### Predicting Compressive Strength of Concrete Using IBM Watson AutoAI Experiment

## Internship Title: RSIP Career Basic ML 131 Duration : 1 Month

**Project ID : SPS\_PRO\_287**

**Project Title : Predicting Compressive Strength of Concrete using IBM Watson AutoAI Experiment**

**Team : AJ**

**Team members: Aravind Joshi**

**Slot : June 15 - Slot (5)**

**Submitted by, Aravind Joshi**

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **Chapter No.** | **Title** | | **Page No.** |
| **1** | **Introduction** | | **3** |
|  | 1.1 | Overview | **3** |
|  | 1.2 | Purpose | **3** |
| **2** | **Literature Survey** | | **4** |
|  | 2.1 | Existing problem | **4** |
|  | 2.2 | Proposed Solution | **4** |
| **3** | **Theoretical Analysis** | | **5** |
|  | 3.1 | Block Diagram | **5** |
|  | 3.2 | Hardware/Software Designing | **5** |
| **4** | **Experimental Investigations** | | **6 - 18** |
| **5** | **Flow Chart** | | **19** |
| **6** | **Result** | | **20** |
| **7** | **Advantages and Disadvantages** | | **20** |
|  | 7.1 | Advantages | **20** |
|  | 7.2 | Disadvantages | **20** |
| **8** | **Applications** | | **21** |
| **9** | **Conclusion** | | **22** |
| **10** | **Future Scope** | | **23** |
| **11** | **Bibliography** | | **23** |

1. **INTRODUCTION**
   1. **Overview:**

Concrete is a material used in construction that has great versatility and which is used across the globe. Concrete has several advantages, including good compressive strength, durability, work ability, construction availability, and low cost. Determining accurate concrete strength is a major civil engineering problem. Test results of 28- day concrete cylinder represent the characteristic strength of the concrete that has been prepared and cast to form the concrete

## Purpose:

An ability to predict the compressive strength of concrete early allows constructors to quickly understand the concrete’s probable weaknesses and make a decision to manage a destruction process or continue with construction. Further, to the benefit of both user (and purchaser) and producer, reliably and rapidly predicting the results of a 28-day test would benefit all stakeholders as opposed to waiting the full, conventional, 28 days

# LITERATURE SURVEY

## Existing Problem:

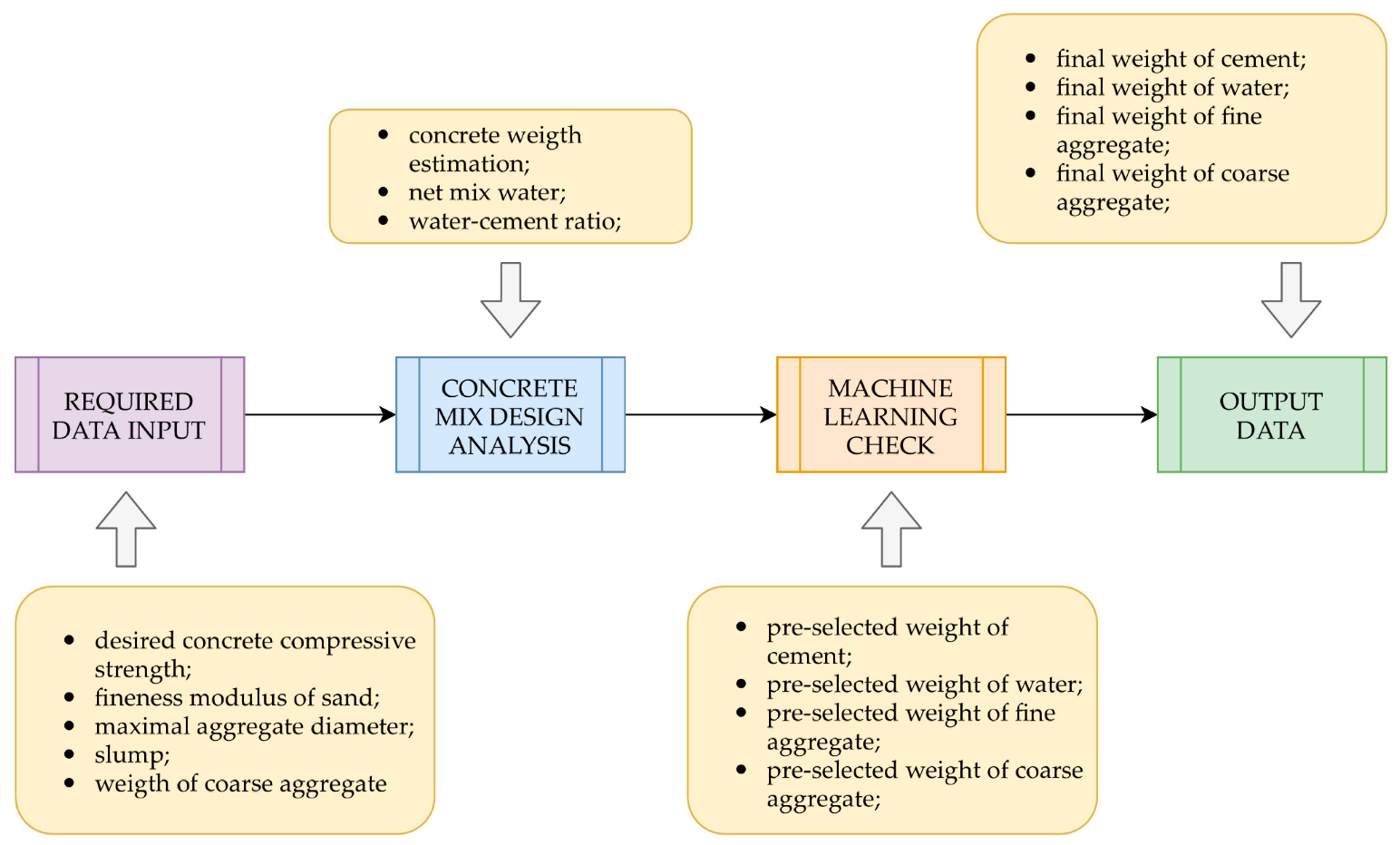
It is important to wait 28 days to ensure the quality control of the process, although it is very time consuming. Machine learning techniques are progressively used to simulate the characteristic of concrete materials and have developed into an important research area. This study proposed a comprehensive study using an advanced machine learning technique to predict the compressive strength of concrete from early age test results

## Proposed Solution:

We are building a Machine Learning model to predict the compressive strength of concrete using IBM Watson AutoAI Machine Learning Service. The model is deployed on IBM cloud to get scoring end point which can be used as API in mobile app or web app building. We are developing a web application which is built using node red service. We make use of the scoring end point to give user input values to the deployed model. The model prediction is then showcased on User Interface.

# THEORETICAL ANALYSIS

## Block Diagram:



* 1. **Hardware/ Software Designing**

This dataset is first tested by using various algorithms in our jupyter notebooks and then implemented in the IBM Cloud Platform. We upload our dataset in the cloud platform and choose the parameter to be predicted and we choose the number of algorithms and pipelines to be used. The cloud platform then predicts the best suited algorithm for our dataset with the rmse values. We can also compare the performance of other algorithms used. The Auto AI function in the IBM cloud aids in deployment of our final machine learning models. This helps us to implement and test our model for our dataset. We have then, created a node red app for our deployed model. This UI will help us predict the concrete strength in real time as we enter the details. This app aids in easy usage and better user interface.

# EXPERIMENTAL INVESTIGATIONS

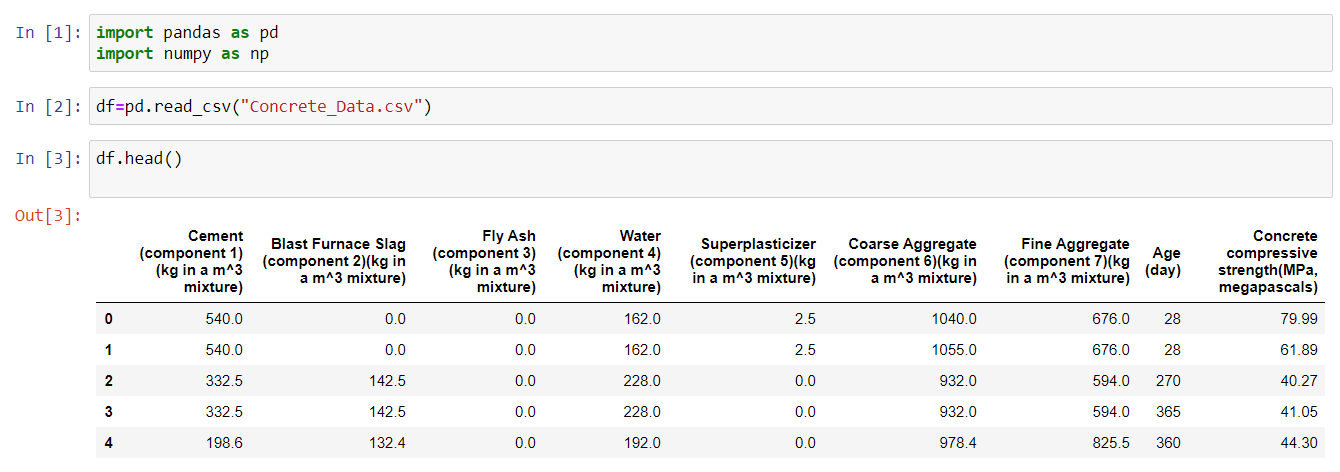
## Step1:- Data Collection: Jupyter Notebook:

We downloaded the dataset provided from Kaagle and did data pre- processing.

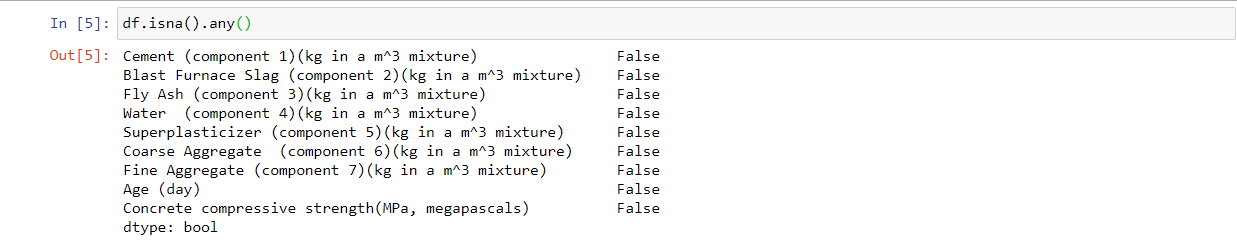
We applied various algorithms in our jupyter notebook to the dataset to find the best one. We need to predict the compressive strength of cement. Hence, we apply the regression algorithms to find the best fit algorithm for the given dataset.

## Data pre-processing and analysing:

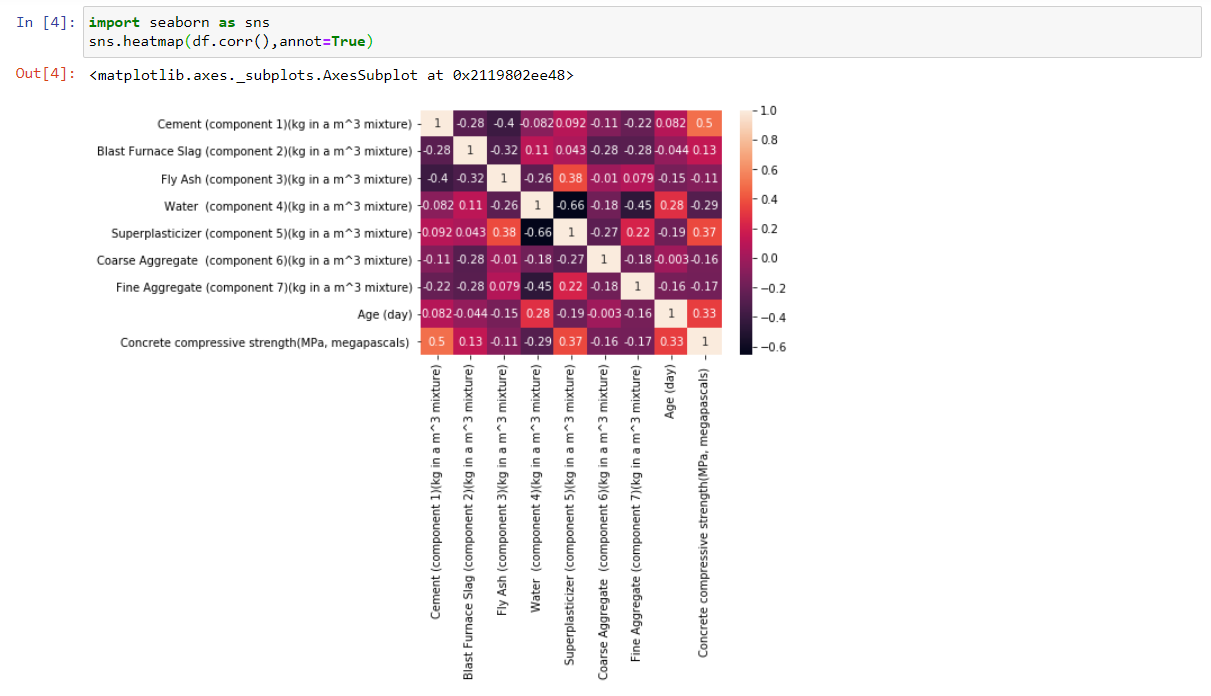
### We import the dataset and find the correlation between the given values. and find if there are any null values.



Here we also check for missing values



We also generate the correlation in the form of a heatmap for better understanding. We remove the parameters date and time as it does not contribute to the prediction value.

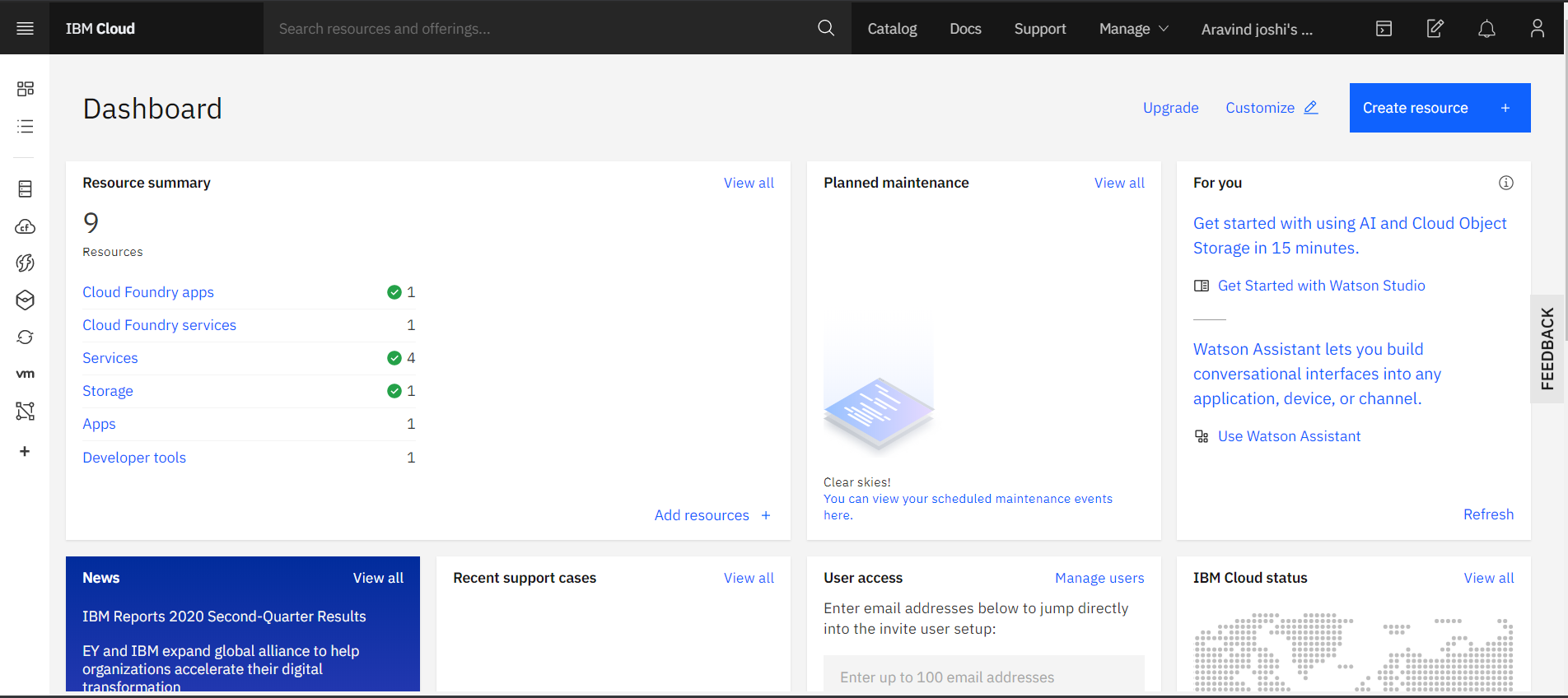


From the heat map, we can see that Intermittent and Snow depth are highly correlated. Next highest is Relative humidity. We then split the data as input and output parameters. We make Snow depth as y as it's the output that is to be predicted and make the factor such as temperature, humidity and intermittency as x values which are the inpu

## Step2: - IBM Cloud Account:

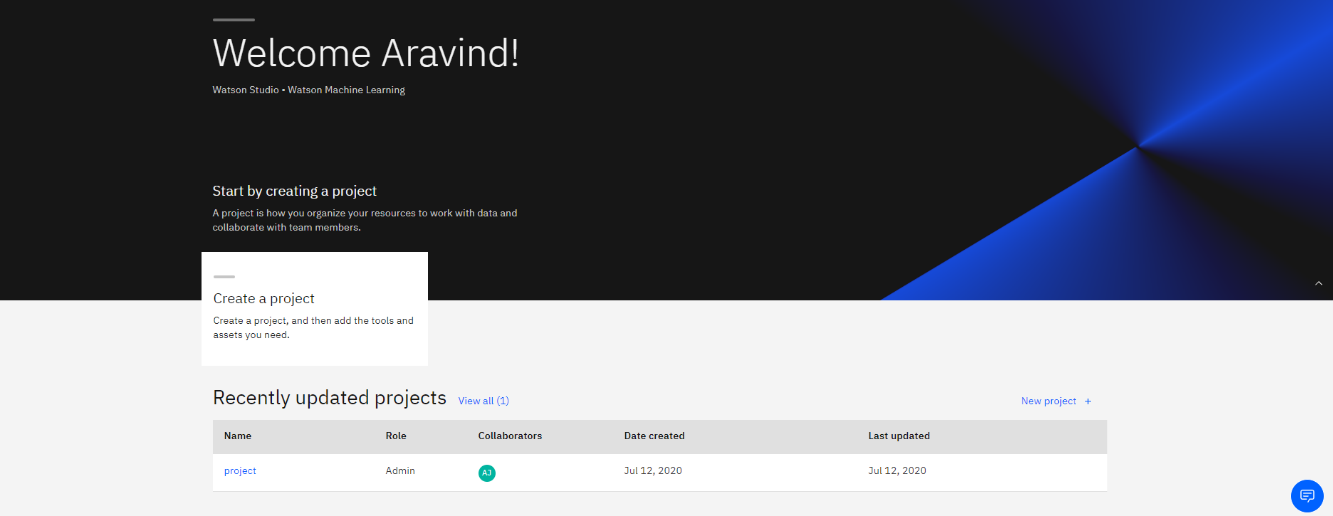
**Creating an account:**

We have successfully created an IBM Cloud account.



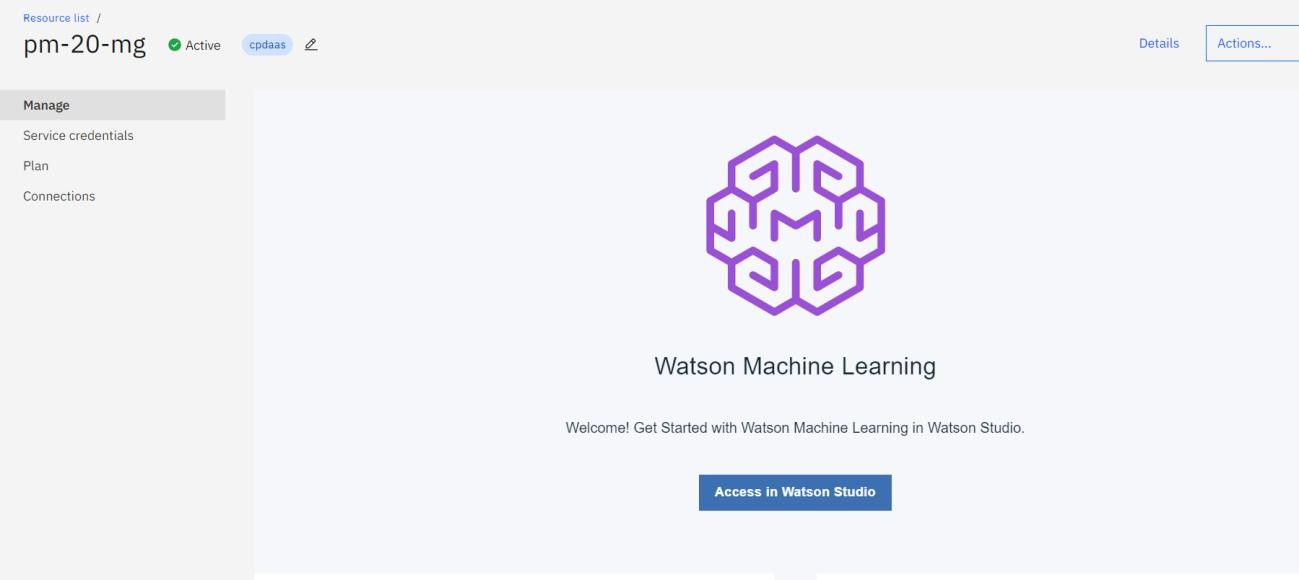
## Creating Watson Studio Platform

We have created a Watson studio platform to predict our data and to implement, deploy and test our model in the real time.



## Creating a ML Service:

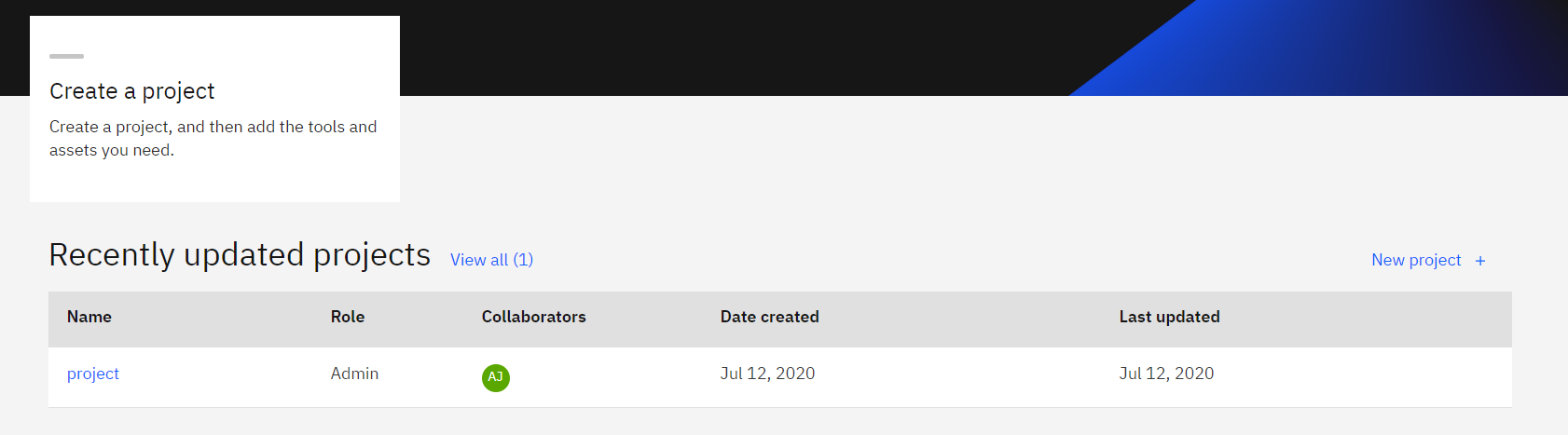
### We have created a ML service to load and process our dataset.



**Step3:- Model Building**

**Creating a project in Watson Studio:**

We have created a project in Watson studio namely Project.



IBM Watson Studio Desktop is a desktop client tool for solving your problems by analysing data with artificial intelligence. With Watson Studio Desktop, you can prepare data and build models on your desktop with visual drag and drop tools. You organize your resources for data analysis tasks in projects. Each [project](https://www.ibm.com/support/knowledgecenter/SSBFT6_1.1.0/wsd/projects.html?view=kc&projects) has its own directory on your computer. You can choose a standard project or to import a project that was previously exported from Watson Studio Desktop.

## Auto AI Experiment in add Projects and set up AI environment:

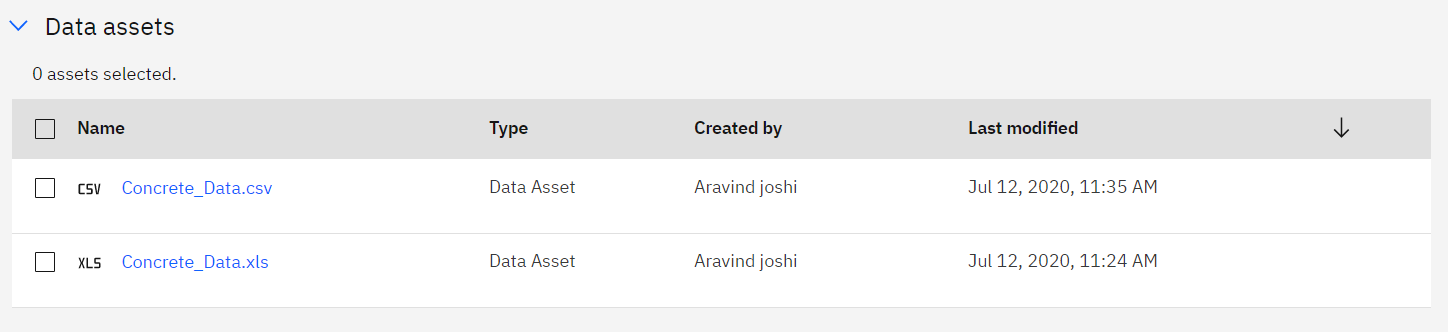
We have created an auto AI experiment called Intern. AutoAI is available within [IBM](https://www.ibm.com/in-en/cloud/watson-studio) [Watson Studio](https://www.ibm.com/in-en/cloud/watson-studio) with one-click deployment through [Watson Machine Learning](https://www.ibm.com/in-en/cloud/machine-learning). To help simplify an AI lifecycle management, AutoAI automates:

* + Data preparation
  + Model development
  + Feature engineering
  + Hyper-parameter optimization



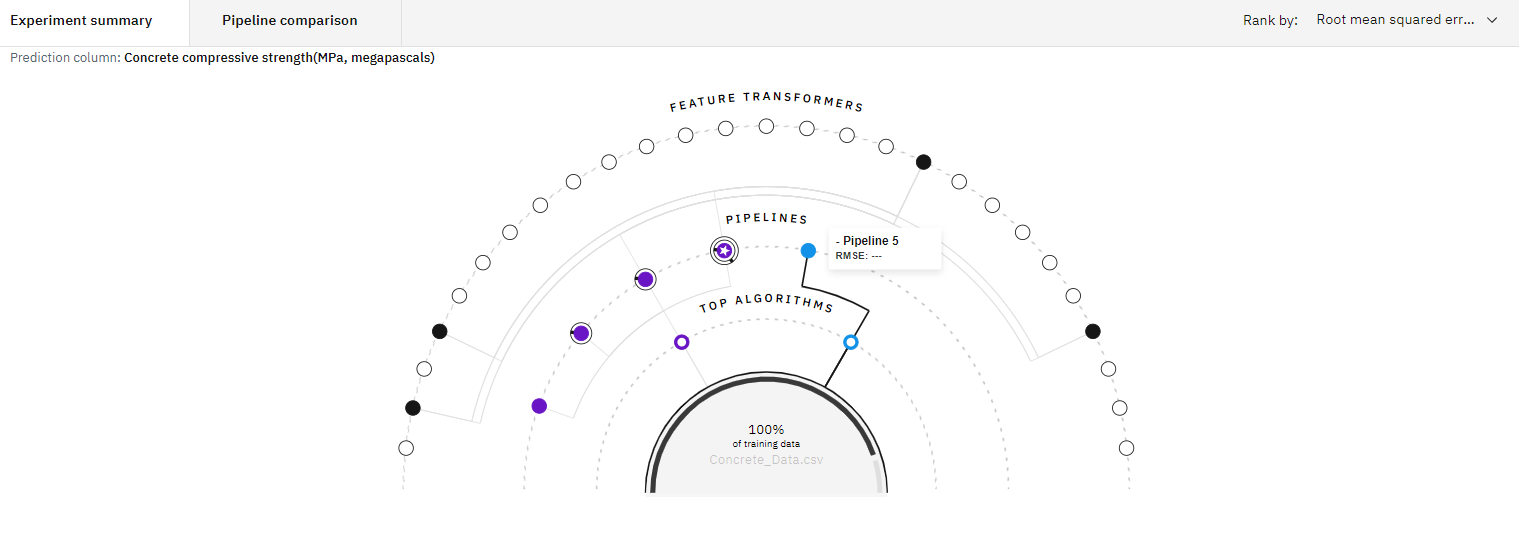
## Import Dataset:

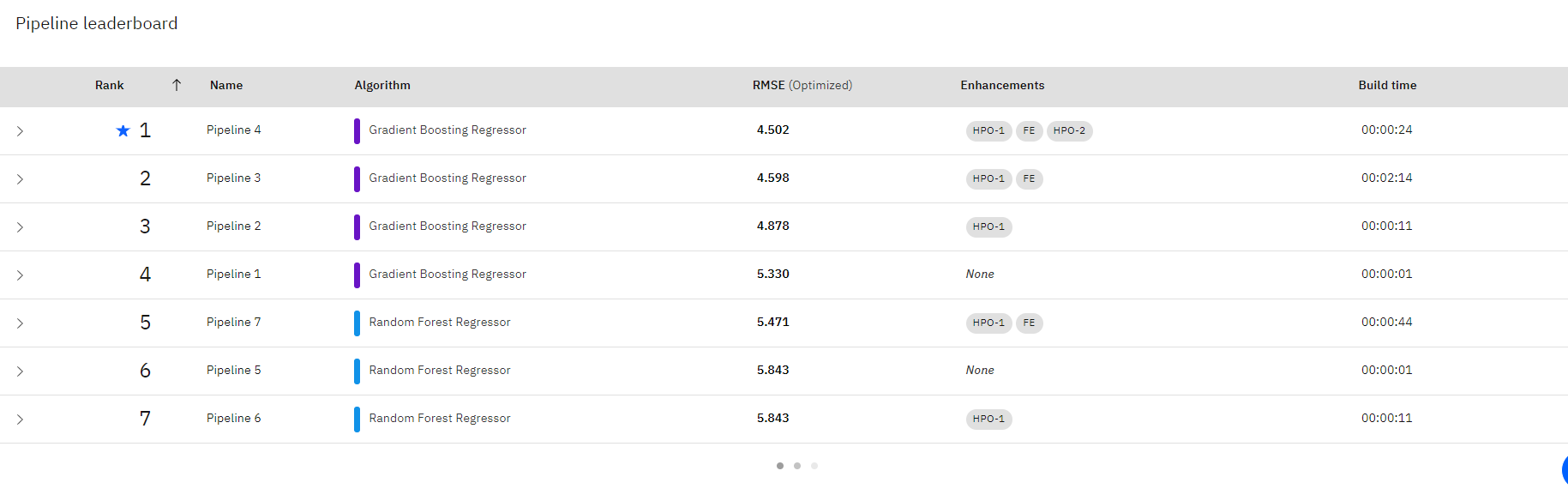
We have then imported the dataset in the name of copy.csv in the assests section.



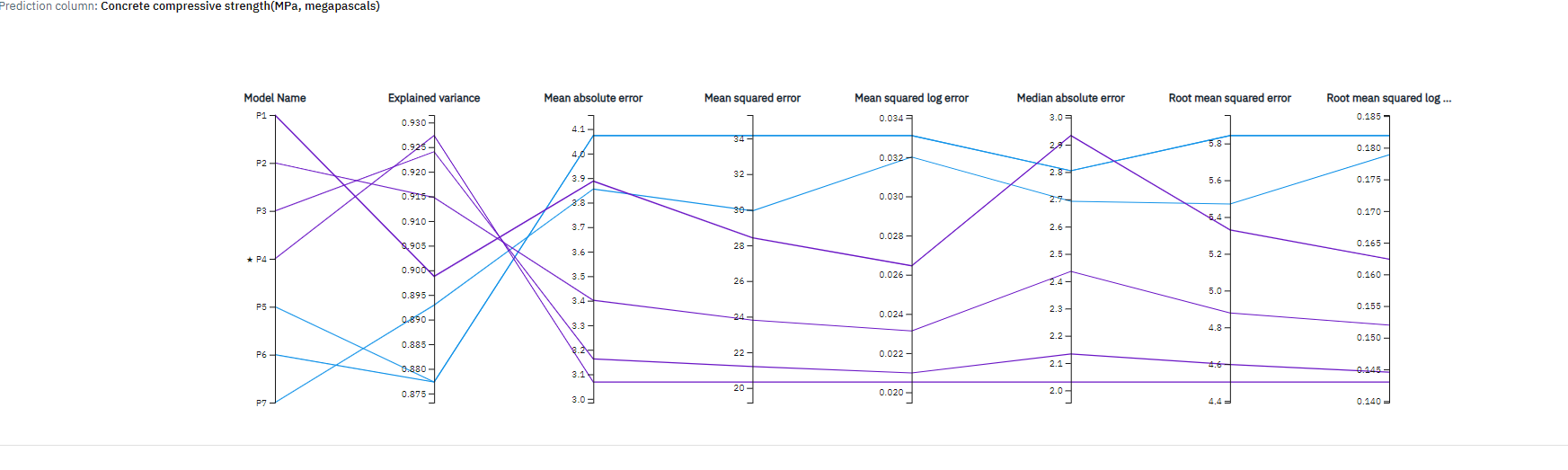
## Run the model and select the pipeline:

We then run the model, by choosing what to predict, the ratio of train and testing dataset and the number of pipelines to be used. We choose the default 90:10 ratio for training and testing and we chose 8 pipelines.



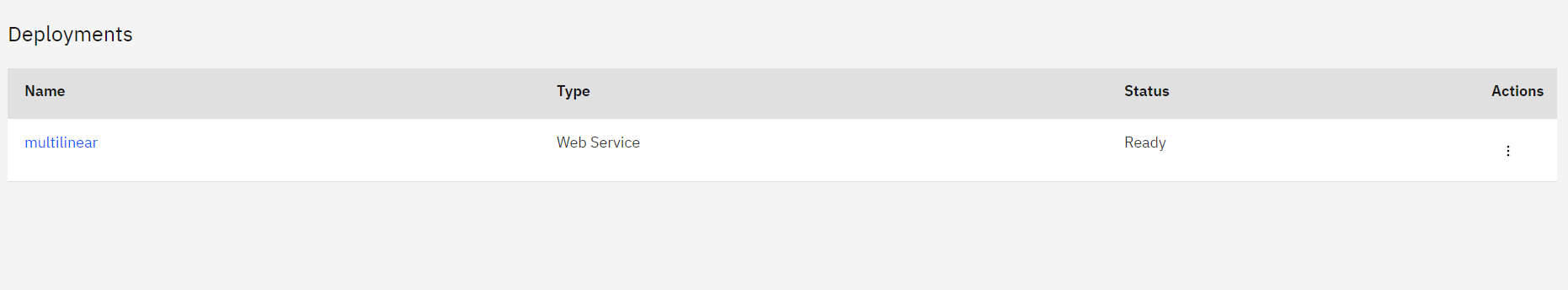


As we can see, the random forest algorithm ranks first as the best algorithm for the given dataset. It has the most less error, (i.e.) rmse value in comparison to all the other algorithms.



## Deploy and test the model in Watson Studio:

We then save this model and deploy it in the Watson studio in the name "Multilinear".



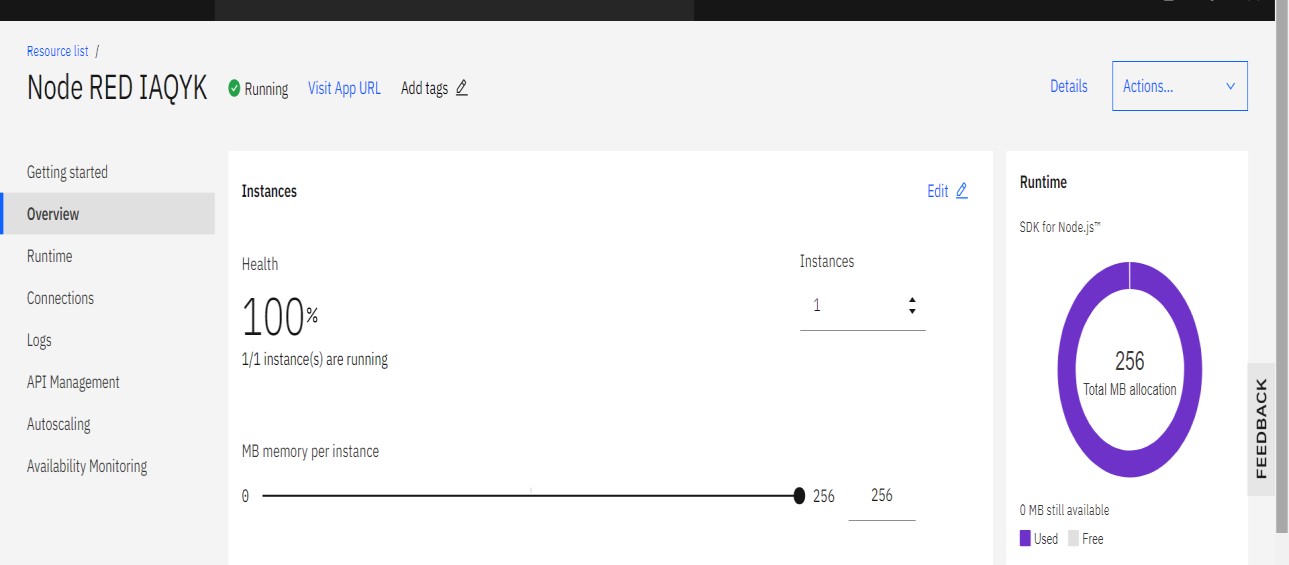
We can see that our model is successfully deployed and ready to implement and test. We need to click on the model and it will direct us to a page where we can find the model overview, implementation and test. We test our model before creating our app.



## Step 4:- Application building:

**Create a Node Red service:**

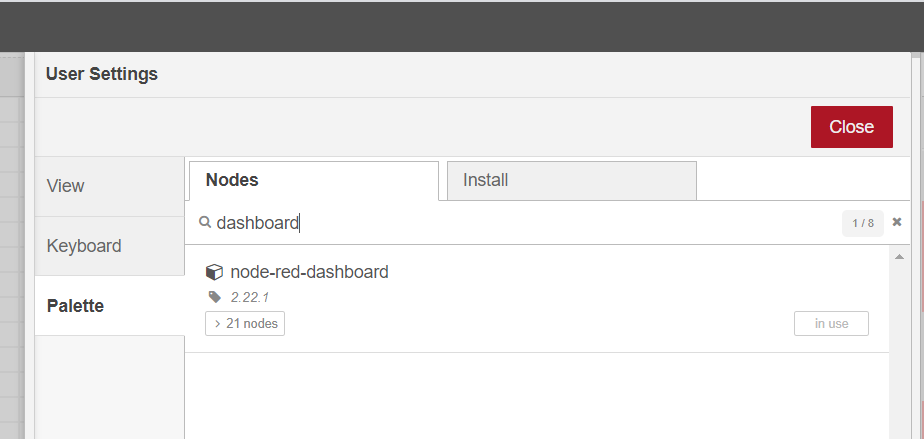
We have created a Node Red app and we can see that the app is running.



Node-RED provides us a browser-based flow editor that makes it easy for us to wire together flows using the wide range of nodes in the palette. Flows can be then deployed to the runtime in a single-click. JavaScript functions can be created within the editor using a rich text editor. A built-in library allows you to save useful functions, templates or flows for re-use.

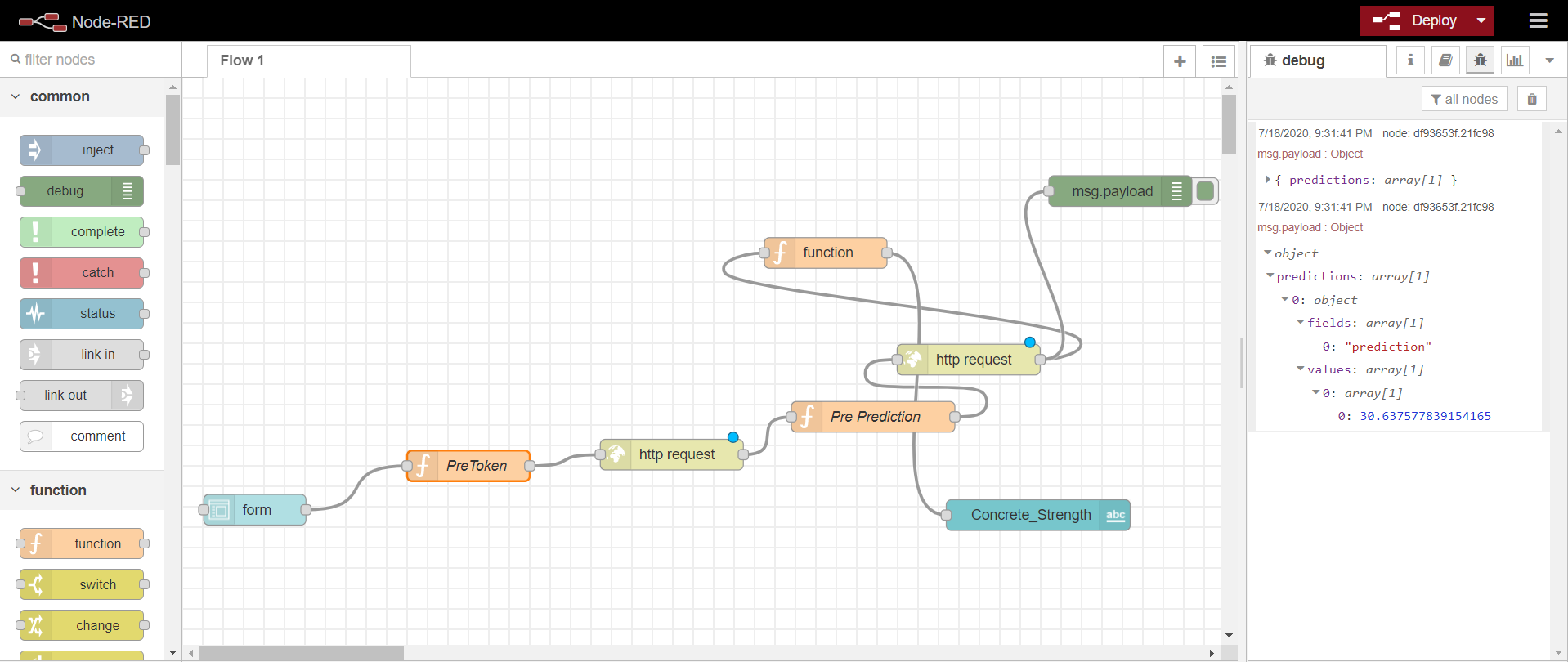
## Install dashboard palette:

We have installed dashboard palette in the node red app and we use those nodes to build our app.

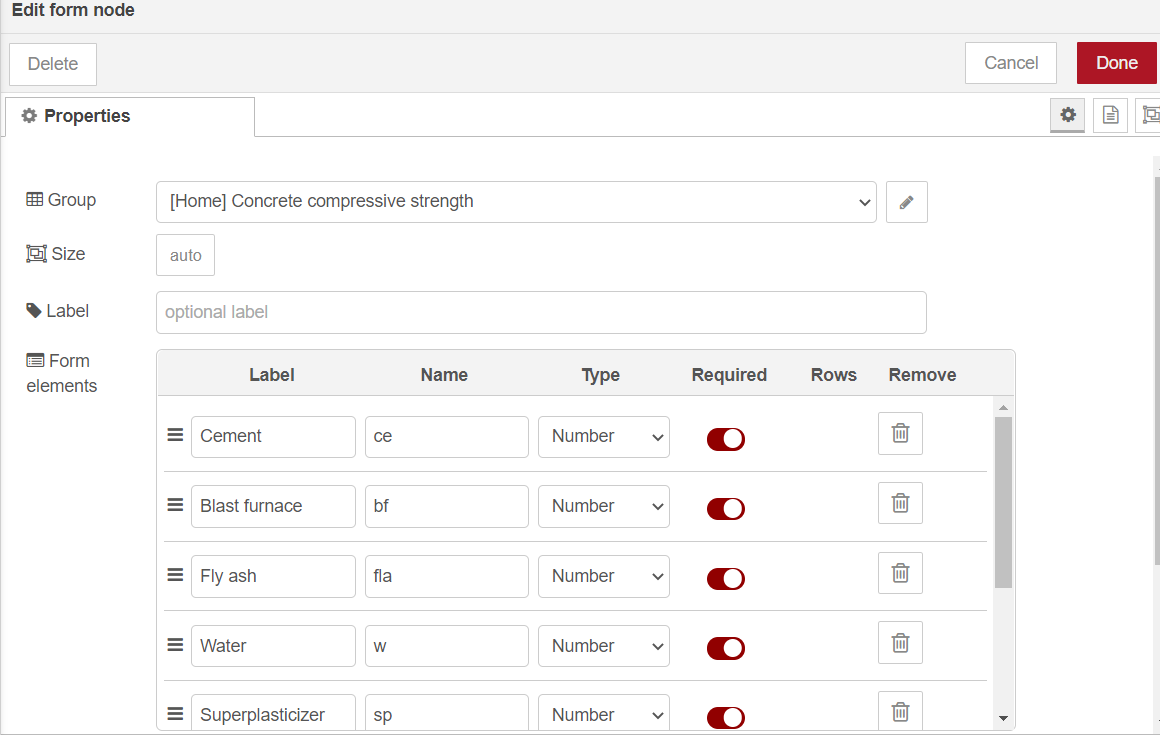


## Building UI with Node Red:

We connect the following nodes in our node red flow.



**Form node :** In the form node, we give the titles and datatypes of our inputs.



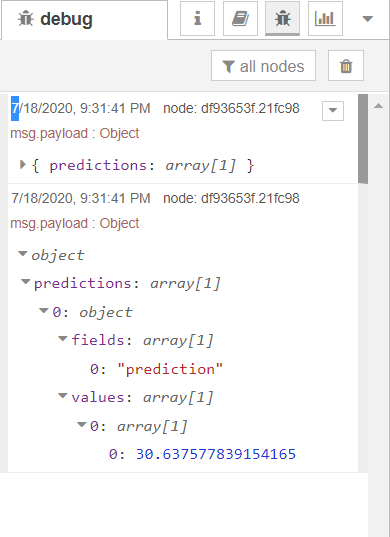
We have the data types as number for temperature, humidity and Intermittency. We have given the data type of date and time to be text.

**Pre Token:** The pre token is a function node. A JavaScript function to run against the messages being received by the node. The messages are passed in as a JavaScript object called msg. We link the api key of our deployment in this node. We get the input values from the user for the input parameters needed and then pass it on to our next node.

**Http request:** This node sends the http request and returns the response. The body of the response. The node can be configured to return the body as a string, attempt to parse it as a JSON string or leave it as a binary buffer.

**Pre Prediction :** The pre prediction node is also a function node. This node links our instance id to access the deployment of our model. The msg.payload in the code sends our fields as a dictionary format to our output node.

In our next http node we link the url of our app and in the next function node we link the path of our predicted output value from the debug message part. This helps us to view our output in our web page.

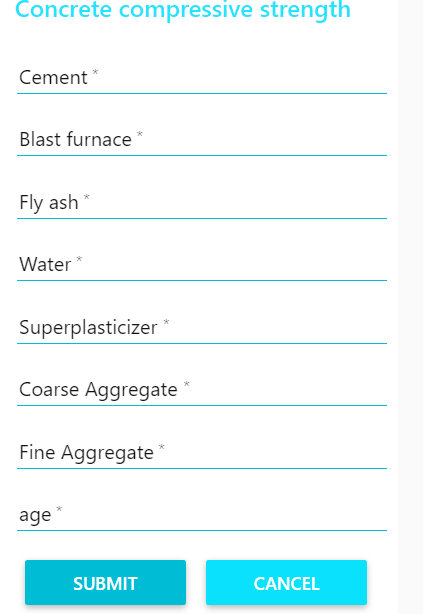


### **msg.payload :** This node displays the value of our prediction (i.e) the depth of the snow in our case. This node displays selected message properties in the debug sidebar tab and optionally the runtime log. By default it displays

msg.payload, but can be configured to display any property, the full message or the result of a JSON data expression.

**Deploy the app and run:**

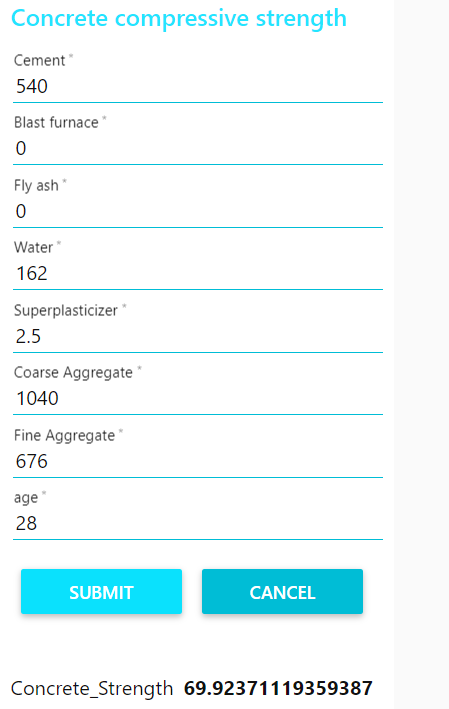
We then deploy our app and we load our ui web page.



### This is our web page, we then enter the values to predict the concrete strength.

### 

We have given all the input values and our app has predicted the concrete strength to be 41.356. This indicated less snow depth and hence avalanche is unlikely to occur.

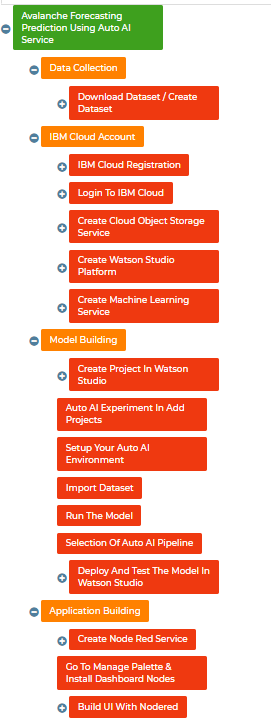


We changed our values and reduced the number of days and we can wee that the concrete strength has raised to be 69.923 indicating stronger build.

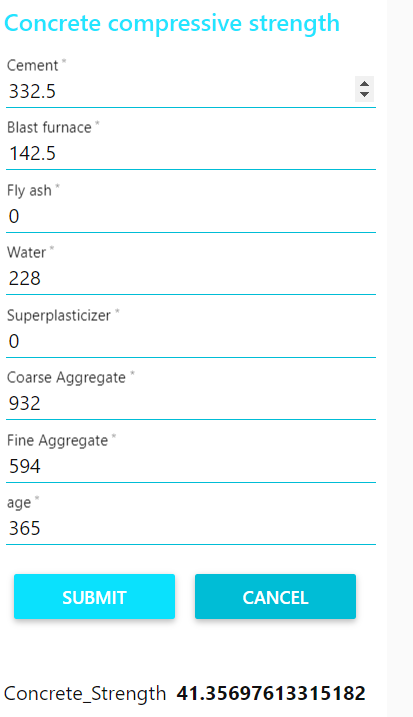
**The url for our project is:**

<https://node-red-pvegn.eu-gb.mybluemix.net/ui/#!/0?socketid=xE8V8Wh-VnONjJePAABK>

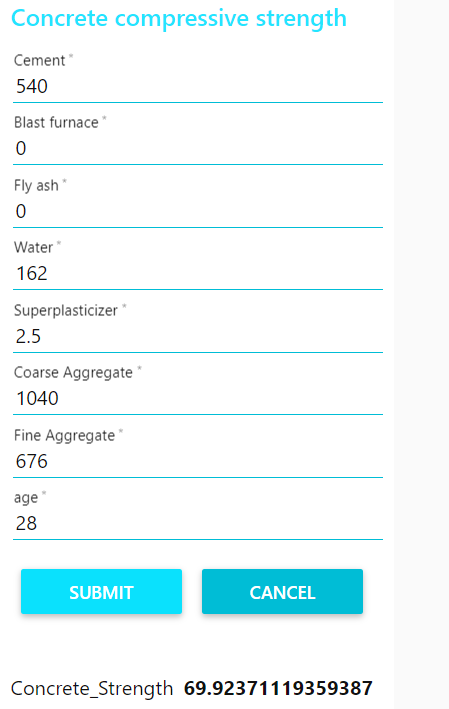
# FLOW CHART



1. **RESULT**



As we can see, our app gives the output cement strength if we enter the input values. If the number of days is more the concrete strength is less. If the number of days is lesser concrete strength is higher



# ADVANTAGES AND DISADVANTAGES

## Advantages:

* + - Our web app predicts the concrete strength with a great accuracy.
    - Our web app is easy to access and provides us a good user interface.
    - This helps people from any region to access the app and predict the compressive concrete strength with the help of the input parameters.
    - The app gives results immediately without any delay.

## Disadvantages:

* + - The user entering the details should know the details of the input parameters in his region.
    - It would be good if the app could automatically get the user's location details and get the weather details in that region and predict the compressive concrete strength according to the raw materials available there

# APPLICATIONS

* + - This might save the life of the surrounding people and also their animals.
    - This app will also help the builders to curb additional cost by pre calculating exact costs.
    - This app will create an awareness among the people regarding the compressive strength of concrete and helps them act accordingly.
    - It also helps new builders to check for the concrete strength before even actually starting the project.

# CONCLUSION

This app presents a simple machine learning model to predict the concrete compressive strength from the early age test results. In this study, the concrete strength characteristic with age is modeled by a multiple linear regression (MLR) mathematical equation. Early age test data are being used in this case to get reliable values of the 28 day strength prediction. Herein, a simple and practical approach has been described for prediction of 28-day compressive strength of concrete and the proposed technique can be used as a reliable tool for assessing the strength of concrete from quite early test results. This will help in making quick decision at site and reduce delay in the execution time of large civil construction projects.. We have tested our dataset in our jupyter notebooks first and have decided the best algorithm that works in this dataset. Random forest algorithm is best suited for the given dataset with less error and higher r2 square values. We have also used the IBM Auto AI services. We have imported the dataset and the cloud service has also predicted the Random Forest Algorithm to be the best algorithm suited for this dataset.. We can also see the comparison of various pipelines used in the IBM cloud. We have also deployed the model with the help of the node red application. This provides a better user interface and helps the user to understand the results in a better and easy way. The user using the app have to enter the details of the required parameters such as days, water, blast furnace etc to predict the compressive concrete strength. Thus, we have predicted the compressive concrete strength with a great accuracy and also have deployed it as an app providing a better user interface.

# FUTURE SCOPE

My app can predict the compressive strength with a great accuracy. In future, our model can be expanded to a mobile app. We can include various features to select the particular region in the map for which the regional raw material availability could be known and concrete strength can be predicted. We can also link our app to the geographical locations of the user. We can also make the app interactive for various builders to analyze on a daily basis for the people living in the risk region and can send them daily updates or notifications regarding the risk percentage without them manually opening the app too. Thus, this will become a default mobile notification along with the various software’s used to architects and builders making it much easier for the users to be aware and safe.

# BIBILOGRAPHY

<https://www.sciencedirect.com/science/article/pii/S1687404813000783><https://www.getprepared.gc.ca/cnt/hzd/vlchs-en.aspx>

<https://cdn.ymaws.com/concretenz.org.nz/resource/resmgr/docs/conf/2019/s2b_p2.pdf>