# Chapter 1 Implementation

## 1.1 Chapter Overview

The Implementation chapter will provide a detailed explanation on the prototype's implementation, including a thorough rundown of the programming languages, technologies, frameworks, and libraries used to build the solarex mobile app. n. A comprehensive discussion of the features of the prototype of Solarex is provided along with how each feature was developed, intended use of each feature, relevant code snippets and screen shots, the challenges encountered and solutions taken while implementing each feature.

## 1.2 Overview of the prototype

The backend will predict the solar irradiance of the given date and time according to its location, and its surrounding obstacles. Once the prediction is made it will provide with a report of the solar productivity. The Flutter framework has been used to build the front end, which will process the data entered by the user and display a report of solar productivity for the specified date and time.

## 1.3 Technology selections

### 1.3.1 Language selection

Python

Python was selected as the main programming language for the implementation of this project. Python is increasingly used in both industry and academic settings as it is extremely powerful and flexible. Availability of tutorials and support with Machine learning related tasks with Python.

### 1.3.2 Libraries/Frameworks selection

Flutter

Flutter is an open source mobile UI framework. It allows you to create native mobile applications using only one code base. This means we can use one programming language and one code base to create two different applications (for iOS and Android).

Numpy

NumPy is an open source digital Python library. NumPy contains multi-dimensional arrays and matrix data structures. It can be used to perform many mathematical operations on arrays, such as trigonometric functions, statistical functions, and algebraic routines.

Pandas

Pandas was used for data analysis tasks in the project. Pandas is an extremely fast and efficient python package and used by professionals for Python projects. Pandas was used to read the CSV datasets and also to view the data in a data frame. Since the dataset had a lot of reviews, it was efficient to use pandas due to how well it performs when compared to other CSV readers.

Seaborn

Seaborn is a Python data visualization library based on matplotlib. It provides an advanced interface for drawing fascinating and informative statistical graphics.

Pylab

PyLab is the process interface of Matplotlib's object-oriented plotting library. Matplotlib is the entire package; matplotlib.pyplot is a module in Matplotlib; PyLab is a module installed with Matplotlib.PyLab is a convenient module that can batch import matplotlib.pyplot (for plotting) and NumPy (for math and using arrays) in a single namespace.

Matplotlib

Matplotlib is a plotting library for the Python programming language and its digital math extension NumPy. It provides an object-oriented API for embedding graphs into applications using a common GUI.

Scikit-learn (Sklearn)

Scikit – learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. Scikit-Learn is a popular Python ML library which offers numerous classification

algorithms, regression algorithms and clustering algorithms.

1.3.3 Summary of components in Solarex

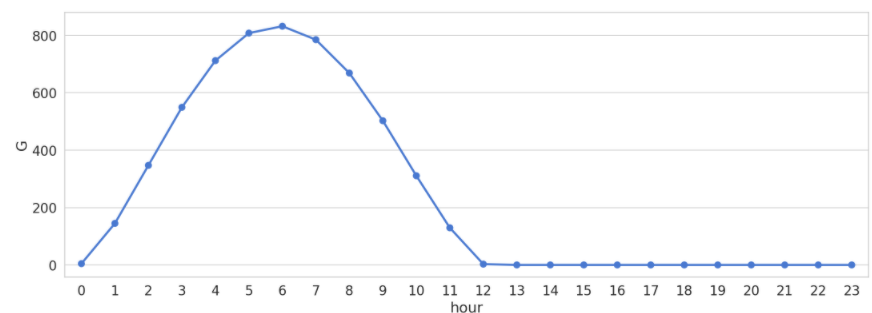
|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1.4 Implementation of the Data science

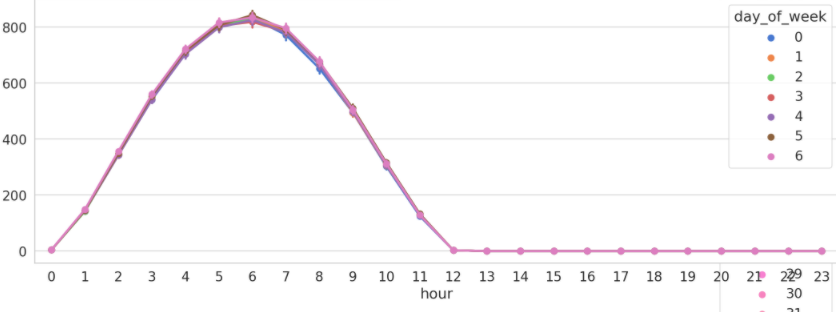
Data science is a broad area that employs scientific methods, processes, algorithms, and systems to extract information and insights from structured and unstructured data, as well as to extend that knowledge and actionable predictive analytics through a wide range of application domains. Data mining, machine learning, and big data are all linked to data science. Data science unifies statistics, data processing, informatics, and their associated processes in order to use data to understand and interpret real phenomena. Three in the context of mathematics, statistics, computer technology, information science, and domain knowledge, it employs techniques and hypotheses from a variety of fields.

In a data science project, the data set is one of the most critical elements. Initially, without a dataset, any data science project could not continue. Therefore, it is important to provide a project dataset. A dataset is collection of related sets of information that is composed of separate elements but can be manipulated as a unit by a computer. As solar irradiance is forecast by the system, the dataset includes attributes relevant to the prediction to be made. Which are solar irradiance from 2005 – 2016, that has been sorted into a particular time, date and for a year. The data set was gathered through digging which was done by the authors on the internet. At first the dataset was unorganized, and a dataset cleaning was need to be done. Getting rid of useless data and removing duplicate data was done. It was done by using Pandas. There were some minor changes to be made to the format of the data and the library Numpy was used. The library Seaborn was used to display the graphs and also it used Matplotlib underneath to plot graphs. It will be used to visualize random distributions. Scikit-learn (Sklearn) was used for the machine learning part and regression model was based on Sklearn. The behavior of the data set was divided into hour of the day, day of the week, day of the month and monthly.

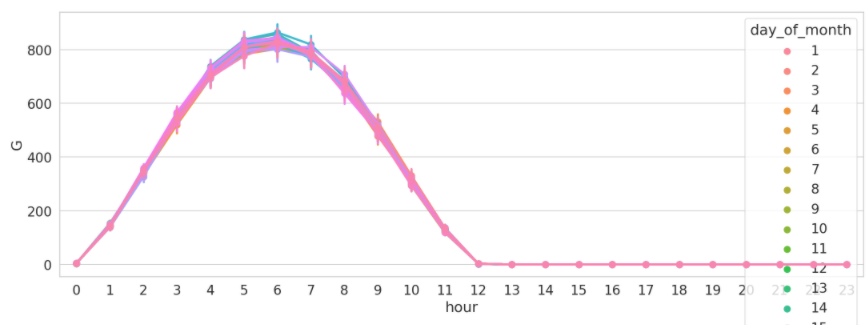
Hour of the Day



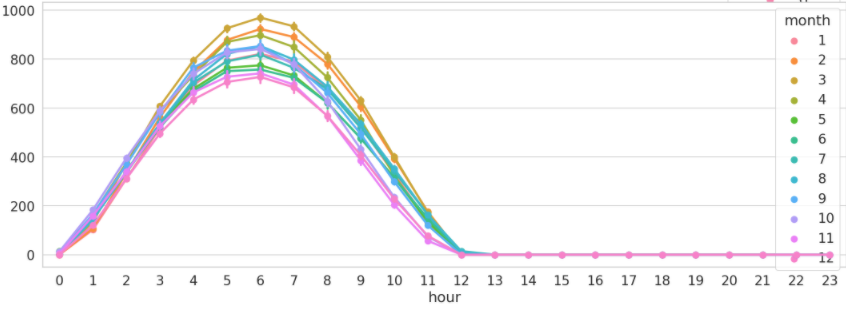
Day of the Week



Day of the Month

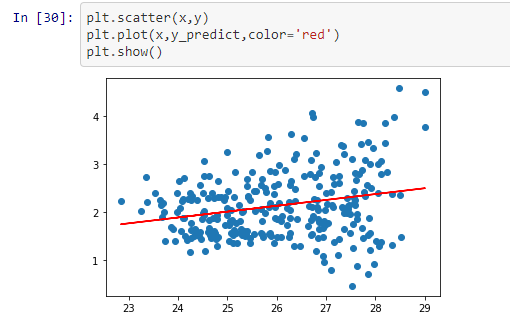


Month of the Year

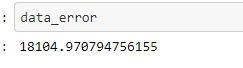


According to the dataset the calculation and the prediction is the Solar Irradiance (G score) for a given date and time using past data. So to predict that several models were tested and trained until the correct one was trained.

Linear regression model was at first. Linear regression model is a simple model which has a linear approach to modelling the relationship between a scalar response and one or more explanatory variables. The behavior of the variables and the constants are much alike seasonal forecasting. So the results were mostly spread. Linear regression model was dropped due to that reason.

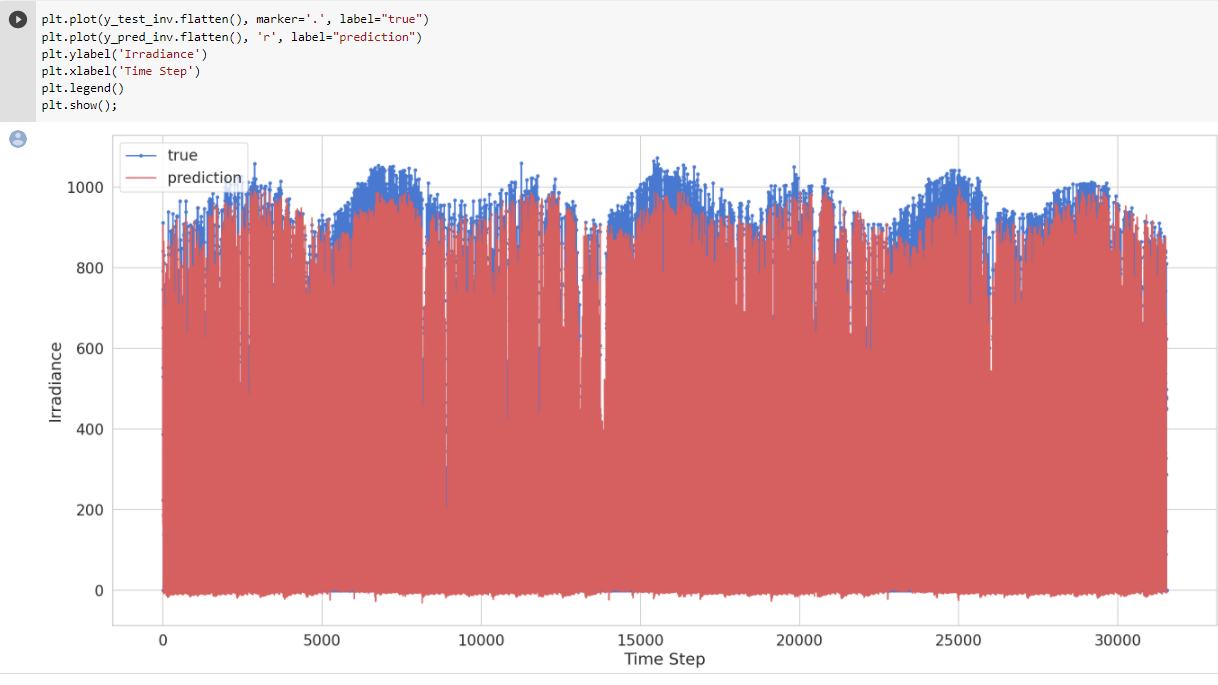


ARIMA model was tested next. Auto Regressive Integrated Moving Average model a model that uses time series analysis. It takes past values from a dataset and calculates them with a help of a formula and predicts future values. But with our dataset the results were unacceptable. The accuracy was low and the data error was bit high.

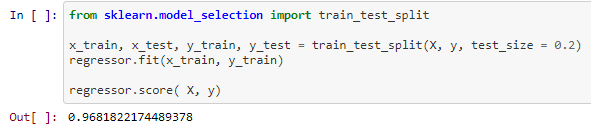


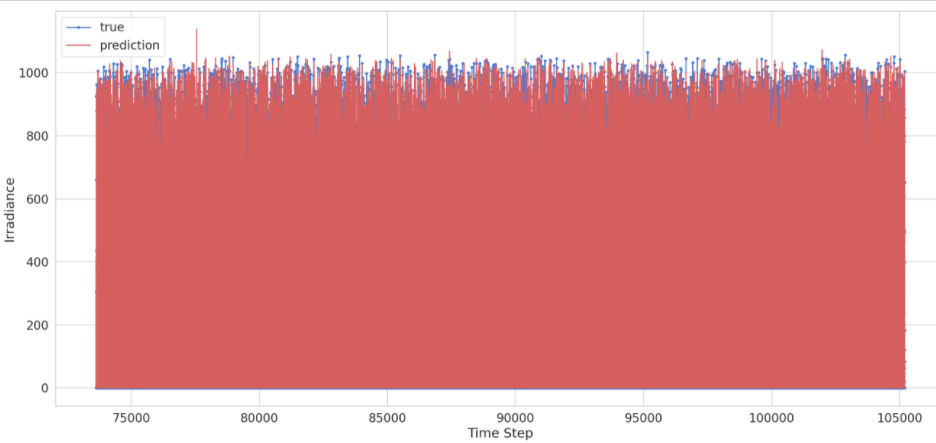


LSTM model was trained and tested next. LSTM is a special kind of recurrent neural network that is capable of learning long term dependencies in data. After analyzing the predictions, the expected results were close, but not 100%. According to the proposed project user has the capability of inputting any given date and a time (any day, month, year). But this model couldn’t predict that. To get a prediction from this model it needs the most recent data. As an example if you want to predict tomorrow’s irradiance the model needs today’s data. But data set only consists data within 2005 to 2016. So model couldn’t able to provide any predictions from 2016. And there was a considerable gap between the predictions and test data.



Finally, Decision Tree regression model was tested. This model is also considered as a classification model, and formed as a tree. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. After implementing and training this model with the dataset the results were better. The accuracy between the test data and the predicted data were above 90%.





1.5 Implementation of the features of the prototype in the backend

1.5.1 Stage 1 – Pre-Processing

1.5.2 Stage 2 - Term Frequency using TF-IDF

1.5.3 Stage 3 - Using the Classification Algorithms

1.5.4 Stage 4 - Testing the model using a review

1.5.5 Stage 5 – Categorization of the issue

1.6 Implementation of the GUI

1.1.1 Implemented Home screen

1.1.2 Implemented submit reviews page

1.1.3 Implemented results page

1.7 Deployments/CI-CD Pipeline

1.8 Chapter Summary

This chapter discussed about the prototype feature implementation process of Solarex. It gave a detailed description about the selected technologies, along with why Python was chosen as the core development language. Afterwards the chapter discussed about the other libraries and the frameworks that were used in building the prototype for Solarex. This chapter also described how each feature of the solution was implemented along with explanations through code snippets and screenshots for of the implementation along with the problems encountered while implementing and the solutions taken. Details about the user manual was also detailed at the end of this chapter. The next chapter will discuss the testing phase of Solarex.

Chapter 2 – TestingSolarex

2.1 Chapter Overview

This chapter covers the various testing done on the system to maintain. The tests covered were both functional and non-functional with black box methods. The objectives of the testing along with the criteria followed are presented with the test plan. Unit testing and integration test done on the code base are also documented. Finally, the benchmarking results are covered.

2.3 Testing criteria

2.4 Testing Functional requirements

2.5 Testing Non-functional requirements

2.6 Unit testing

2.7 Performance testing

2.8 Usability testing

2.9 Compatibility testing

2.10 Chapter Summary

Chapter 8 – Evaluation

3.1 Chapter Overview

3.3 Evaluation method

3.4 Evaluation types

3.5 Quantitative evaluation

3.6 Qualitative evaluation

3.1.1 Feedback gathered from end users

3.1.2 Feedback gathered from domain experts

3.1.3 Feedback gathered from Industry experts

3.7 Self-Evaluation

3.8 Chapter Summary