Abstract.

The Project investigates the relationship between weather conditions and household energy consumption in Melbourne, Australia. Using multiple datasets which contains weather data of Melbourne, and the power usage data of two house in Melbourne area. And one data set contains the solar power usage of one household. The project aims to predict the power consumption of the households based on the climatic factors. The whole project consists of through data cleaning, data preprocessing, feature engineering and the development of the machine learning model that will be used to predict the power consumption of the models. For energy optimization and development of sustainable energy solutions, an understanding of how weather influences household power consumption is important. To project power usage patterns, this study merged weather data with two (02) Melbourne households’ energy use records. The method involves the integration of datasets from different sources, their preprocessing, feature selection and extraction as well as application of neural network model for electricity forecasting.

In the first step of the study, the datasets containing weather data (bom\_year.csv), electricity consumption for two houses (House 3\_Melb East.csv and House 4\_Melb West.csv), and solar power generated (House 4\_Solar.csv) were loaded from Google Drive. The column names were changed to normal formats, and the date and time stamps were changed into datetime types. These datasets were then joined based on the date and time stamp fields to make a single set of data. Data cleaning was done by deleting unwanted fields and managing the missing values. Exploratory data analysis included feature creation in which new fields like average temperature, temperature difference was introduced and feature selection where relevant fields like temperature and rainfall were incorporated into the prediction model. Before the data was used to train the model, the set was divided into the training and testing set and the feature data then normalized. Neural network was designed with TensorFlow’s Keras, the architecture included dense network layers with the ReLU activation function for the intermediate layers and the final layer for regression. For the prediction of the power consumption of House 3, House 4 a model was trained. Mean Absolute Error was applied for model assessment, and training-validation loss over epoch was plotted. The model utilized in the study was satisfactorily effective as identified by the MAE results of the two houses. The plots of performance and losses gave understanding of the training and validation process regarding the models for both houses.

From the project, it is evident that artificial neural network correctly estimates the power consumption by the household subject to the climate data. The proposed architecture and the use of different datasets in the same process, and the use of preprocessing and feature engineering, as well as applying a machine learning algorithm, created a solid background for energy consumption prediction and analysis.

Intro

The swift progress in machine learning has created novel approaches to address complex issues in several fields, such as energy management. Predicting power consumption, which is essential for optimizing energy use, improving grid stability, and cutting operating costs, is one of machine learning's most important uses. Precise forecasting of power use facilitates educated decision-making for utilities and consumers, encouraging economical energy utilization and bolstering the endurance of energy reserves.Machine learning algorithms are especially well-suited for power consumption prediction because of their capacity to analyze massive amounts of data and identify subtle patterns. In order to predict future energy demands, these algorithms can evaluate previous power usage data in addition to many influencing elements like the time of day, weather, and occupancy levels. The effectiveness of methods like Artificial Neural Networks (ANN), Support Vector Machines (SVM), Random Forests, and Long Short-Term Memory (LSTM) networks in predicting power usage has been thoroughly investigated.

Mantri et al. (2021) explore a hybrid approach combining neural networks and k-Nearest Neighbors (k-NN) for weather prediction and classification. Their model successfully predicts temperature and humidity with high accuracy, as indicated by R-squared values close to one and Mean Squared Error (MSE) values approaching zero. The classification model, which uses the outputs of the prediction model, also performs well, achieving high accuracy in classifying weather conditions. This hybrid model shows the potential of combining different machine learning techniques to enhance the accuracy and reliability of weather forecasts.

Another notable approach is the use of Long Short-Term Memory (LSTM) networks, as discussed in a study focusing on weather prediction using LSTM neural networks. LSTMs are particularly well-suited for time series prediction due to their ability to capture long-term dependencies in data. The study demonstrates that LSTM networks outperform traditional methods in predicting weather variables like temperature and precipitation, providing more accurate and stable forecasts. This highlights the effectiveness of deep learning models in handling complex temporal patterns in weather data, further solidifying their importance in the field of meteorology.

Model.

In the model several libraries are used for data manipulation, machine learning and visualization. Pandas library is called for data manipulation and analysis, this library provides data frame structures to handle structured data efficiently. NumPy library is a fundamental library for scientific computing and to handle multi-dimensional arrays. Scikit learn library provides powerful tools for predictive analysis and includes train\_test\_split for dividing the dataset for trsining and testing. Also, standardscaler is used in this to normalize the data. Also Tensorflow and its high level API keras is used for building and training neural network models. The sequential model from the keras library is used to create a stack of layers, which includes the dense layers for the filly connected networks, LSTM model is used here because it is more suitable for timeseries data, The dropout layers prevent the layers from overfitting. And batchnormalization layers are used to normalize the output. And the L2 regularization function prevents overfitting as well in the model. Adam optimizer function is used for efficient training of the model. And the matplotlib is used to visualize the model performance for a better understanding.

This code mounts the user's Google Drive, loads particular CSV files into Pandas DataFrames, and verifies the data. It is intended to run in a Google Colab environment. The code first mounts Google Drive, using drive.flush\_and\_unmount() and drive.mount() to make sure that changes are preserved and visible. It imports the required libraries, such as NumPy for scientific computation and Pandas for data handling. The datasets' file locations are specified, and pd.read\_csv() is used to load the CSV files into DataFrames. The first few rows of every DataFrame are printed in order to confirm the data. Next, using the names option and header=None, the code defines column names for datasets without headers and reloads the datasets with these new column names. This guarantees that the datasets are appropriately organized and prepared for analysis.

The below code snippet focuses on preprocessing and cleaning the datasets to prepare them for analysis and feed them into the createdt model. Firastly, it converts the Date and Timestamp columns in the weather data, House 3 data, and House 4 datasets into datetime objects to ensure consistency in date and time formats. The power consumption and solar generation data for House 4 are then merged, and total power consumption is calculated by summing grid and solar power values. Subsequently, the weather data, House 3 data, and combined House 4 data are merged into a single DataFrame based on the date and timestamp columns. Unnecessary columns are dropped to streamline the dataset, and any rows with null values are removed to ensure data integrity. There are no null values and it is confirmed, resulting in a clean and well-structured dataset ready for further analysis and model training.

In this section of the code the feature engineering part is described for the reader to have a better understanding.The reasons as to wehy these five features are seleceted is explained here. First feature is the MAX and MIN Temperature, These are the fundamental indicators of the daily weather conditions and this can impact the energy consumption patterns of the houses. High temperature means that the houses turn oin the cooling systems, and high temperature meqan that the heat usage will be increased. Next feature is Total rainfall, rainfall can affect power consumption of a household which has outdoor activities or solar power generators, heavy rain can reduce the efficiency of the solar power generators. Next feature selected is the temperature at 9 AM and 3 PM. In the data set given these features gives a idea about how daily temperature variations can influence the power usage of a house. There for this is going to be a key feature as well. Another feature will be the average temperature. The avgerage temperature provides a single metrics that summarize the overall warmth or the coolness, this will simplify the models interpretation of the temperature. Temperature difference, This feature helps to capture the variability in the temperature through out the day, which will influence the energy consumption of houses. All the selceted featrures are explained and now I will explain the code snippet for a better understanding.

In the featuring engineeeriring code first i convereted the Max\_temperature and Min\_temperature columns in to the numeric type to make sure that all the temperature values are trurned into the numerical operations. The second code is for creating 2 new features. avg\_temperature is calculated by adding max and min temperature and and the temp\_difference by getting the difference between max and min temperatur. These new features can provide information relating to the daily temperature and average temperature. after that i selected the features that will be used to train the model and also the target variableas for the House 3 and house 4 respectively. then seperate the features from the sleceted target variables y\_house3 and y\_house4. Now print the three dataframes to verify if they are correctly spearted before trainig the model. and then i split the datasets respectively for testing and training for house\_3 and house\_4. the parameter for test is 0.2 whihc means it is 20% and the trainig is 80%. The random\_state 42 makes sures the data is split the exact same way each time the model is trained and tested. Then i have agai= printed the datasets to verify the split of the dataset. After that i have used a function called StandardScaler() to standardize the features to have a mean of 0 and a sd of 1. this is an important feature to make sure the features are on the same scale which will improve it's features.