Low Level Document (LLD)

Concrete Compressive Strength Prediction

Rajesh Gottoju

Contents

Abstract

Abstract

The quality of concrete is determined by its compressive strength, which is measured using a conventional crushing test on a concrete cylinder. The strength of the concrete is also a vital aspect in achieving the requisite longevity. It will take 28 days to test strength, which is a long period. So, what will we do now? We can save a lot of time and effort by using Data Science to estimate how much quantity of which raw material we need for acceptable compressive strength.

1.Introduction

Why this Low-Level Design Documentation?

The purpose of this documentation is detailed description of restaurant rating prediction system which will explain the purpose and the feature of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will perform under different parameters. This document is intended for both the stack holders and developers of the system and will be proposed for the higher management for its approval.

This project can be delivered in three phases

Phase 1: Building Machine learning model depending on the requirements.

Phase 2: Integration of UI and database to all the functionalities.

Phase 3: Deployment of project on cloud.

Scope

This software system will be a web application, this system will be designed to predict the Strength Of cement based on user's input.

Constraints

This project is based on Infra domain, this system can get excepted results.

Out Of Scope

System will not perform correctly if the data in good format

2. Technical Specifications

Data: Concrete Compressive Strength Prediction

Finalized: Yes

Data Set overview

1030 rows

9 columns

	cement	blast_furnace_slag	fly_ash	water	superplasticizer	coarse_aggregate	fine_aggregate	age	concrete_compressive_strength
0	540.0	0.0	0.0	162.0	2.5	1040.0	676.0	28	79.99
1	540.0	0.0	0.0	162.0	2.5	1055.0	676.0	28	61.89
2	332.5	142.5	0.0	228.0	0.0	932.0	594.0	270	40.27
3	332.5	142.5	0.0	228.0	0.0	932.0	594.0	365	41.05
4	198.6	132.4	0.0	192.0	0.0	978.4	825.5	360	44.30
...	...	...	...	...	...	...	...	...	...
1025	276.4	116.0	90.3	179.6	8.9	870.1	768.3	28	44.28
1026	322.2	0.0	115.6	196.0	10.4	817.9	813.4	28	31.18
1027	148.5	139.4	108.6	192.7	6.1	892.4	780.0	28	23.70
1028	159.1	186.7	0.0	175.6	11.3	989.6	788.9	28	32.77
1029	260.9	100.5	78.3	200.6	8.6	864.5	761.5	28	32.40

1030 rows × 9 columns

Input schema

```
df.info()
```

```
RangeIndex: 1030 entries, 0 to 1029
Data columns (total 9 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   cement                        1030 non-null   float64
 1   blast_furnace_slag            1030 non-null   float64
 2   fly_ash                       1030 non-null   float64
 3   water                         1030 non-null   float64
 4   superplasticizer              1030 non-null   float64
 5   coarse_aggregate              1030 non-null   float64
 6   fine_aggregate                1030 non-null   float64
 7   age                           1030 non-null   int64
 8   concrete_compressive_strength 1030 non-null   float64
dtypes: float64(8), int64(1)
memory usage: 72.5 KB
```

Predicting

- The system displays Strength Of Cement according to user's Input.
- The system prevents the set of inputs required from the user.

- The user gives required information.
- The system should able to predict the Compressive Strength Of Concrete According to the user input given.

Logging

✓ We have chosen File logging.

✓ System logs each and every system flow.

✓ Each and every user's input information is logged.

Database

The system stores each and every data given by the user or received on request to the database. We have used Cassandra.

Deployment

1.Heroku

3.Technology Stack

* Python

* Stream lit

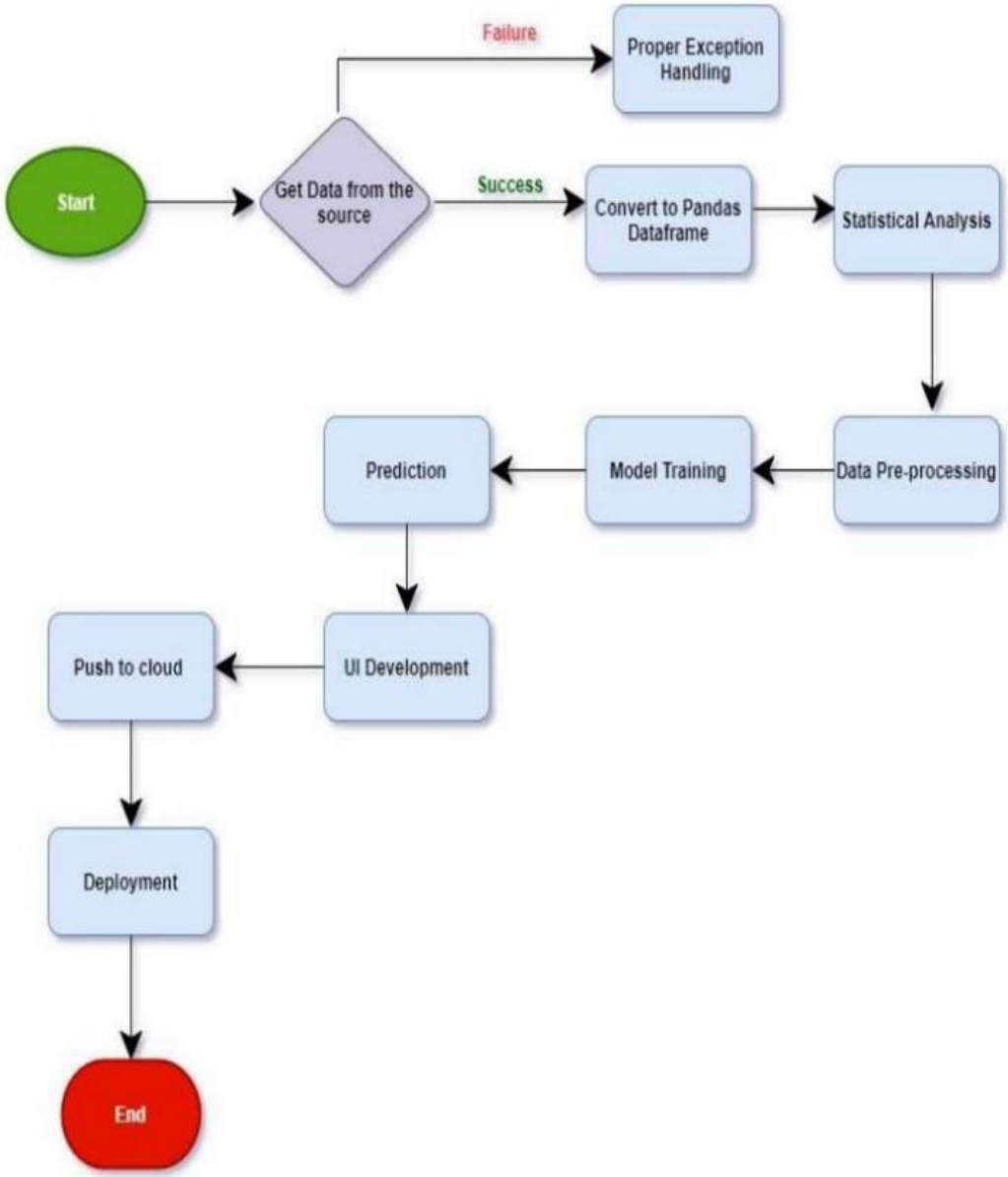
* Python Libraries

* Machine Learning algorithms

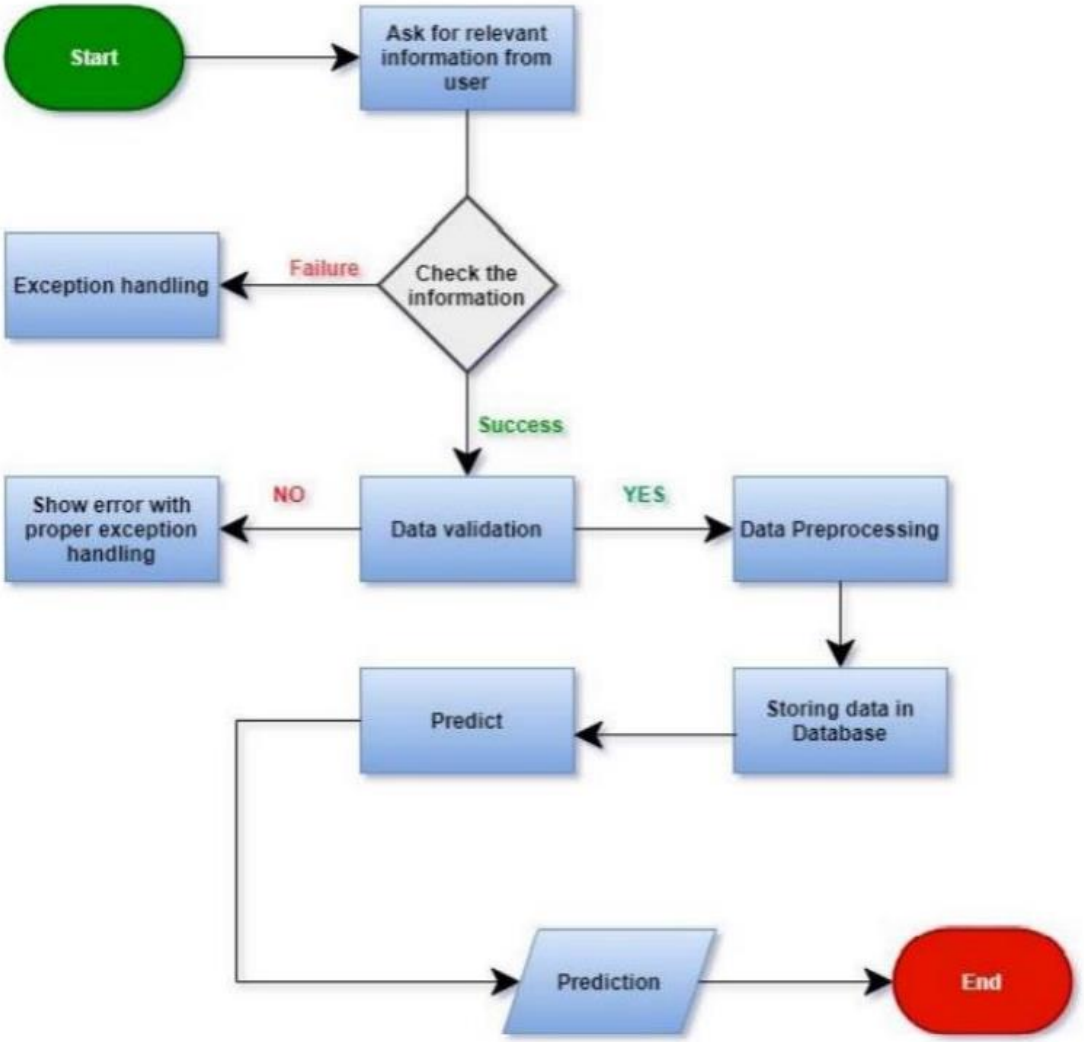
4. Proposed Solution

This system requires some sensor column values.

5.Model Training / Validation Workflow


Failure
Proper Exception Handling
Start
Get Data from the source
Success
Convert to Pandas Dataframe
Statistical Analysis
Data Pre-processing
Model Training
Prediction
UI Development
Push to cloud
Deployment
End

6. User I/O Workflow



7.Test Cases.

Test Case Description	Pre-Requisite	Expected Result
Verify whether the Application URL is accessible to the user	Application URL should be defined	Application URL should be accessible to the user
Verify whether the Application loads completely for the user when the URL is accessed	1. Application URL is accessible 2. Application is deployed	The Application should load completely for the user when the URL is accessed
Verify whether user is able to edit all input fields	1. Application is accessible 2. User is logged in to the application	User should be able to edit all input fields
Verify whether user gets Submit button to submit the inputs	1. Application is accessible 2. User is logged in to the application	User should get Submit button to submit the inputs
Verify whether user is presented with recommended results on clicking submit	1. Application is accessible 2. User is logged in to the application	User should be presented with recommended results on clicking submit
Verify whether the recommended results are in accordance to the selections user made	1. Application is accessible 2. User is logged in to the application	The recommended results should be in accordance to the selections user made