PREDICTING THE COMPRESSIVE STRENGTH OF CONCRETE

A PROJECT REPORT

SUBMITTED BY:

MANASA MOTEPALLI

TABLE OF CONTENTS

1. INTRODUCTION

1.1 OVERVIEW

This is an internship offered by smartbridge on MACHINE LEARNING.
We are well trained with python, Data Pre-Processing, Supervised algorithms, Unsupervised algorithms, flask integration and access the IBM CLOUD.
The project assigned to me is an individual project of topic 'PREDICTING COMPRESSIVE STRENGTH OF CONCRETE USING IBM WATSON AUTO-AI EXPERIMENT'.
We use the database given to us and the relevant algorithm to obtain the model using Auto-AI.

In this project there are 4 tasks:

1. Data collection
   - Download Dataset/Create Dataset
2. IBM Cloud account
   - IBM Cloud Registration
   - Login To IBM Cloud
   - Create Cloud Object Storage Service
   - Create Watson Studio Platform
   - Create Machine Learning service
3. Model building
   - Create Project In Watson Studio
   - Auto AI Experiment In Add Experiment
   - Setup Your Auto AI Environment
   - Import Dataset
   - Run The Model
   - Selection of Auto-AI Pipeline
   - Deploy And Test The Model In Watson Studio
4. Application building
   - Create Node-Red Service
   - Build UI With Node-Red

The skills gained are:
Python, Python for Data Analysis, Python for Data Visualization, Exploratory Data Analysis, IBM Cloud, IBM WATSON.

1.2 PURPOSE

The purpose of this project is to predict the compressive strength of concrete using IBM Watson Studio. The purpose of this project is to learn about Machine Learning and the facilities available in IBM cloud and explore its wide range of services. The IBM Cloud includes Infrastructure as a service, Software as a service and Platform as a service. IBM offers tools for cloud-based collaboration, development and test, application development, analytics, business-to-business integration, and security.

The purpose of this project is to gain knowledge and insights of Machine Learning and building the model for predicting the compressive strength of concrete. Concrete is used in construction field. Model identifies trend and patterns. Using the model no human Intervention is needed. Machine Learning makes it easy to handle multi-dimensional and multi-variety data and the web application built can be used by everyone.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

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 work.
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.

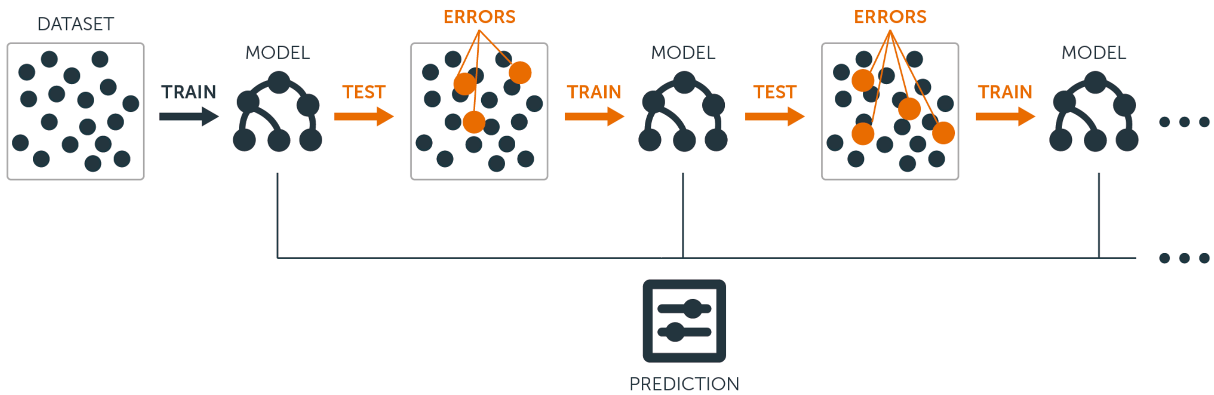
2.2 PROPOSED SOLUTION

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.
We are building a Machine Learning model to predict the compressive strength of concrete using IBM Watson Auto-AI 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.

3. THEORETICAL ANALYSIS

3.1 BLOCK DIAGRAM

The algorithm used in this project is **GRADIENT BOOSTING REGRESSOR ALGORITHM**



3.2 HARDWARE/SOFTWARE DESIGNING

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

The idea of gradient boosting originated in the observation by Leo Breiman that boosting can be interpreted as an optimization algorithm on a suitable cost function. Explicit regression gradient boosting algorithms were subsequently developed by Jerome H. Friedman, simultaneously with the more general functional gradient boosting perspective of Llew Mason, Jonathan Baxter, Peter Bartlett and Marcus Frean. The latter two papers introduced the view of boosting algorithms as iterative functional gradient descent algorithms. That is, algorithms that optimize a cost function over function space by iteratively choosing a function (weak hypothesis) that points in the negative gradient direction. This functional gradient view of boosting has led to the development of boosting algorithms in many areas of machine learning and statistics beyond regression and classification.

The gbm package uses a predict() function to generate predictions from a model, similar to many other machine learning packages in R. When you see a function like predict() that works on many different types of input (a GBM model, a RF model, a GLM model, etc), that indicates that predict() is an "alias" for a GBM-specific version of that function.

To begin with, gradient boosting is an ensembling technique, which means that prediction is done by an ensemble of simpler estimators. While this theoretical framework makes it possible to create an ensemble of various estimators, in practice we almost always use GBDT — gradient boosting over decision trees.

THE CLASSIFIER USED FOR GRADIENT BOOSTING ALGORITHM IS:

GradientBoostingClassifier(*, *loss='deviance'*, *learning_rate=0.1*, *n_estimators=100*, *subsample=1.0*, *criterion='friedman_mse'*, *min_samples_split=2*, *min_samples_leaf=1*, *min_weight_fraction_leaf=0.0*, *max_depth=3*, *min_impurity_decrease=0.0*, *min_impurity_split=None*, *init=None*, *random_state=None*, *max_features=None*, *verbose=0*, *max_leaf_nodes=None*, *warm_start=False*, *presort='deprecated'*, *validation_fraction=0.1*, *n_iter_no_change=None*, *tol=0.0001*, *ccp_alpha=0.0*)

THE PARAMETERS ARE:

loss{‘deviance’, ‘exponential’}, default=’deviance’

loss function to be optimized. ‘deviance’ refers to deviance (= logistic regression) for classification with probabilistic outputs. For loss ‘exponential’ gradient boosting recovers the AdaBoost algorithm.

earning_ratefloat, default=0.1

learning rate shrinks the contribution of each tree by learning_rate.There is a trade-off between learning_rate and n_estimators.

n_estimatorsint, default=100

The number of boosting stages to perform. Gradient boosting is fairly robust to over-fitting so a large number usually results in better performance.

subsamplefloat, default=1.0

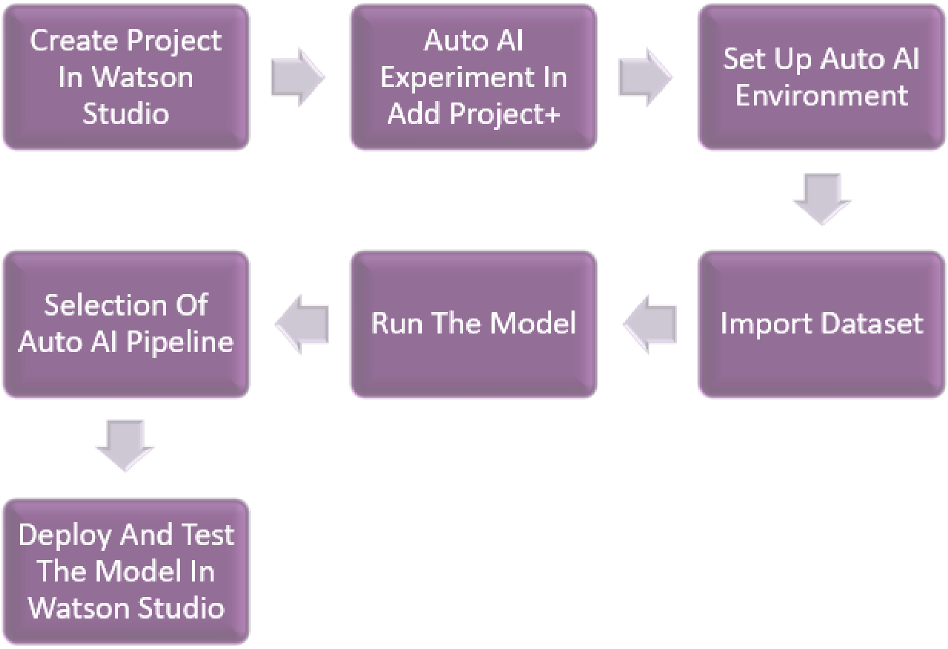
The fraction of samples to be used for fitting the individual base learners. If smaller than 1.0 this results in Stochastic Gradient Boosting. Subsample interacts with the parameter n_estimators. Choosing sub-sample < 1.0 leads to a reduction of variance and an increase in bias.

criterion{‘friedman_mse’, ‘mse’, ‘mae’}, default=’friedman_mse’

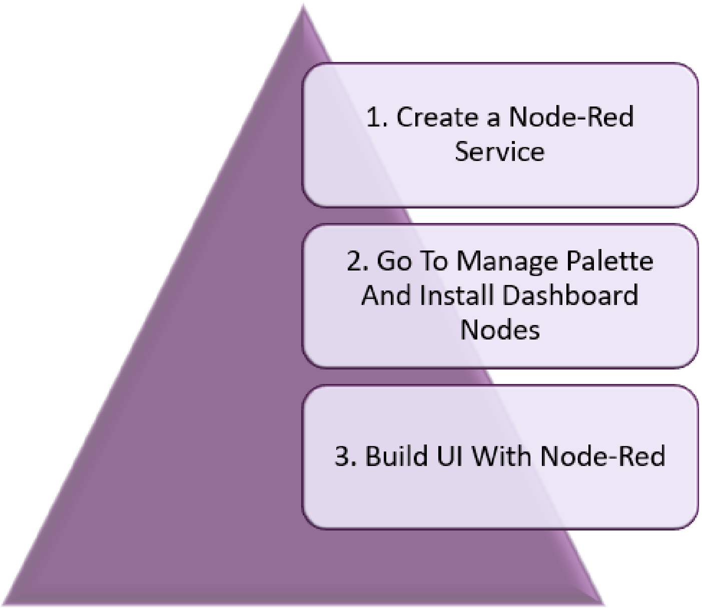
The function to measure the quality of a split. Supported criteria are ‘friedman_mse’ forthe mean squared error with improvement score by Friedman, ‘mse’ for mean squared error, and ‘mae’ for the mean absolute error. The default value of ‘friedman_mse’ is generally the best as it can provide a better approximation in some cases.

4. EXPERIMENTAL INVESTIGATIONS

- **AUTO AI EXPERIMENT WORKFLOW**



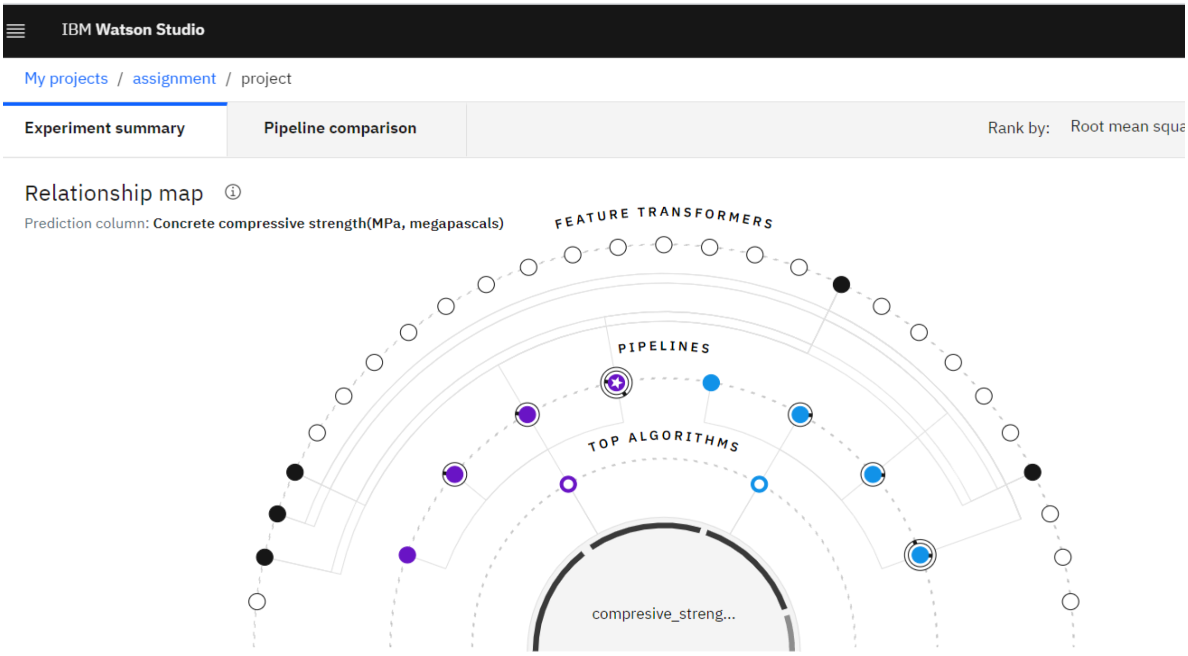
- **APPLICATION BUILDING WORKFLOW**

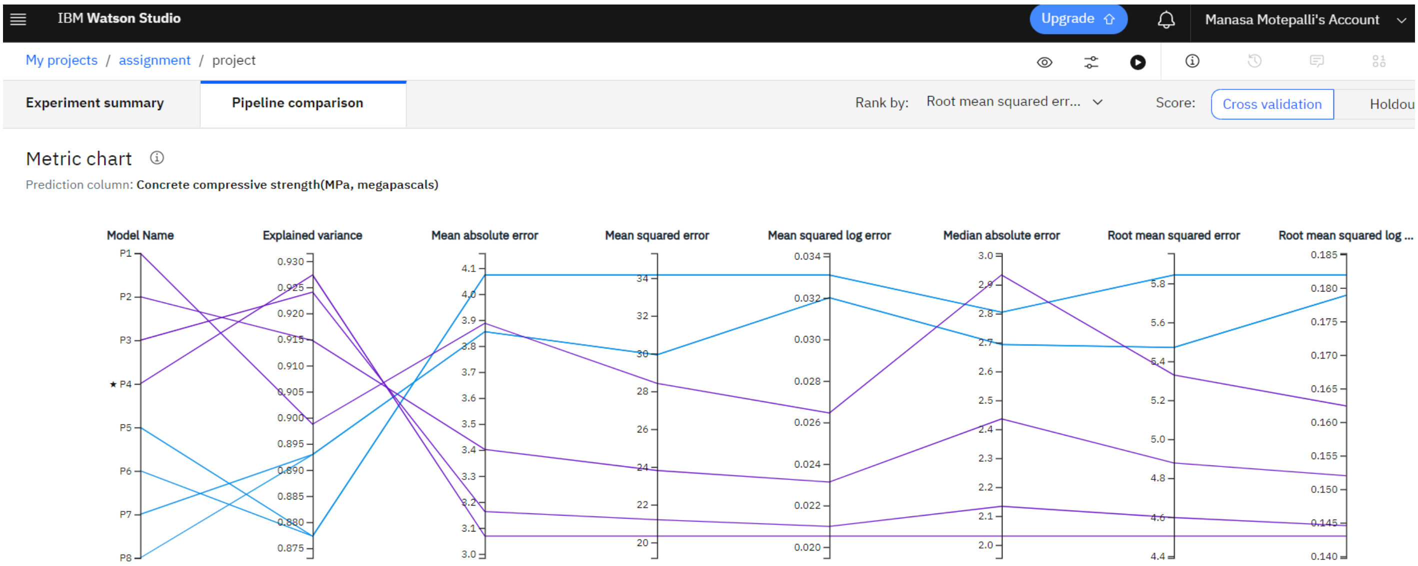


-

1. AUTO AI EXPERIMENT

- In Machine Learning, we create a project in IBM Watson Studio and we add an Auto AI Experiment to the project.
- We need to import our dataset to calculate the required output. Hence, we import/upload the 'COMPRESSIVE_STRENGTH_CONCRETE' csv file and give 'Concrete Compressive Strength' column of the dataset as the prediction column.
- Run the model and we get our required Auto AI Pipeline. We get the Relationship Map and Pipeline Comparison. We have to save the Auto AI pipeline as a model.
- We should deploy the saved model.
- Then we get to test the deployed model.



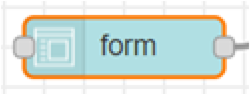


Pipeline leaderboard

Rank	Name	Algorithm	RMSE (Optimized)	Enhancements	Build time
★ 1	Pipeline 4	Gradient Boosting Regressor	**4.502**	HPO-1 FE HPO-2	00:00:19
2	Pipeline 3	Gradient Boosting Regressor	**4.598**	HPO-1 FE	00:01:53
3	Pipeline 2	Gradient Boosting Regressor	**4.878**	HPO-1	00:00:10
4	Pipeline 1	Gradient Boosting Regressor	**5.330**	*None*	00:00:01
5	Pipeline 7	Random Forest Regressor	**5.471**	HPO-1 FE	00:00:39
6	Pipeline 8	Random Forest Regressor	**5.471**	HPO-1 FE HPO-2	00:00:33
7	Pipeline 5	Random Forest Regressor	**5.843**	*None*	00:00:01
8	Pipeline 6	Random Forest Regressor	**5.843**	HPO-1	00:00:09

2. APPLICATION BUILDING (NODE-RED)

- From the catalog of IBM Cloud we should install Node-Red App.
- In this, we have 'Nodes' to the left side of the page and we use these nodes to create a 'flow'.
- We need to install 'Dashboard Nodes' by going to the 'Manage Palette' section.
- The nodes which we use in this project are:
- Form node

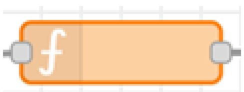


Adds a form to user interfaceHelps to collect multiple value from the user on submit button click as an object in msg.payload.Multiple input elements can be added using add elements button

- Timestamp node

Injects a message into a flow either manually or at regular intervals. The message payload can be a variety of types, including strings, JavaScript objects or the current time.

- function node



A JavaScript function block to run against the messages being received by the node. The messages are passed in as a JavaScript object called msg.By convention it will have a msg.payload property containing the body of the message. The function is expected to return a message object (or multiple message objects), but can choose to return nothing in order to halt a flow

- http request node

Sends HTTP requests and returns the response.

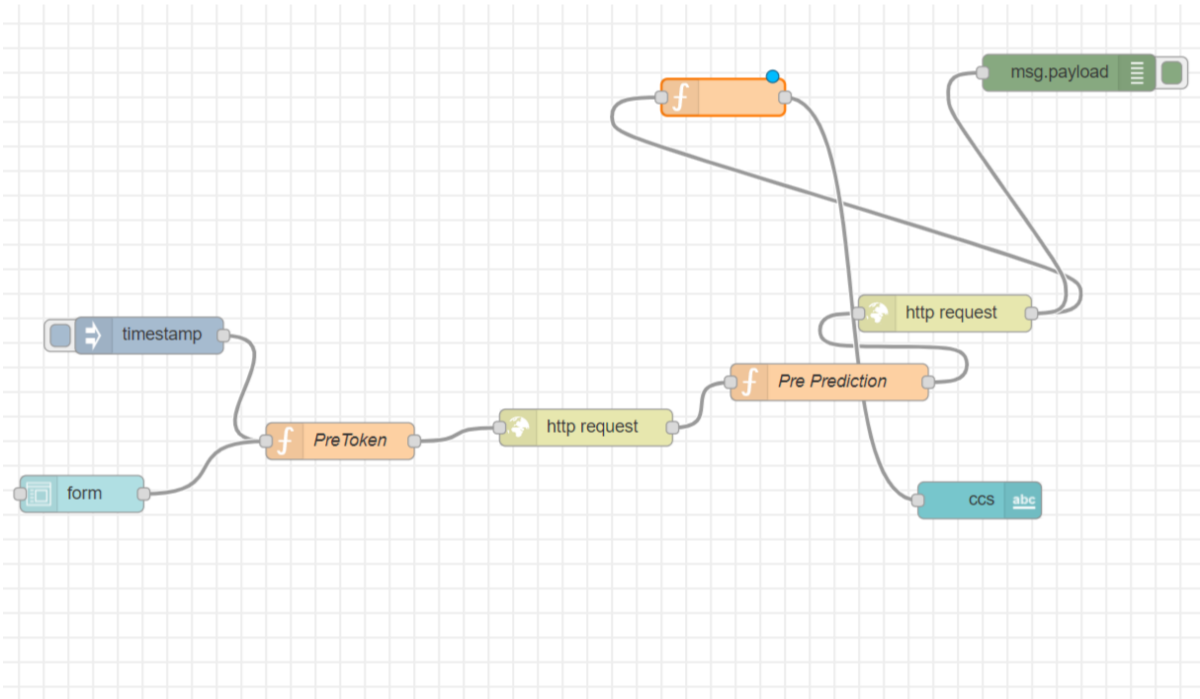
- Text node

Will display a non-editable text field on the user interface.Each received msg.payload will update the text based on the provided Value Format.The Value Format field can be used to change the displayed format and can contain valid HTML and Angular filters

- Debug 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 JSONata expression.

5. FLOWCHART

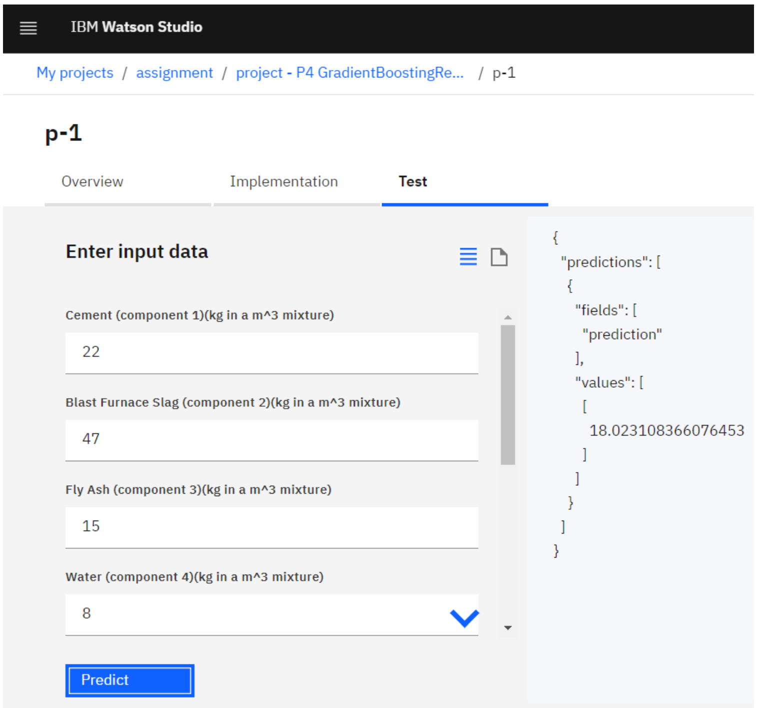


- FORM NODE is used to give the input columns
- TIMESTAMP NODE is not that required
- PRE-TOKEN FUNCTION NODE is used to enter the 'API-KEY' and set the input columns to global.
- HTTP REQUEST NODE is used to give the 'scoring-end-Point url'.
- PRE-PREDICTION NODE is used to enter the 'INSTANCE-ID' and get the input columns which were set to global.
- In FUNCTION NODE we give the path of the output obtained.
- TEXT NODE is used to give a title to the output.
- DEBUG NODE is used to get the output.
- After giving all the inputs to the respective nodes, we should deploy the flow.
- Then we have to launch the dashboard where we enter random values and get the predicted output for the entered values.
- The input in the form node is passed to pre-token node.
- The pre-token node is connected to http-request node passing its input.
- The http-request node is connected to pre-prediction node.
- The pre-prediction node is connected to http-request node containing scoring-end-point url.
- The http-request node is connected to function node as well as debug node
- The function node is connected to the text node.

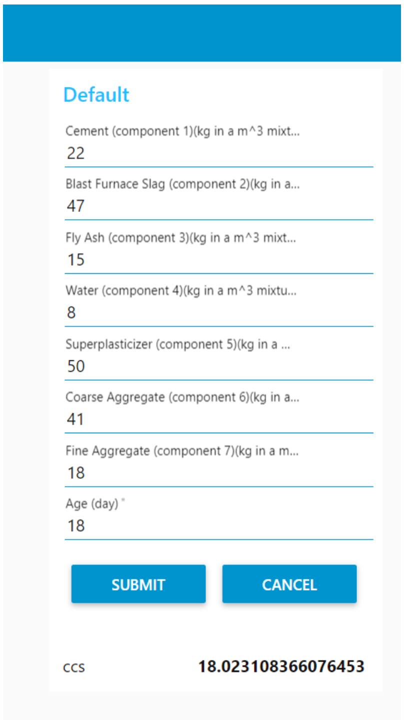
6. RESULT

By creating the flow, we obtained the required output. The predicted value obtained in IBM Watson Studio and the predicted value obtained in Node-Red matches when we give the same input values in both the applications. Hence, the flow is successfully built without any errors.

- **CASE-1**

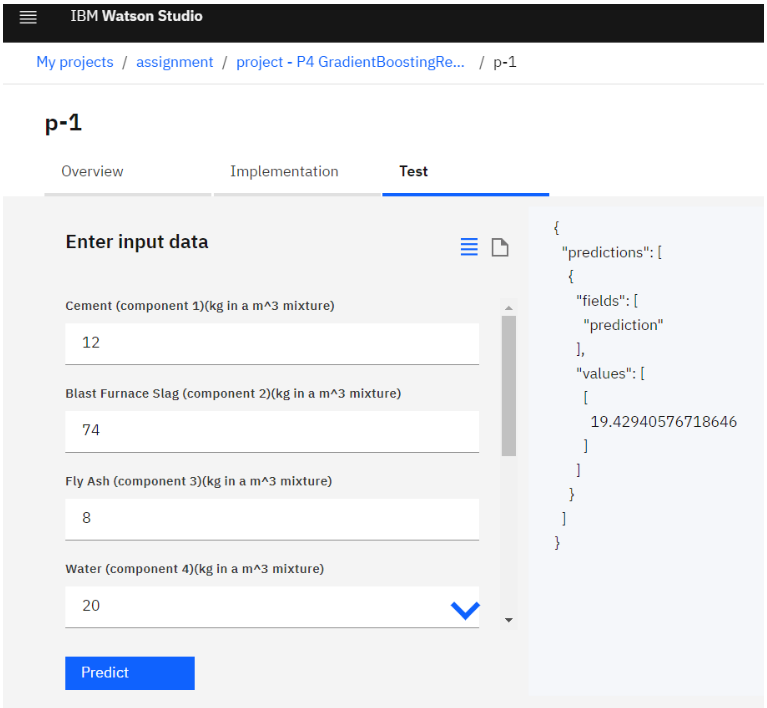


IBM WATSON STUDIO (AUTO AI)

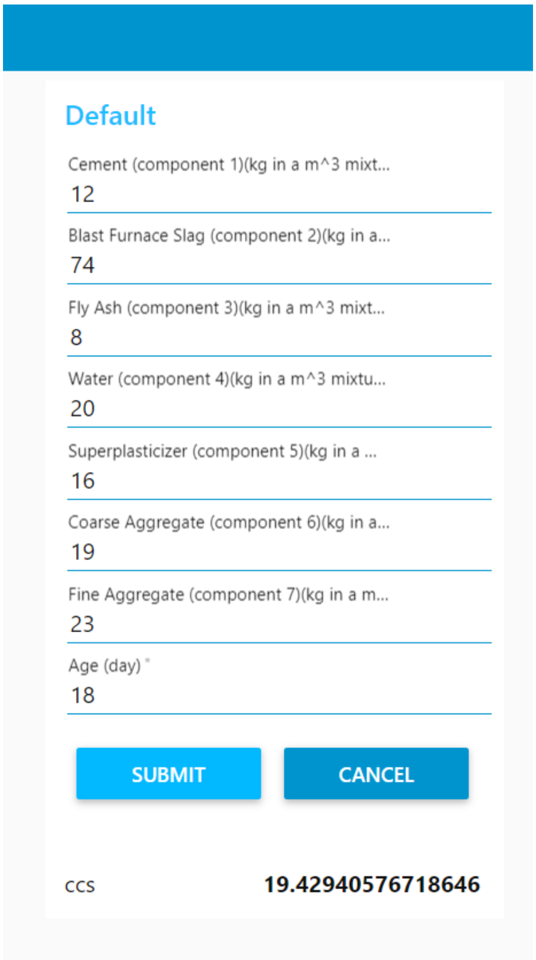


NODE-RED APPLICATION

- **CASE-2**



IBM WATSON STUDIO (AUTO AI)



NODE-RED APPLICATION

7. ADVANTAGES AND DISADVANTAGES

SNO	ADVANTAGES	DISADVANTAGES
1.	Easily identifies trends and patterns	Data Acquisition
2.	No human intervention needed (automation)	Time and Resources
3.	Continuous Improvement	Interpretation of Results
4.	Handling multi-dimensional and multi-variety data	High error-susceptibility
5.	Wide Applications	

8. APPLICATIONS

The applications of Machine Learning are:

- Virtual Personal Assistants
  - Siri, Alexa, Google Now are some of the popular examples of virtual personal assistants. As the name suggests, they assist in finding information, when asked over voice.
- Predictions While Commuting
  - We all have been using GPS navigation services. While we do that, our current locations and velocities are being saved at a central server for managing traffic. This data is then used to build a map of current traffic. While this helps in preventing the traffic and does congestion analysis, the underlying problem is that there are less number of cars that are equipped with GPS.
  - When booking a cab, the app estimates the price of the ride. When sharing these services, how do they minimize the detours? The answer is machine learning.
- Videos Surveillance
  - The video surveillance system nowadays are powered by AI that makes it possible to detect crime before they happen. They track unusual behaviour of people like standing motionless for a long time, stumbling, or napping on benches etc. The system can thus give an alert to human attendants, which can ultimately help to avoid mishaps.
- Social Media Services
  - From personalizing your news feed to better ads targeting, social media platforms are utilizing machine learning for their own and user benefits.
- Email Spam And Malware Filtering
  - There are a number of spam filtering approaches that email clients use. To ascertain that these spam filters are continuously updated, they are powered by machine learning.

9. CONCLUSION

Using the IBM Watson Auto AI Experiment we tested the model and using node red Application in IBM cloud we created a UI where we deployed our model.The predicted value obtained in IBM Watson Studio Auto AI and the predicted value obtained in Node-red application matches each other when we give the same input values are given .Therefore model is successfully deployed can predict the Compressive Strength Of Concrete.

10. FUTURE SCOPE

The scope of Machine Learning in India, as well as in other parts of the world, is high in comparison to other career fields when it comes to job opportunities. According to Gartner, there will be 2.3 million jobs in the field of Artificial Intelligence and Machine Learning by 2022. Also, the salary of a Machine Learning Engineer is much higher than the salaries offered to other job profiles

11. BIBLIOGRAPHY

- https://www.ibm.com/cloud
- https://node-red-dnqjh.eu-gb.mybluemix.net/red/#flow/ec0d381.d1fc1c8
- https://en.wikipedia.org/wiki/Machine_learning
- https://intellipaat.com/blog/future-scope-of-machine-learning/#:~:text=The%20scope%20of%20Machine%20Learning%20in%20India%2C%20as%20well%20as,and%20Machine%20Learning%20by%202022.
- https://data-flair.training/blogs/advantages-and-disadvantages-of-machine-learning/

APPENDIX

SOURCE CODE:


```
[
  {
    "id": "ec0d381.d1fc1c8",
    "type": "tab",
    "label": "Flow 1",
    "disabled": false,
    "info": ""
  },
  {
    "id": "78bfc14b.6f442",
    "type": "ui_base",
    "theme": {
      "name": "theme-light",
      "lightTheme": {
        "default": "#0094CE",
        "baseColor": "#0094CE",
        "baseFont": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif",
        "edited": true,
        "reset": false
      },
      "darkTheme": {
        "default": "#097479",
        "baseColor": "#097479",
        "baseFont": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif",
        "edited": false
      },
      "customTheme": {
        "name": "Untitled Theme 1",
        "default": "#4B7930",
        "baseColor": "#4B7930",
        "baseFont": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"
      },
      "themeState": {
        "base-color": {
          "default": "#0094CE",
```

"value": "#0094CE",
"edited": false
},
"page-titlebar-backgroundColor": {
"value": "#0094CE",
"edited": false
},
"page-backgroundColor": {
"value": "#fafafa",
"edited": false
},
"page-sidebar-backgroundColor": {
"value": "#ffffff",
"edited": false
},
"group-textColor": {
"value": "#1bbfff",
"edited": false
},
"group-borderColor": {
"value": "#ffffff",
"edited": false
},
"group-backgroundColor": {
"value": "#ffffff",
"edited": false
},
"widget-textColor": {
"value": "#111111",
"edited": false
},
"widget-backgroundColor": {
"value": "#0094ce",
"edited": false
},
"widget-borderColor": {
"value": "#ffffff",
"edited": false
},
"base-font": {
"value": "-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"
}
},

```
        "angularTheme": {
            "primary": "indigo",
            "accents": "blue",
            "warn": "red",
            "background": "grey"
        }
    },
    "site": {
        "name": "Node-RED Dashboard",
        "hideToolbar": "false",
        "allowSwipe": "false",
        "lockMenu": "false",
        "allowTempTheme": "true",
        "dateFormat": "DD/MM/YYYY",
        "sizes": {
            "sx": 48,
            "sy": 48,
            "gx": 6,
            "gy": 6,
            "cx": 6,
            "cy": 6,
            "px": 0,
            "py": 0
        }
    }
},
{
    "id": "1667480b.b3af08",
    "type": "ui_group",
    "z": "",
    "name": "Default",
    "tab": "1fdfd828.a10798",
    "order": 1,
    "disp": true,
    "width": "6",
    "collapse": false
},
{
    "id": "1fdfd828.a10798",
    "type": "ui_tab",
    "z": "",
    "name": "Home",
    "icon": "dashboard",
    "disabled": false,
```

"hidden": false
},
{
"id": "5998ad14.618e54",
"type": "ui_group",
"z": "",
"name": "Default",
"tab": "942bc37e.860b",
"order": 1,
"disp": true,
"width": "6",
"collapse": false
},
{
"id": "942bc37e.860b",
"type": "ui_tab",
"z": "",
"name": "Home",
"icon": "dashboard",
"disabled": false,
"hidden": false
},
{
"id": "6f1deacd.dbde84",
"type": "function",
"z": "ec0d381.d1fc1c8",
"name": "PreToken",
"func": "global.set(\"ce\",msg.payload.ce)\nglobal.set(\"bl\",msg.payload.bl)\nglobal.set(\"fl\",msg.payload.fl)\nglobal.set(\"wa\",msg.payload.wa)\nglobal.set(\"su\",msg.payload.su)\nglobal.set(\"co\",msg.payload.co)\nglobal.set(\"fi\",msg.payload.fi)\nglobal.set(\"ag\",msg.payload.ag)\nvar apikey=\"FgG4_sX0GHKoEiVpYheVn0fjj7vcPav8WRuyR5PWo2-S\";\nmsg.headers={\"content-type\":\"application/x-www-form-urlencoded\"}\nmsg.payload={\"grant_type\":\"urn:ibm:params:oauth:grant-type:apikey\",\"apikey\":apikey}\nreturn msg;",
"outputs": 1,
"noerr": 0,
"x": 280,
"y": 400,
"wires": [
[
"5e45de5d.b163b"
]
]
},

{
    "id": "5e45de5d.b163b",
    "type": "http request",
    "z": "ec0d381.d1fc1c8",
    "name": "",
    "method": "POST",
    "ret": "obj",
    "paytoqs": false,
    "url": "https://iam.bluemix.net/identity/token",
    "tls": "",
    "persist": false,
    "proxy": "",
    "authType": "",
    "x": 479,
    "y": 388.9999952316284,
    "wires": [
        [
            "8a8fedc1.233a2"
        ]
    ]
},
{
    "id": "c27237e6.91c558",
    "type": "inject",
    "z": "ec0d381.d1fc1c8",
    "name": "",
    "topic": "",
    "payload": "",
    "payloadType": "date",
    "repeat": "",
    "crontab": "",
    "once": false,
    "onceDelay": 0.1,
    "x": 125.5,
    "y": 313.99999809265137,
    "wires": [
        [
            "6f1deacd.dbde84"
        ]
    ]
},
{
    "id": "a8ab0a69.90b0e8",
    "type": "debug",

"z": "ec0d381.d1fc1c8",
"name": "",
"active": true,
"tosidebar": true,
"console": false,
"tostatus": false,
"complete": "payload",
"targetType": "msg",
"x": 870,
"y": 100,
"wires": []
},
{
"id": "8a8fedc1.233a2",
"type": "function",
"z": "ec0d381.d1fc1c8",
"name": "Pre Prediction",
"func": "var ce = global.get('ce')\nvar bl = global.get('bl')\nvar fl = global.get('fl')\nvar wa = global.get('wa')\nvar su = global.get('su')\nvar co = global.get('co')\nvar fi = global.get('fi')\nvar ag = global.get('ag')\nvar token=msg.payload.access_token\nvar instance_id=\"c9b4fa9d-b68b-4294-b436-3dc9172f6b88\"\nmsg.headers={'Content-Type': 'application/json',\"Authorization\":\"Bearer \"+token,\"ML-Instance-ID\":instance_id}\nmsg.payload={\"input_data\": [{\"fields\": [\"Cement (component 1)(kg in a m^3 mixture)\", \"Blast Furnace Slag (component 2)(kg in a m^3 mixture)\", \"Fly Ash (component 3)(kg in a m^3 mixture)\", \"Water (component 4)(kg in a m^3 mixture)\", \"Superplasticizer (component 5)(kg in a m^3 mixture)\", \"Coarse Aggregate (component 6)(kg in a m^3 mixture)\", \"Fine Aggregate (component 7)(kg in a m^3 mixture)\", \"Age (day)\"],\"values\": [[ce,bl,fl,wa,su,co,fi,ag]]}]}\nreturn msg;",
"outputs": 1,
"noerr": 0,
"x": 676.0000076293945,
"y": 351.9999990463257,
"wires": [
[
"e32d2582.307908"
]
]
},
{
"id": "e32d2582.307908",
"type": "http request",
"z": "ec0d381.d1fc1c8",
"name": "",
"method": "POST",
"ret": "obj",

"paytoqs": false,
"url":
"https://eu-gb.ml.cloud.ibm.com/v4/deployments/61cbc095-847e-4bc0-a23d-289357c1875f/predictions",
"tls": "",
"persist": false,
"proxy": "",
"authType": "",
"x": 769.5000076293945,
"y": 295.9999990463257,
"wires": [
[
"f262cc34.67ef1",
"a8ab0a69.90b0e8"
]
]
},
{
"id": "c7084cc.98803b",
"type": "ui_form",
"z": "ec0d381.d1fc1c8",
"name": "",
"label": "",
"group": "1667480b.b3af08",
"order": 1,
"width": 0,
"height": 0,
"options": [
{
"label": "Cement (component 1)(kg in a m^3 mixture)",
"value": "ce",
"type": "number",
"required": true,
"rows": null
},
{
"label": "Blast Furnace Slag (component 2)(kg in a m^3 mixture)",
"value": "bl",
"type": "number",
"required": true,
"rows": null
},
{
"label": "Fly Ash (component 3)(kg in a m^3 mixture)",
"value": "fl",

"type": "number",
"required": true,
"rows": null
},
{
"label": "Water (component 4)(kg in a m^3 mixture)",
"value": "wa",
"type": "number",
"required": true,
"rows": null
},
{
"label": "Superplasticizer (component 5)(kg in a m^3 mixture)",
"value": "su",
"type": "number",
"required": true,
"rows": null
},
{
"label": "Coarse Aggregate (component 6)(kg in a m^3 mixture)",
"value": "co",
"type": "number",
"required": true,
"rows": null
},
{
"label": "Fine Aggregate (component 7)(kg in a m^3 mixture)",
"value": "fi",
"type": "number",
"required": true,
"rows": null
},
{
"label": "Age (day)",
"value": "ag",
"type": "number",
"required": true,
"rows": null
}
],
"formValue": {
"ce": "",
"bl": "",
"fl": "",

```
        "wa": "",
        "su": "",
        "co": "",
        "fi": "",
        "ag": ""
    },
    "payload": "",
    "submit": "submit",
    "cancel": "cancel",
    "topic": "",
    "x": 71,
    "y": 442.9999952316284,
    "wires": [
        [
            "6f1deacd.dbde84"
        ]
    ]
},
{
    "id": "e43f1416.e146c8",
    "type": "ui_text",
    "z": "ec0d381.d1fc1c8",
    "group": "1667480b.b3af08",
    "order": 2,
    "width": 0,
    "height": 0,
    "name": "",
    "label": "ccs",
    "format": "{{msg.payload}}",
    "layout": "row-spread",
    "x": 797.6000366210938,
    "y": 448.20000743865967,
    "wires": []
},
{
    "id": "f262cc34.67ef1",
    "type": "function",
    "z": "ec0d381.d1fc1c8",
    "name": "",
    "func": "msg.payload=msg.payload.predictions[0].values[0][0]\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 570,
    "y": 100,
```

```
        "wires": [
            [
                "e43f1416.e146c8"
            ]
        ]
    }
]
```