Università degli Studi di Milano-Bicocca

ARTIFICIAL INTELLIGENCE FOR SCIENCE AND TECHNOLOGY

Name: Subhrajyoti Sen Gupta
Email: s.sengupta1@campus.unimib.it
Matriculation: 898072

Supervised Deep Learning for RSSI Fingerprinting in Indoor Localization

AMBIENT INTELLIGENCE AND DOMOTICS

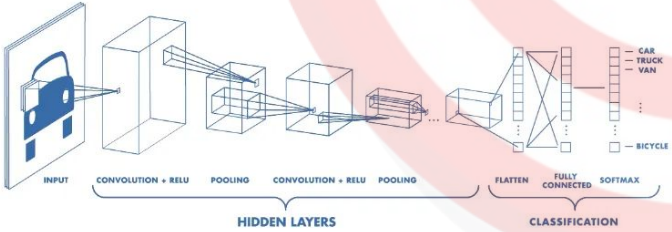
Project topic:

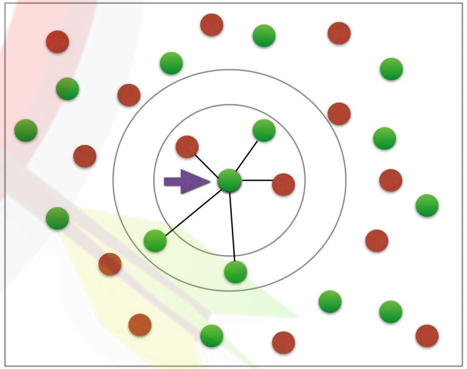
Supervised Deep Learning for RSSI Fingerprinting in Indoor Localization

Comparison Models

Convolution Neural Network(CNN) with simple and complex structuring.

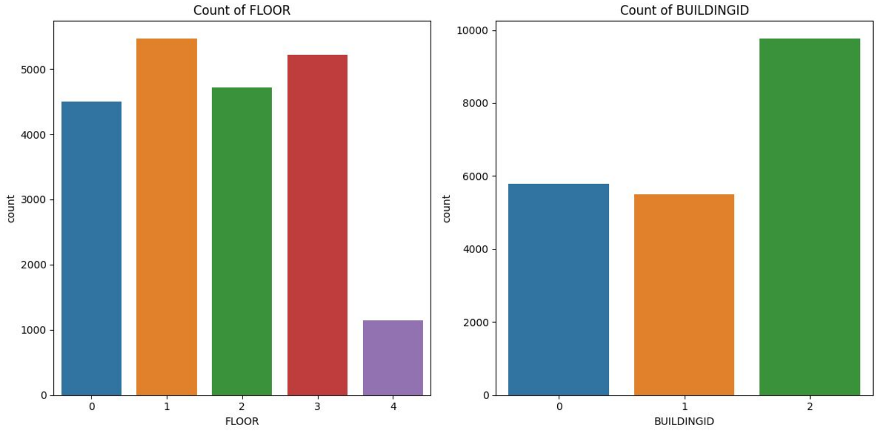
K-Nearest Neighbors Regressor (KNN)





Dataset Used:

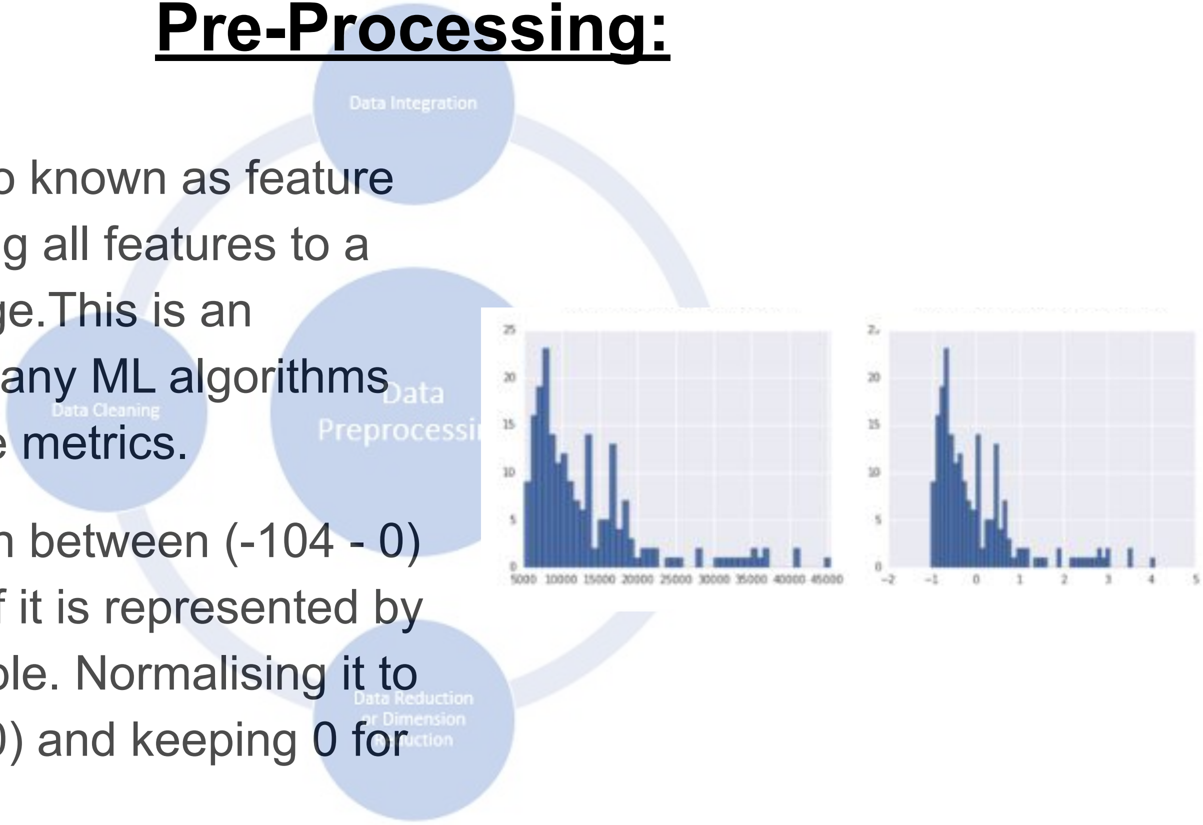
UjiIndoorLoc: An indoor localization dataset is a popular dataset for indoor localisation and positioning. The basic goal of the dataset is to determine precised location of the user or the device. It holds a good record for indoor navigation, asset tracking, and context-aware services.



Pre-Processing:

Normalisation: also known as feature scaling, aims to bring all features to a similar scale or range.This is an important step as many ML algorithms depends of distance metrics.

RSSI fingerprint is in between (-104 - 0) it is reachable and if it is represented by 100 it is not reachable. Normalising it to (0.25,1) for (-104 - 0) and keeping 0 for 100.



Models:

Simple Neural Network:

Here we have **Input layer** followed by **Dense layer** and **Output layer**.

Series of **fully connected (Dense) layers** with numbers of units **(neurons)** and **ReLU (Rectified Linear Unit) activation functions**.

The model is compiled using **“adam” optimizer**, loss function is **"mse" (Mean Squared Error)** and additionally, it tracks the **"accuracy"**.

Convolution Neural Network(CNN) model using Sequential API:

Here we have **input layer** implicitly defined with an input_shape of **(520, 1)** followed by **Convolutional Layers** contains three 1D convolutional layers and **max-pooling layers**.

Flatten Layer: added to **transform the 3D output into a 1D vector**. Dense layers are fully connected and intended for **feature processing and representation**.

Then we have output layer with same compiler.

Convolution Neural Network(CNN) model less complex :

The **input layer** implicitly defined with an input_shape of **(520, 1)** followed by **Convolutional Layers** contains two 1D convolutional layers and **max-pooling layers**.

Flatten Layer: added to **transform the 3D output into a 1D vector**. Dense layers are fully connected and intended for **feature processing and representation**.

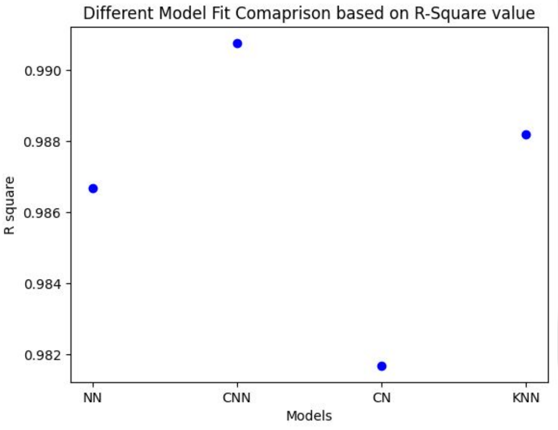
Then we have output layer with same compiler.

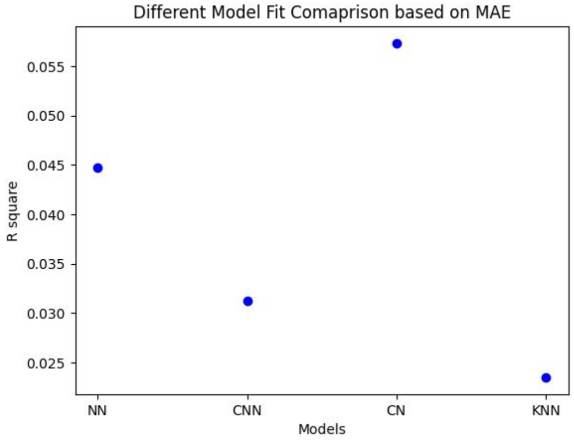
K-Nearest Neighbors Regressor (KNN) model :

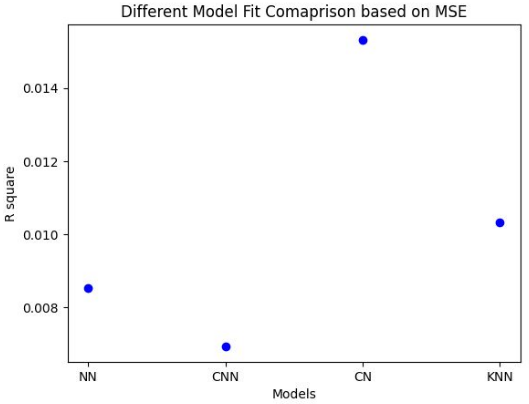
Here we define **k-Nearest Neighbors (k-NN) algorithm** for regression. We use **scikit-learn machine learning library** helps in creating an instance of the **KNeighborsRegressor class**.

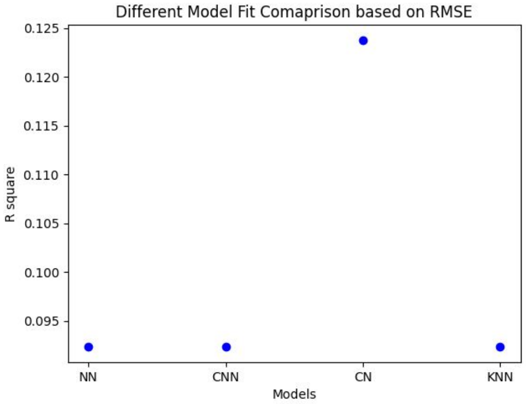
Hyperparameter **“n_neighbors”** is set to 20, which means that the **k-NN algorithm will consider the 20 nearest neighbors when making predictions**.

Results:


Different Model Fit Comaprison based on R-Square value


Different Model Fit Comaprison based on MAE


Different Model Fit Comaprison based on MSE


Different Model Fit Comaprison based on RMSE

Code Link:

Colab Link:

https://colab.research.google.com/drive/10HF7c-JHzNaPUn8jcJX05o0HIcNC3Tfi?usp=sharing