TABLE OF CONTENT

Chapter 2 : Introduction

Chapter 3 : Approach

Chapter 4 : Data Description

Chapter 5 : Modeling

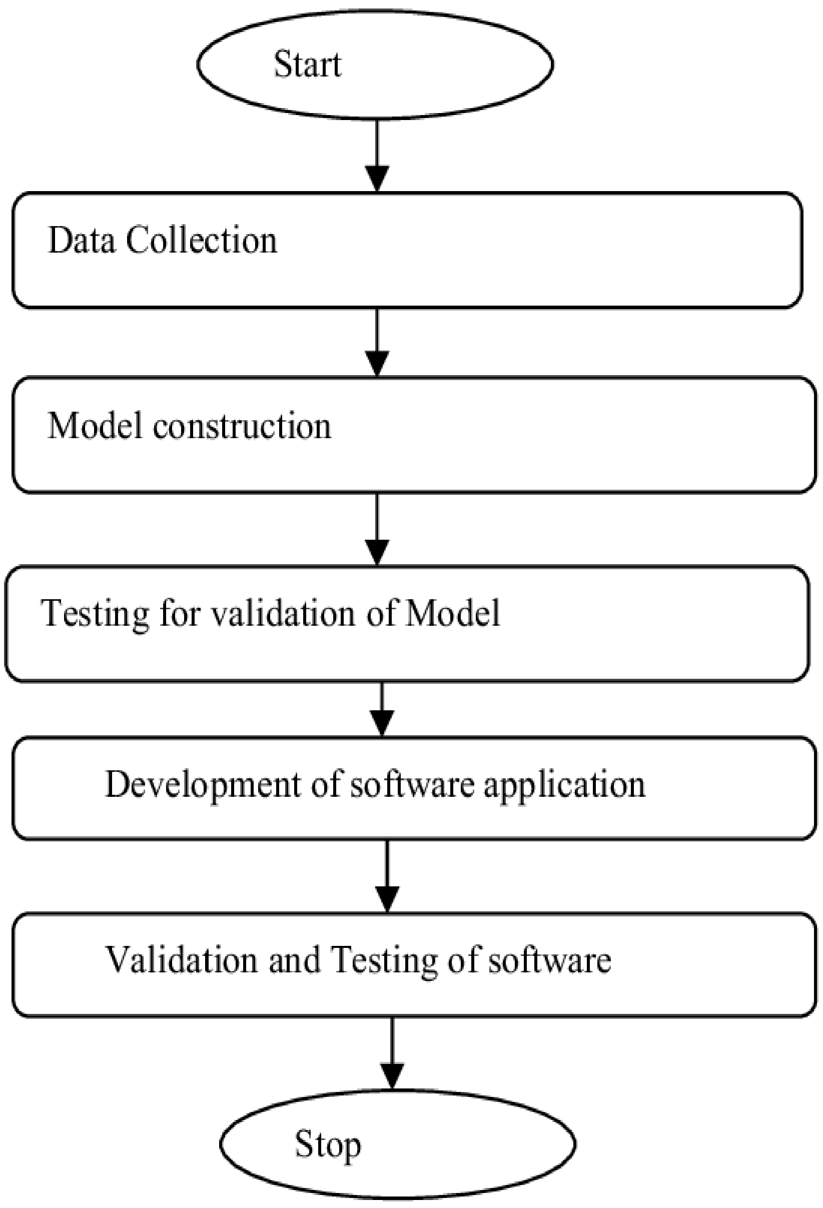
Chapter 6 : Result

Chapter 7 : Suggestion and Future Scope

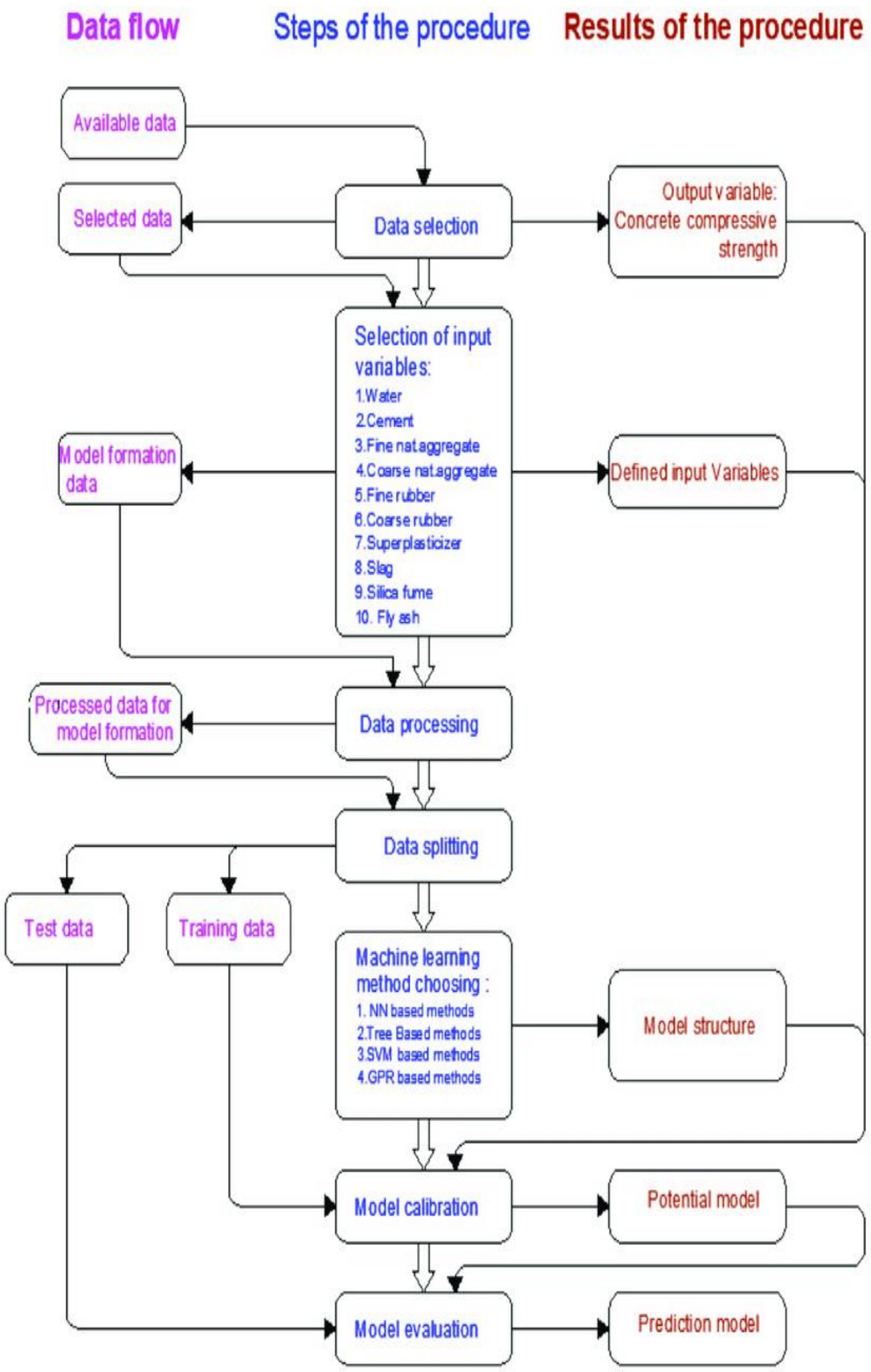
INTRODUCTION :

Concrete is such a construction material that is widely used in the world. The advantages of concrete are low cost, availability of construction, workability, durability and convenient compressive strength that make it popular near engineers and builders. However, these advantages seriously depend on the correct mix, placing and curing . In construction industry, strength is a primary criterion in selecting a concrete for a particular application. Concrete used for construction gains strength over a long period of time after pouring the characteristic strength of concrete is defined as the compressive strength of a sample that has been aged for 28 days. Neither waiting 28 days from such a test would serve the rapidity of construction, nor neglecting it, would serve the quality control process on concrete in large construction sites. Therefore, rapid and reliable prediction for the strength of concrete would be of great significance . For example, it provide a chance to do the necessary adjustment on the mix proportion used to avoid situation where concrete does not reach the required design strength or by avoiding concrete that is unnecessarily strong, and also, for more economic use of raw materials and fewer construction failures, hence reducing construction cost . Prediction of concrete strength, therefore, has been an active area of research and a Considerable number of studies have been carried out. Many attempts have been made to obtain a suitable mathematical model which is capable of predicting strength of concrete at various ages with acceptable (high) accuracy .

Approach:



Start
Data Collection
Model construction
Testing for validation of Model
Development of software application
Validation and Testing of software
Stop


Data flow
Steps of the procedure
Results of the procedure
Available data
Selected data
Data selection
Output variable: Concrete compressive strength
Selection of input variables:
1.Water
2.Cement
3.Fine nat.aggregate
4.Coarse nat.aggregate
5.Fine rubber
6.Coarse rubber
7.Superplasticizer
8.Slag
9.Silica fume
10. Fly ash
Model formation data
Defined input Variables
Processed data for model formation
Data processing
Data splitting
Test data
Training data
Machine learning method choosing :
1. NN based methods
2.Tree Based methods
3.SVM based methods
4.GPR based methods
Model structure
Model calibration
Potential model
Model evaluation
Prediction model

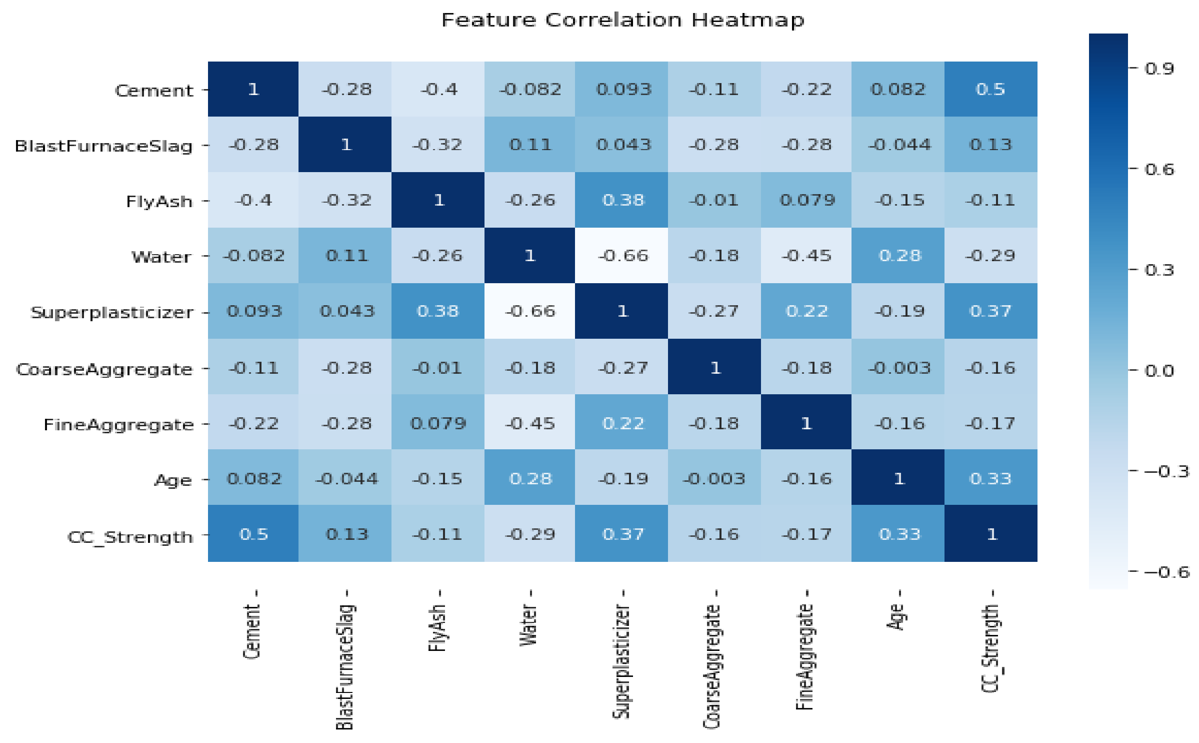
Data Description:

The dataset consists of 1030 instances with 9 attributes and has no missing values. There are 8 input variables and 1 output variable. Seven input variables represent the amount of raw material (measured in kg/m³) and one represents Age (in Days). The target variable is Concrete Compressive Strength measured in (MPa — Mega Pascal). We shall explore the data to see how input features are affecting compressive strength.

Exploratory Data Analysis

The first step in a Data Science project is to understand the data and gain insights from the data before doing any modelling. This includes checking for any missing values, plotting the features with respect to the target variable, observing the distributions of all the features and so on. Let us import the data and start analysing.

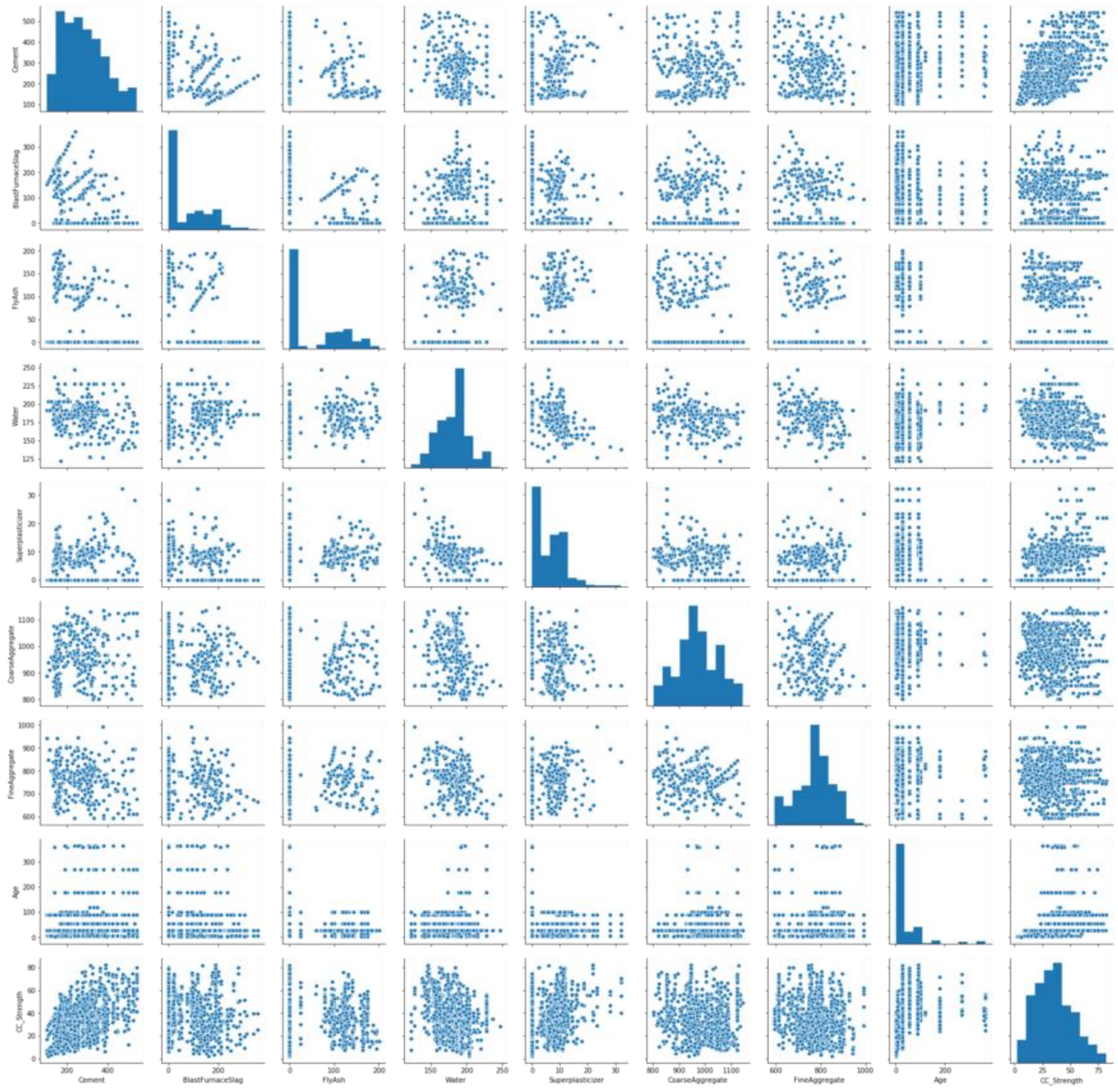
Let us check the correlations between the input features, this will give an idea about how each variable is affecting all other variables. This can be done by calculating Pearson correlations between the features as shown in the code below.



We can observe a high positive correlation between compressive Strength (CC_Strength) and Cement. this is true because strength concrete indeed increases with an increase in the amount of cement used in preparing it. Also, Age and Super Plasticizer are the other two factors influencing Compressive strength.

These correlations are useful to understand the data in detail, as they give an idea about how a variable is affecting the other. We can further use a **pairplot** in seaborn to plot pairwise relations between all the features and distributions of features along the diagonal.

```
sns.pairplot(cement)
```



The pair plot gives a visual representation of correlations between all the features.

Data preprocessing

Before we fit machine learning models on the data, we need to split the data into train, test splits. The features can be rescaled to have a mean of zero and a standard deviation of 1 i.e. all the features fall into the same range.

```
# define y
y=cement['Concrete Compressive Strength(MPa, megapascals) ']
#define x
x=cement[['Cement (kg in a m^3 mixture)',
       'Blast Furnace Slag (kg in a m^3 mixture)',
       'Fly Ash (kg in a m^3 mixture)', 'Water (kg in a m^3 mixture)',
       'Superplasticizer (kg in a m^3 mixture)',
       'Coarse Aggregate (kg in a m^3 mixture)',
       'Fine Aggregate (kg in a m^3 mixture)', 'Age (day)']]
#split data
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.7,random_st
ate=2529)
x_train.shape,x_test.shape,y_train.shape,y_test.shape.
```

Modeling:

After preparing the data, we can fit different models on the training data and compare their performance to choose the algorithm with good performance. As this is a regression problem, we can use RMSE (Root Mean Square Error) and R^2 score as evaluation metrics.

Linear Regression

We will start with Linear Regression since this is the go-to algorithm for any regression problem. The algorithm tries to form a linear relationship between the input features and the target variable i.e. it fits a straight line given by,

$$y = W * X + b = \sum_{i=1}^{n} w_i * x_i + b$$

Where w_i corresponds to the coefficient of feature x_i.

The magnitude of these coefficients can be further controlled by using regularization terms to the cost functions. Adding the sum of the magnitudes of the coefficients will result in the coefficients being close to zero, this variation of linear regression is called **Lasso** Regression. Adding the sum of squares of the coefficients to the cost function will make the coefficients be in the same range and this variation is called **Ridge** Regression. Both these variations help in reducing the model complexity and therefore reducing the chances of overfitting on the data.

```
# select model
from sklearn.linear_model import LinearRegression
model= LinearRegression()
# train model
model.fit(x_train,y_train)
# predict with model
y_pred=model.predict(x_test)
#model accuaracy
from sklearn.metrics import mean_absolute_error,mean_absolute_percentage_e
rror ,mean_squared_error
# mean_absolute_error
mean_absolute_error(y_test,y_pred)
# mean_absolute_percentage_error
mean_absolute_percentage_error(y_test,y_pred)
#mean_squared_error
mean_squared_error(y_test,y_pred)
```

Result:

Model name : linear Regression

```
# mean_absolute_error
mean_absolute_error(y_test,y_pred)
Output: 8.683767775410708
# mean_absolute_percentage_error
mean_absolute_percentage_error(y_test,y_pred)
Output: 0.3134440184320867
#mean_squared_error
mean_squared_error(y_test,y_pred)
Output: 120.40313453787677
```

Suggestion and Future Scope:

We have analysed the Compressive Strength Data and used Machine Learning to Predict the Compressive Strength of Concrete. We have used Linear Regression and its variations, Decision Trees and Random Forests to make predictions and compared their performance. Random Forest Regressor has the lowest RMSE and is a good choice for this problem. Also, we can further improve the performance of the algorithm by tuning the hyperparameters by performing a grid search or random search.

Future Prediction

```
#future prediction
# define X_new
x_new=x.sample()
#predict for X_new
model.predict(x_new)
```

Output: **array([52.23280615])**