_generator: 4993
Total samples in test_generator: 615
Data batch shape: (32, 224, 224, 3), Label batch shape: (32, 24)
Steps per epoch (train): 156
Validation steps (test): 19
Epoch 1/20
/opt/conda/lib/python3.10/site-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121:
UserWarning: Your `PyDataset` class should call `super().__init__(**kwargs)` in its constructor.
`**kwargs` can include `workers`, `use_multiprocessing`, `max_queue_size`. Do not pass these arguments
to `fit()`, as they will be ignored.
  self._warn_if_super_not_called()
156/156 ━━━━━━━━━━━━━━━━━━━━ 276s 2s/step - accuracy: 0.5410 - loss: 2.6969 - val_accuracy: 0.8783 -
val_loss: 0.4482
Epoch 2/20
156/156 ━━━━━━━━━━━━━━━━━━━━ 2s 877us/step - accuracy: 1.0000 - loss: 0.0192 - val_accuracy: 0.7143 -
val_loss: 0.7699
Epoch 3/20
/opt/conda/lib/python3.10/contextlib.py:153: UserWarning: Your input ran out of data; interrupting
training. Make sure that your dataset or generator can generate at least `steps_per_epoch * epochs`
batches. You may need to use the `.repeat()` function when building your dataset.
  self.gen.throw(typ, value, traceback)
156/156 ━━━━━━━━━━━━━━━━━━━━ 275s 2s/step - accuracy: 0.9845 - loss: 0.0563 - val_accuracy: 0.9243 -
val_loss: 0.3771
Epoch 4/20
156/156 ━━━━━━━━━━━━━━━━━━━━ 2s 1ms/step - accuracy: 1.0000 - loss: 0.0015 - val_accuracy: 0.8571 -
val_loss: 0.2236
Epoch 5/20
156/156 ━━━━━━━━━━━━━━━━━━━━ 269s 2s/step - accuracy: 1.0000 - loss: 0.0011 - val_accuracy: 0.9194 -
val_loss: 0.4200
Epoch 6/20
156/156 ━━━━━━━━━━━━━━━━━━━━ 11s 62ms/step - accuracy: 1.0000 - loss: 0.0057 - val_accuracy: 0.8571 -
val_loss: 0.1305
Epoch 7/20
156/156 ━━━━━━━━━━━━━━━━━━━━ 271s 2s/step - accuracy: 1.0000 - loss: 4.9764e-04 - val_accuracy: 0.9112
- val_loss: 0.4374
Epoch 8/20
156/156 ━━━━━━━━━━━━━━━━━━━━ 2s 1ms/step - accuracy: 1.0000 - loss: 5.9189e-05 - val_accuracy: 1.0000 -
val_loss: 0.0304
Epoch 9/20
156/156 ━━━━━━━━━━━━━━━━━━━━ 271s 2s/step - accuracy: 1.0000 - loss: 2.2103e-04 - val_accuracy: 0.9128
- val_loss: 0.4623
Epoch 10/20
156/156 ━━━━━━━━━━━━━━━━━━━━ 2s 778us/step - accuracy: 1.0000 - loss: 1.8633e-04 - val_accuracy: 0.7143
- val_loss: 1.8950
Epoch 11/20
156/156 ━━━━━━━━━━━━━━━━━━━━ 270s 2s/step - accuracy: 1.0000 - loss: 1.5063e-04 - val_accuracy: 0.9112
- val_loss: 0.4777
Epoch 12/20
156/156 ━━━━━━━━━━━━━━━━━━━━ 11s 59ms/step - accuracy: 1.0000 - loss: 2.2863e-04 - val_accuracy: 0.8571
- val_loss: 0.1162
Epoch 13/20
156/156 ━━━━━━━━━━━━━━━━━━━━ 271s 2s/step - accuracy: 1.0000 - loss: 9.6746e-05 - val_accuracy: 0.9079
- val_loss: 0.4944
Epoch 14/20
156/156 ━━━━━━━━━━━━━━━━━━━━ 2s 847us/step - accuracy: 1.0000 - loss: 7.8199e-05 - val_accuracy: 1.0000
- val_loss: 0.0273
Epoch 15/20
156/156 ━━━━━━━━━━━━━━━━━━━━ 275s 2s/step - accuracy: 1.0000 - loss: 8.0174e-05 - val_accuracy: 0.9013
- val_loss: 0.5085
Epoch 16/20
156/156 ━━━━━━━━━━━━━━━━━━━━ 2s 1ms/step - accuracy: 1.0000 - loss: 7.0432e-05 - val_accuracy: 1.0000 -
val_loss: 0.0030
Epoch 17/20
156/156 ━━━━━━━━━━━━━━━━━━━━ 271s 2s/step - accuracy: 1.0000 - loss: 5.7610e-05 - val_accuracy: 0.9079
- val_loss: 0.5048
Epoch 18/20
156/156 ━━━━━━━━━━━━━━━━━━━━ 2s 753us/step - accuracy: 1.0000 - loss: 1.8365e-05 - val_accuracy: 0.7143
- val_loss: 0.6171
Epoch 19/20
156/156 ━━━━━━━━━━━━━━━━━━━━ 270s 2s/step - accuracy: 1.0000 - loss: 4.7687e-05 - val_accuracy: 0.9030
- val_loss: 0.5209
Epoch 20/20
156/156 ━━━━━━━━━━━━━━━━━━━━ 2s 711us/step - accuracy: 1.0000 - loss: 5.2908e-05 - val_accuracy: 0.8571
- val_loss: 0.2784
```









https://www.kaggle.com/code/giraldonainggolan/fruit-classification-using-cnn-and-image-processin

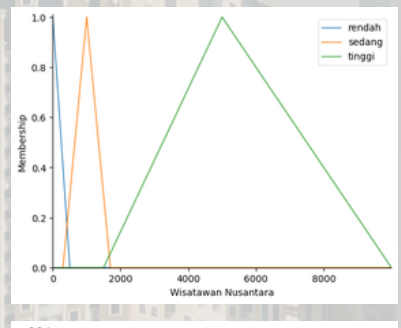
Online shopping-KNN-ROC

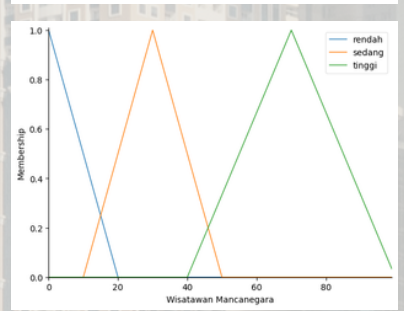
Dataset properties:

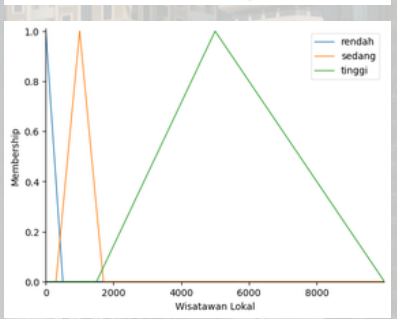
The dataset provided contains data about several tourist attractions along with the predicted number of tourists who will visit these places. For example, Mutiara Beach is predicted to attract 50,000 tourists, while Maobu Beach has a predicted number of visitors ranging from 16,695 to 17,079 tourists, and Bone Labunta is expected to invite between 16,708 to 21,843 tourists.

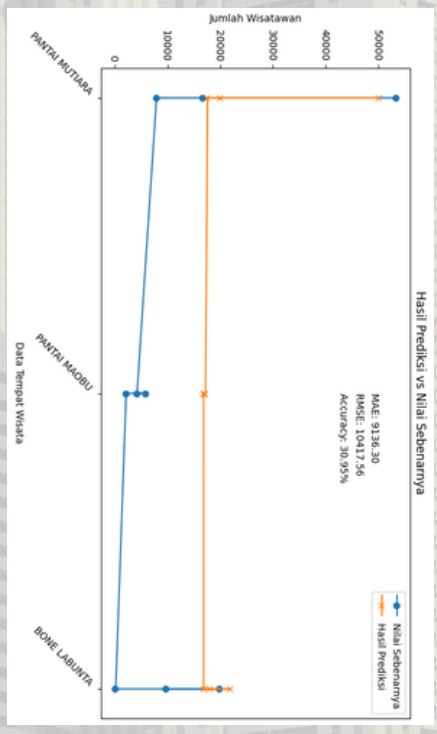
Technologies Used:

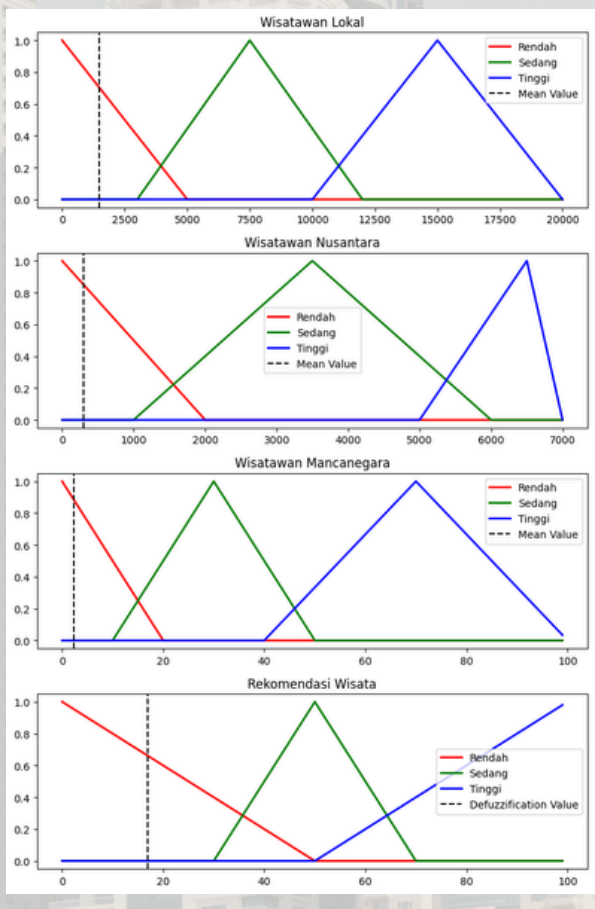
numpy as np, pandas as pd, scikit-fuzzy, matplotlib.pyplot as plt, skfuzzy as fuzz, sklearn.model_selection, train_test_split, skfuzzy, control as ctrlsklearn.metrics, mean_absolute_error, mean_squared_error, matplotlib.pyplot as plt











https://www.kaggle.com/code/giraldonainggolan/sistem-rekomendasi-tempat-wisata/

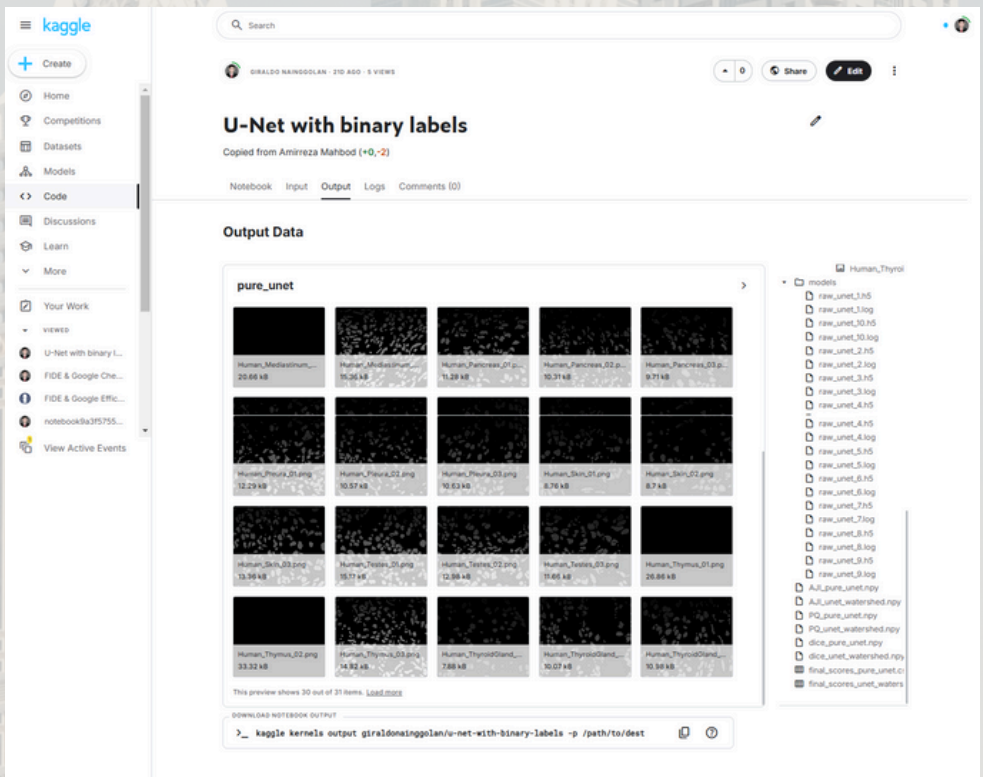
U-Net with binary labels

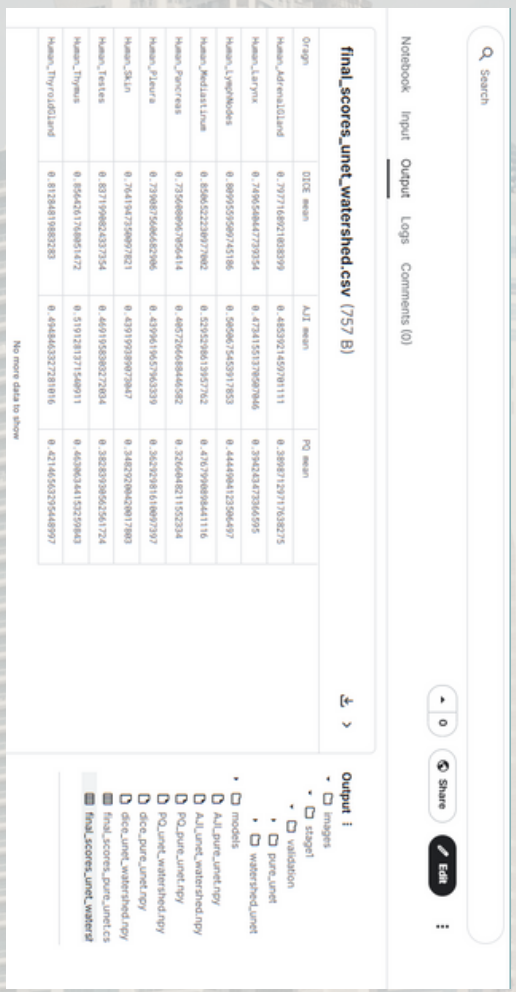
Dataset properties:

This notebook demonstrates the use of the U-Net architecture for image segmentation with binary labels. The goal is to classify each pixel in an image into two categories: the target object and the background.

Technologies Used:

Deep Learning Framework: TensorFlow/Keras, Model Architecture: U-Net, Programming Language: Python, Data Handling Libraries: NumPy, Pandas, Visualization Tools: Matplotlib, Experiment Platform: Kaggle Notebook





https://www.kaggle.com/code/giraldonainggolan/u-net-with-binary-labels

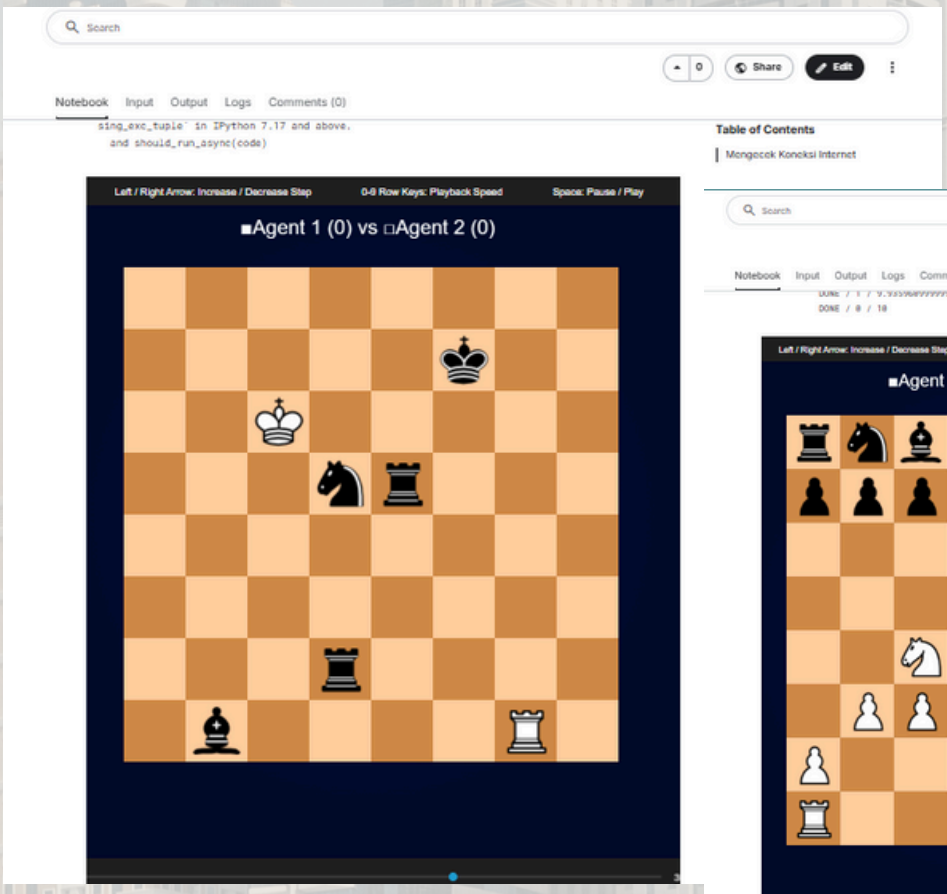
FIDE & Google Chess AI

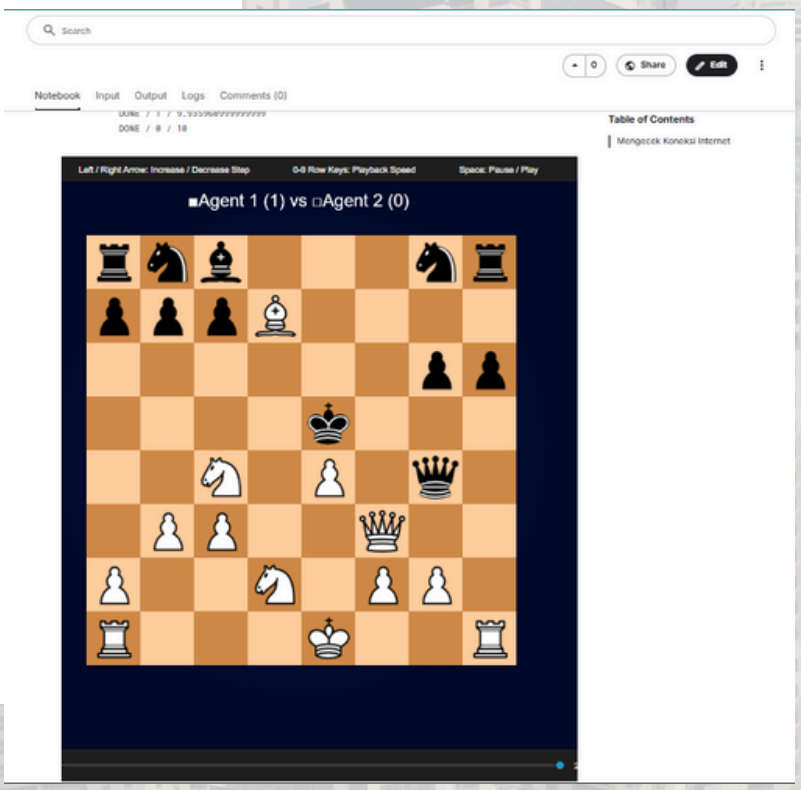
Dataset properties:

The "FIDE & Google Chess AI: Thinking Smarter, Not Hard" project explores AI technology in chess by developing a more strategic and ef icient system beyond brute-force calculations. By integrating optimized search algorithms like Monte Carlo Tree Search (MCTS) and Alpha-Beta Pruning with Google's AI technology, the project enhances move analysis and game strategy. Leveraging FIDE's chess database, the AI adapts to Grandmaster play patterns in real-time, making its decision-making process more intelligent and dynamic.

Technologies Used:

Python, TensorFlow & PyTorch, Stockfish & AlphaZero, Google Cloud AI, FIDE Chess Database, Monte Carlo Tree Search (MCTS), Alpha-Beta Pruning, Jupyter Notebook.





https://www.kaggle.com/code/giraldonainggolan/fide-google-chess-ai-thinking-smarter-not-hard

Machine Learning Regression

Dataset properties:

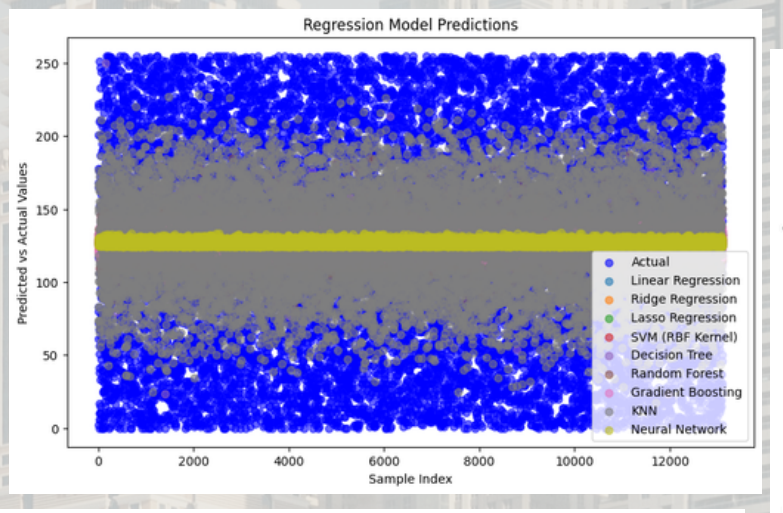
This dataset is designed to explore various regression methods in machine learning, including Linear Regression, Polynomial Regression, Ridge Regression, Lasso Regression, and Decision Tree Regression. It is suitable for training and experimenting with different regression models.

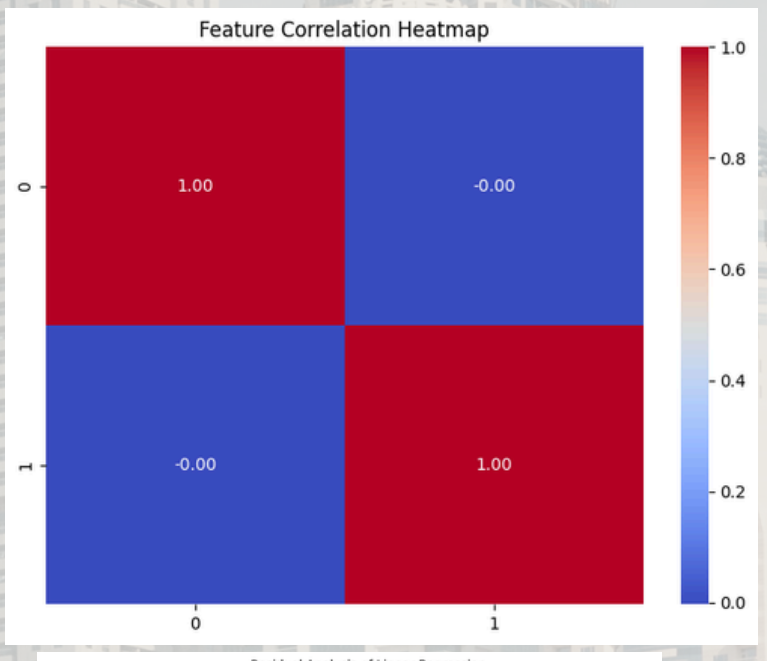
Variables in the dataset:

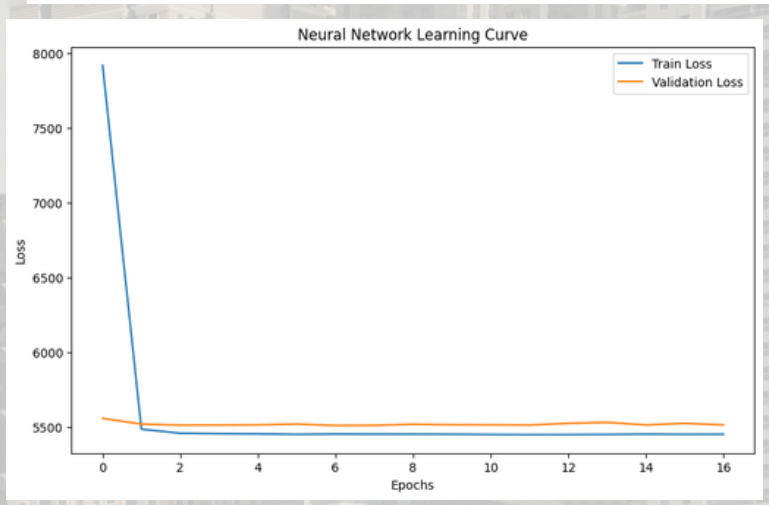
- *input1 → The first numerical variable used for prediction.*
- *input2 → The second numerical variable used for prediction.*
- *output → The target variable to be predicted.*

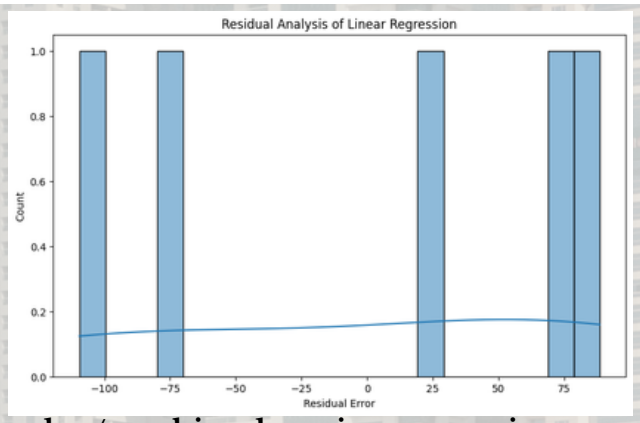
Technologies Used:

Python, Pandas , NumPy , Exel, Matplotlib & Seaborn, Scikit-Learn, Jupyter Notebook









https://www.kaggle.com/code/giraldonainggolan/machine-learning-regression

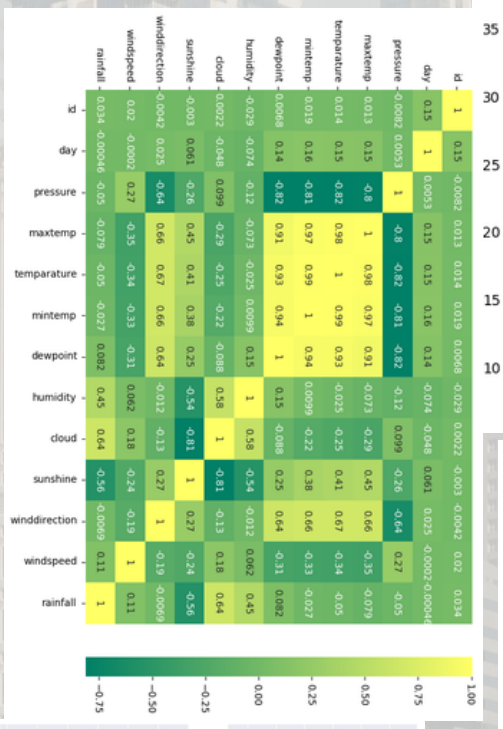
Data-Driven Rainfall Prediction

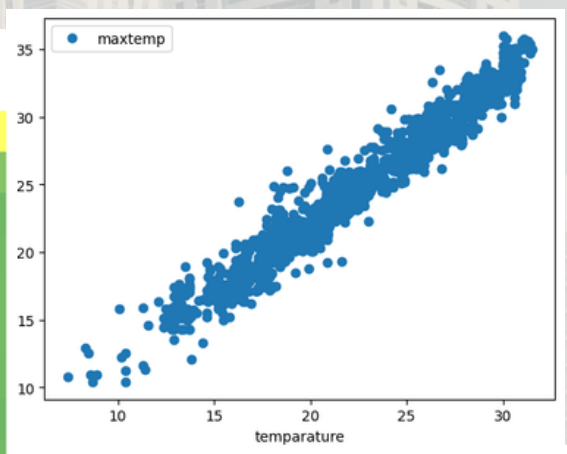
Dataset properties:

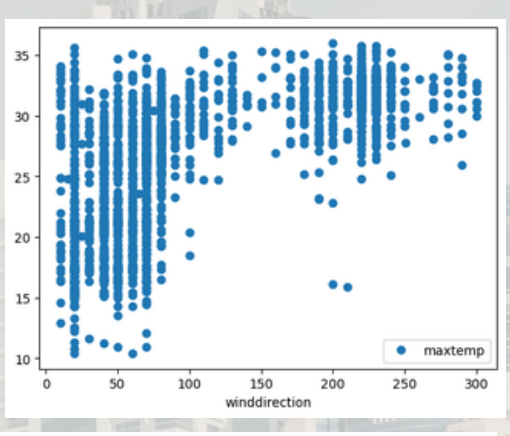
It includes Date & Time (YYYY-MM-DD HH:MM:SS) to track temporal patterns, Temperature (°C) representing the daily average temperature, and Humidity (%), which indicates moisture levels in the air. Additionally, Air Pressure (hPa) is included to analyze atmospheric conditions, along with Wind Speed (km/h) and Wind Direction (°) to assess air movement patterns. The dataset also records Rainfall (mm) as the primary target variable, measuring the amount of precipitation per day. Lastly, Weather Conditions provide categorical data on the general state of the weather (e.g., Clear, Cloudy, Rainy), further enhancing the prediction model's accuracy.

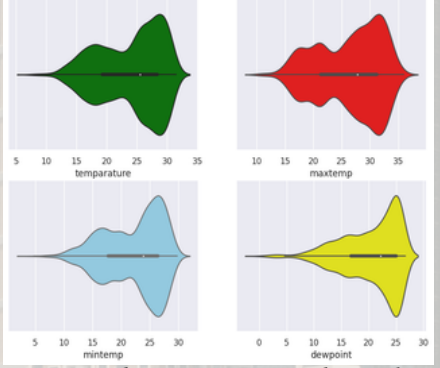
Technologies Used:

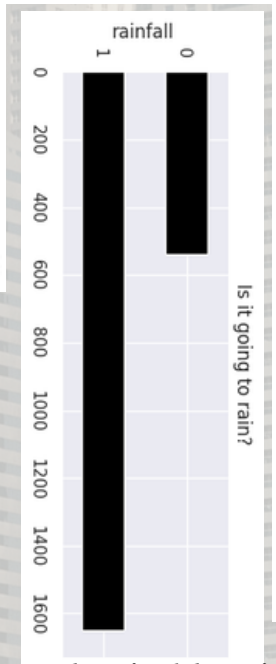
Data Preprocessing & Analysis, Machine Learning Model, Deployment & Tools, Output / Goals

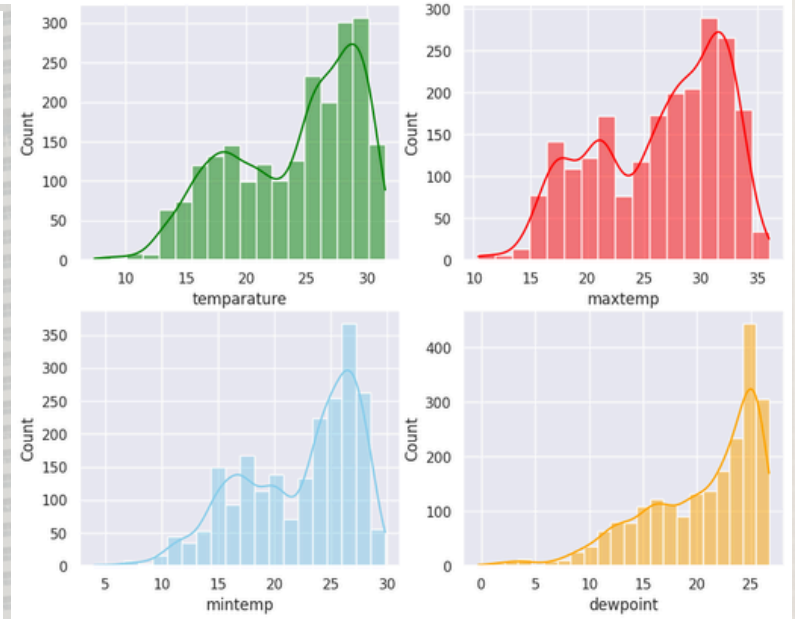












https://www.kaggle.com/code/giraldonainggolan/data-driven-rainfall-prediction