AVALANCHE FORECASTING PREDICTION USING AUTO AI

Internship Title : RSIP Career Basic ML 129
Duration : 1 Month
Project ID : SPS_PRO_297
Project Title : Avalanche Forecasting Prediction Using Auto AI Service
Team : RM SR
Team members : Saarusri R L , Ranjith M
Slot : June 15 - Slot (5)

Submitted by,
Saarusri R L
Ranjith M

INDEX

1.INTRODUCTION

1.1 Overview:

An avalanche, which is also called a snow slide is an event that occurs when a slab of snow lying upon a weaker layer of snow fractures and slides down a steep slope. Avalanches usually accelerate rapidly and grow in mass and volume as they contain more snow. If the avalanche moves fast enough, some of the snow may mix with the air forming a powder snow avalanche, which is a type of gravity current. Slides of rocks or debris, behaving in a similar way to snow, are also referred to as avalanches. Although primarily composed of flowing snow and air, large avalanches have the capability to entrain ice, rocks, trees, and other surficial material. Our project is to predict the occurrence of these avalanches with the help of the given dataset information.

1.2 Purpose:

The main purpose of this project is to predict the occurrence of avalanches with the help of the given parameters like temperature, humidity and intermittency. Provided a particular date and time, we can conclude the risk of avalanches and the probability of it's occurrence. Avalanche kills more than 150 people. If a man stays in avalanche more than 15 minutes it is impossible for him to survive. Avalanches can also kill a lot of animals. A powerful avalanche can destroy even the houses and other buildings. It can completely affect transportation for days. It can also terminate power connection for days and can also have harmful effects in the crops. This project will help the people of the region to be more aware of the situation and helps them to stay safe and be prepared for this situation in prior.

2.LITERATURE SURVEY

2.1 Existing Problem:

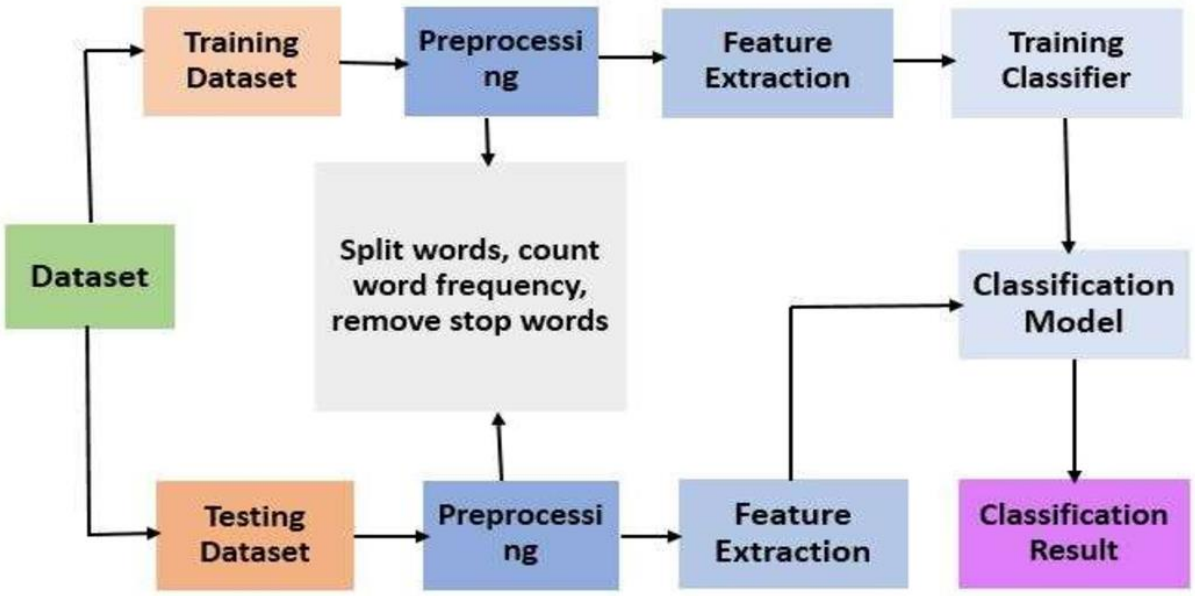
The word Avalanche refers to snow and ice. It means a mass of snow, ice, rocks, slush falling rapidly down a mountain. Snow avalanches are among the most destructive natural hazards threatening human life, ecosystems, built structures, and landscapes in mountainous regions. Each year avalanche kills more than 150 people worldwide. The most common cause of death by avalanche is asphyxiation. If the person buried under an avalanche more than 15 minutes then there is no chance of survive. So, the life of the people in that region is difficult to live. We need to predict if avalanche occurs or not and the strength of the avalanche with the help of the given dataset information.

2.2 Proposed Solution:

This project prevents the people from the avalanche by priory informing them there is a chance to the occurrence of avalanche or not. The model gets the data from the IOT based sensors. After that we want to process those data using a suitable algorithm, then our model display whether the avalanche occur or not and how strength it was. To analyse the data coming from different sensors we are applying various machine learning algorithms. If there is a chance of avalanche then the notification will be sent to people so that they can take decisions accordingly and the model is been built in Auto AI. We apply the dataset in various algorithms and find the one with the highest accuracy. We can predict the strength of the snow with the help of various factors like temperature, humidity and intermittency. The strength of the snow being more than 8 inches indicates high risks of avalanche. The lesser indicates it's still safer.

3.THEORETICAL ANALYSIS

3.1 Block Diagram:



3.2 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 along with the r2 score and 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 avalanche in real time as we enter the details. This app aids in easy usage and better user interface.

4.EXPERIMENTAL INVESTIGATIONS

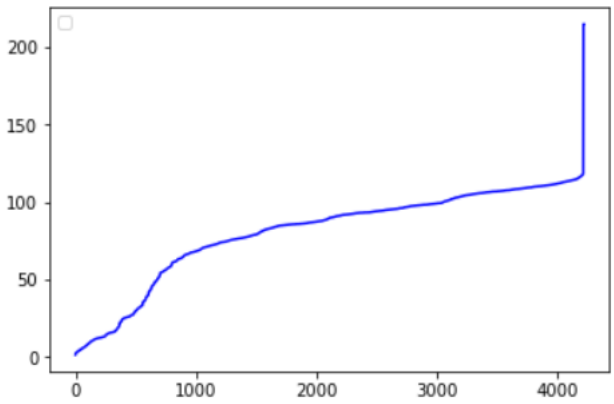
Step1:- Data Collection:

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 depth or strength of the avalanche(snow). The given dataset is not a set of values which are categorical like 0 or 1. Hence we apply the regression algorithms to find the best fit algorithm for the given dataset.

This is the sorted graph of the snow depth value.

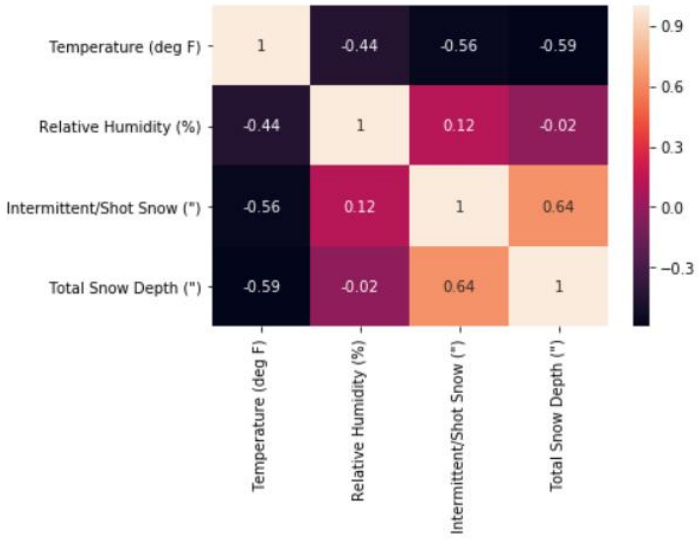


Data pre-processing and analysing:

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

	Temperature (deg F)	Relative Humidity (%)	Intermittent/Shot Snow (")	Total Snow Depth (")
Temperature (deg F)	1.000000	-0.438784	-0.556054	-0.593025
Relative Humidity (%)	-0.438784	1.000000	0.124476	-0.020216
Intermittent/Shot Snow (")	-0.556054	0.124476	1.000000	0.641948
Total Snow Depth (")	-0.593025	-0.020216	0.641948	1.000000

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 inputs.

1.Simple Linear Regression:

We, first apply linear regression in the given dataset and check if it's a suitable one. We take the x value as Intermittency since it is the highly correlated one. We take y as Snow depth. We split the data into training and testing models in the ratio of 80:20 and predicted the r2 score which is considered as the fitting score for the algorithms. We got an r2 score of 45.29 indicating this is not the best method.

```
In [80]: from sklearn.metrics import r2_score
         r2_score(y_test,y_pred2)

Out[80]: 0.4529769823304063
```

2.Decision tree Regressor:

We then applied decision tree regressor to our dataset. We gave the features and labels correspondingly and predicted the r2 score to find how well the model suits the data. We got an r2 score of 63.19. This is good score compared to the previous one but we can still look for better models.

```
In [32]: from sklearn.metrics import r2_score

In [33]: r2_score(y_test,y_pred)
Out[33]: 0.6319033794059443
```

3.Extra trees Regressor:

We applied extra trees regressor to our dataset. We got an r2 score of 75.79.This is a good score compared to previous models.

```
Out[86]: 0.7579089537236828
```

4.Random Forest:

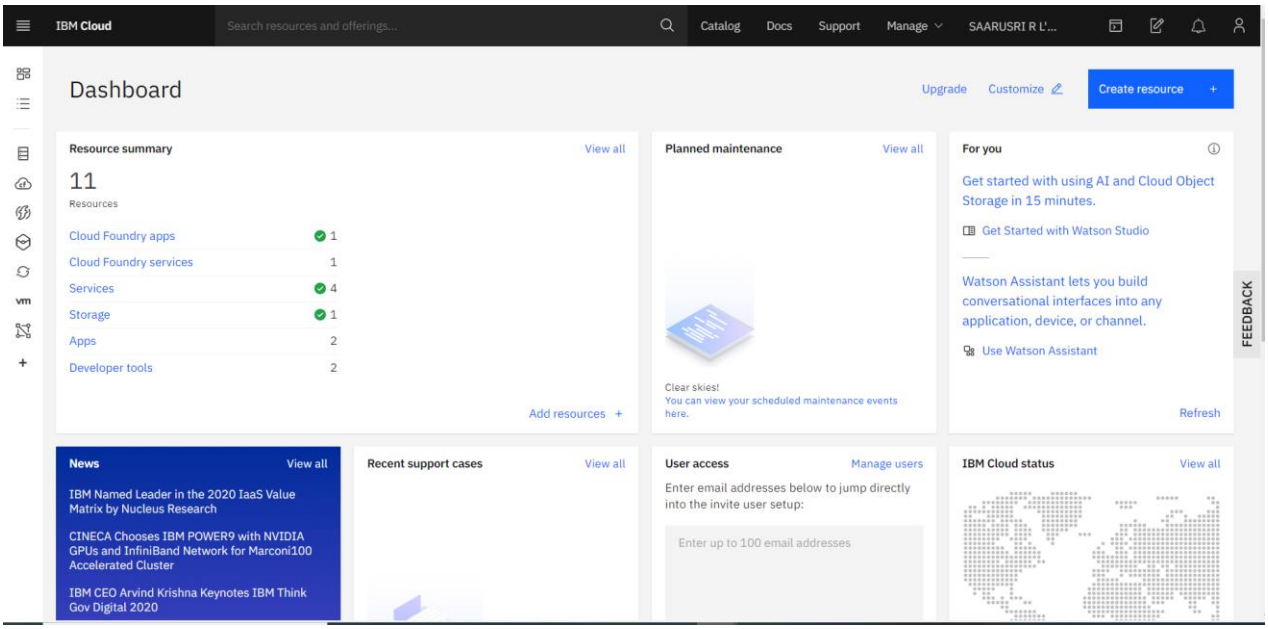
We apply random forest algorithm to our given dataset. We predict the r2 score to be **78.97**. This is the **best r2 score**, in comparison with all the previous algorithms and **hence, we chose to go with this method in our model implementation and deployment.**

```
In [94]: r2_score(y_test,y_pred)
Out[94]: 0.7897108485564904
```

Step2:- IBM Cloud Account:

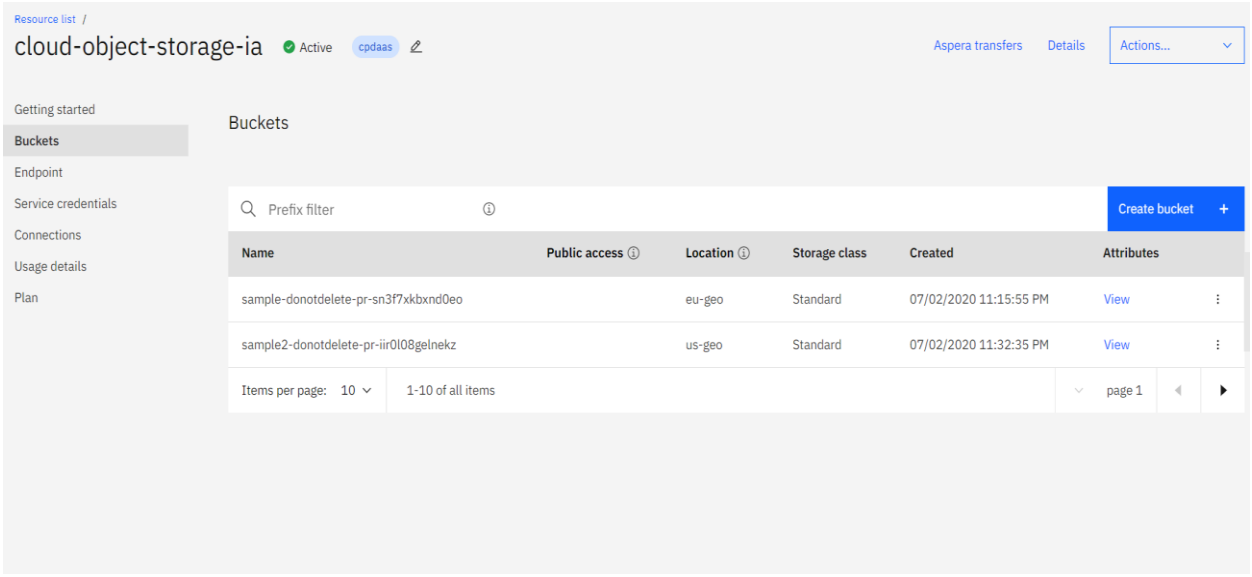
Creating an account:

We have successfully created an IBM Cloud account.



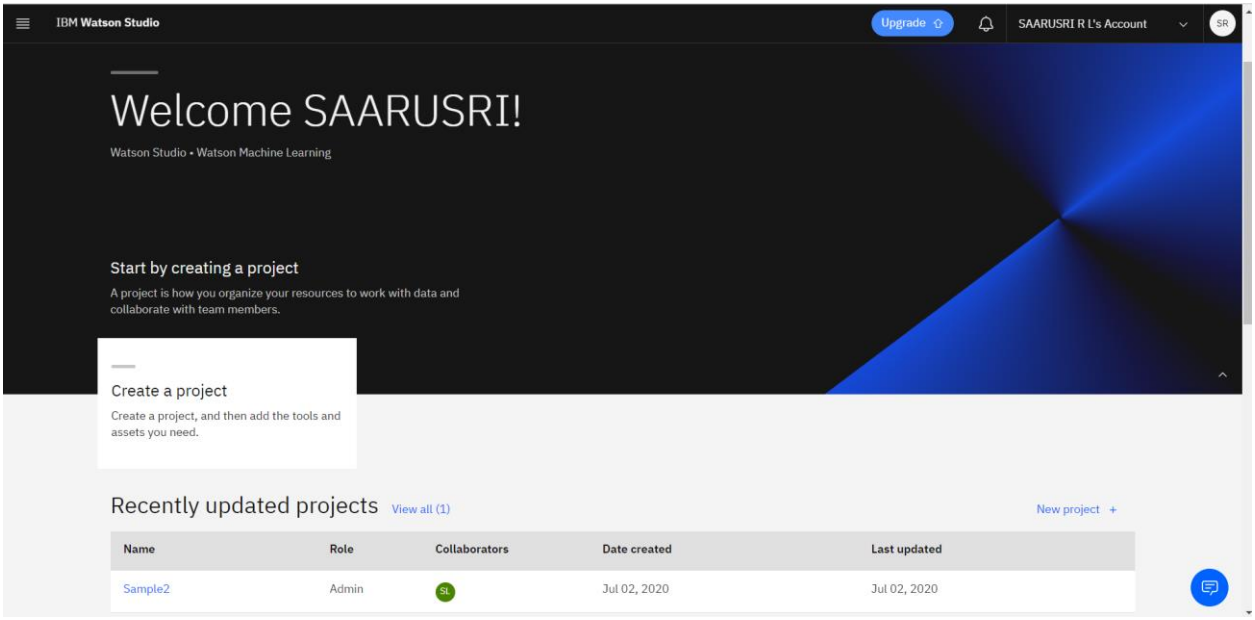
Creating cloud object storage:

We have created a storage and have created a bucket to store our projects.



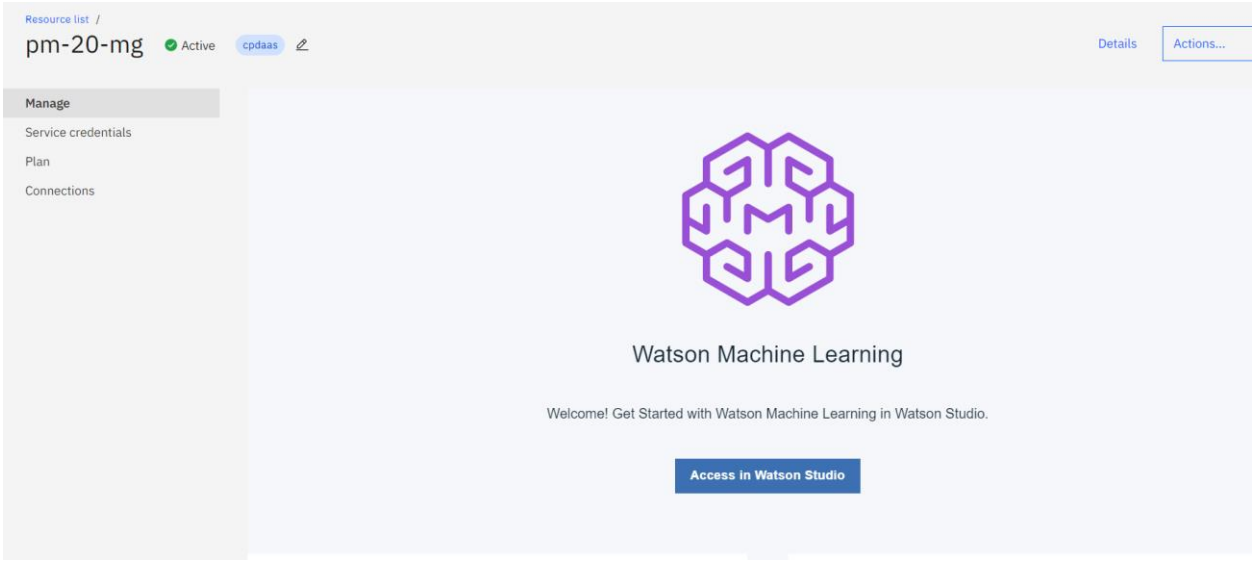
Creating Watson Studio Platform

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:

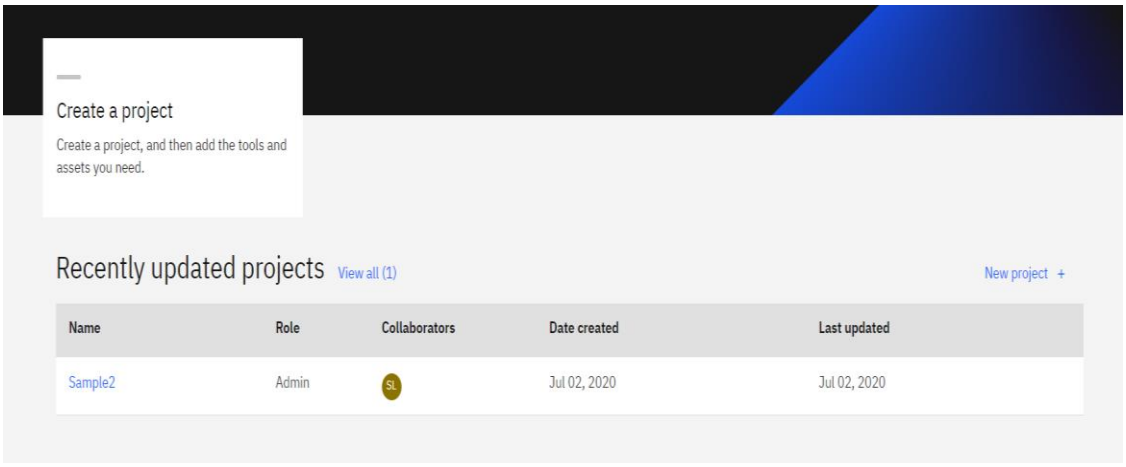
We have created a ML service to load and process our dataset.



Step3:- Model Building

Creating a project in Watson Studio:

We have created a project in Watson studio namely sample2.

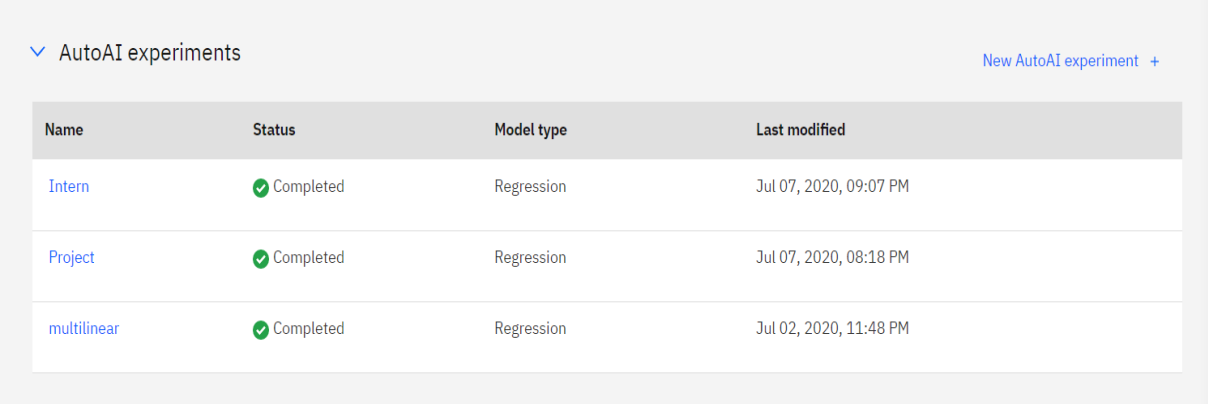


IBM Watson Studio Desktop is a desktop client tool for solving your problems by analysing data with artificial intelligence. With Watson Studio Desktop, you can prepare data and build models on your desktop with visual drag and drop tools. You organize your resources for data analysis tasks in projects. Each project 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:

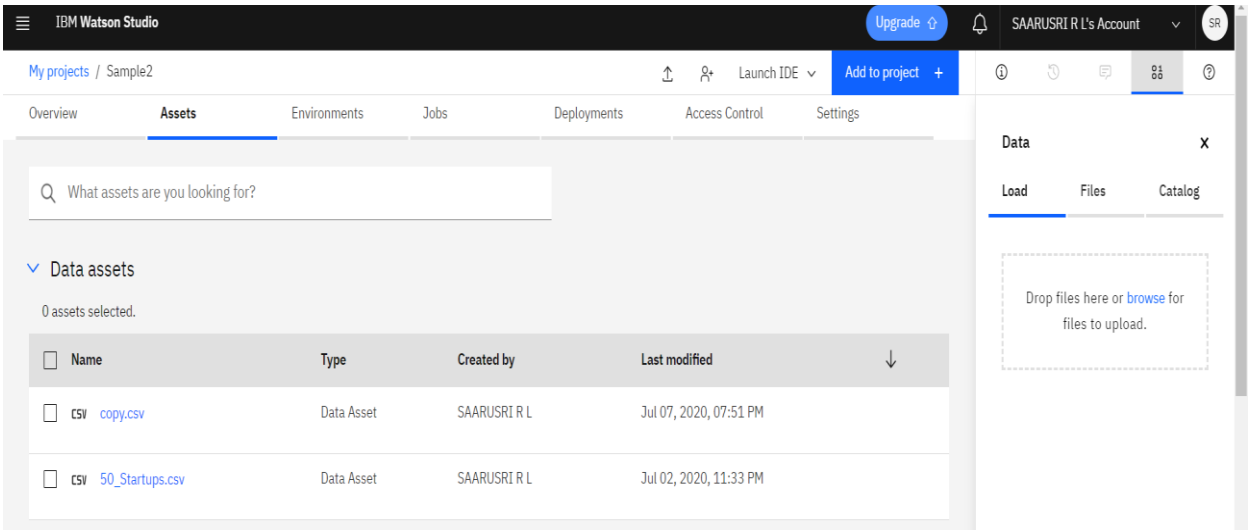
We have created an auto AI experiment called Intern. AutoAI is available within IBM Watson Studio with one-click deployment through Watson Machine Learning. To help simplify an AI lifecycle management, AutoAI automates:

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



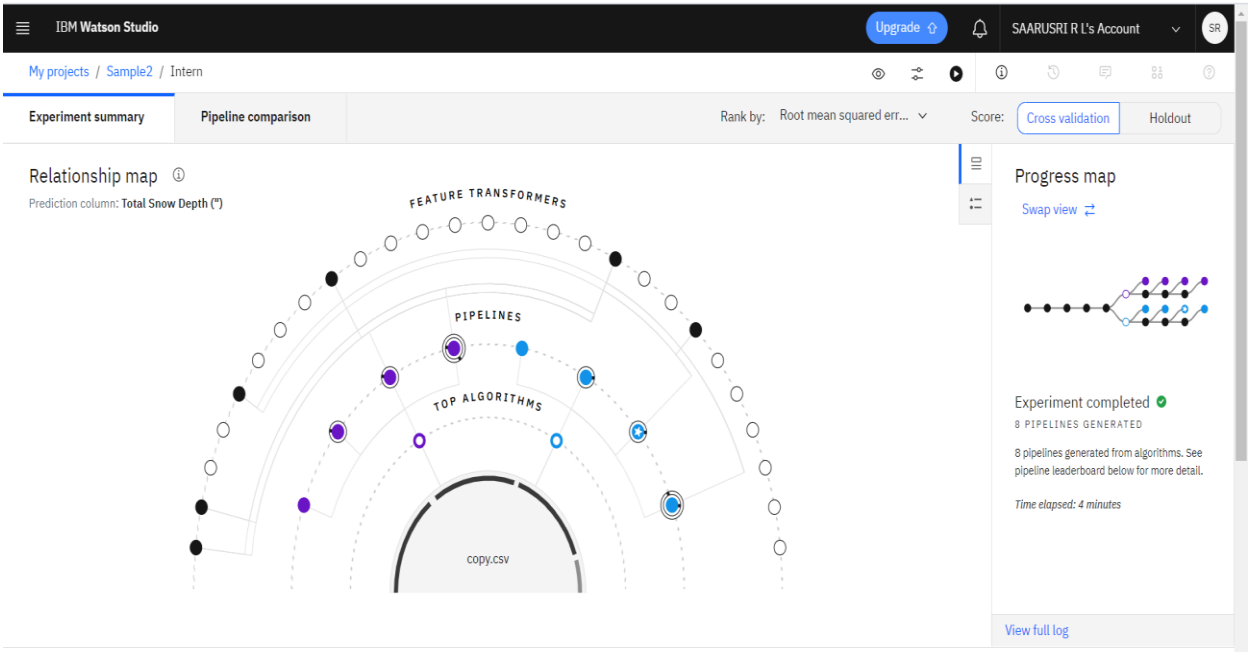
Import Dataset:

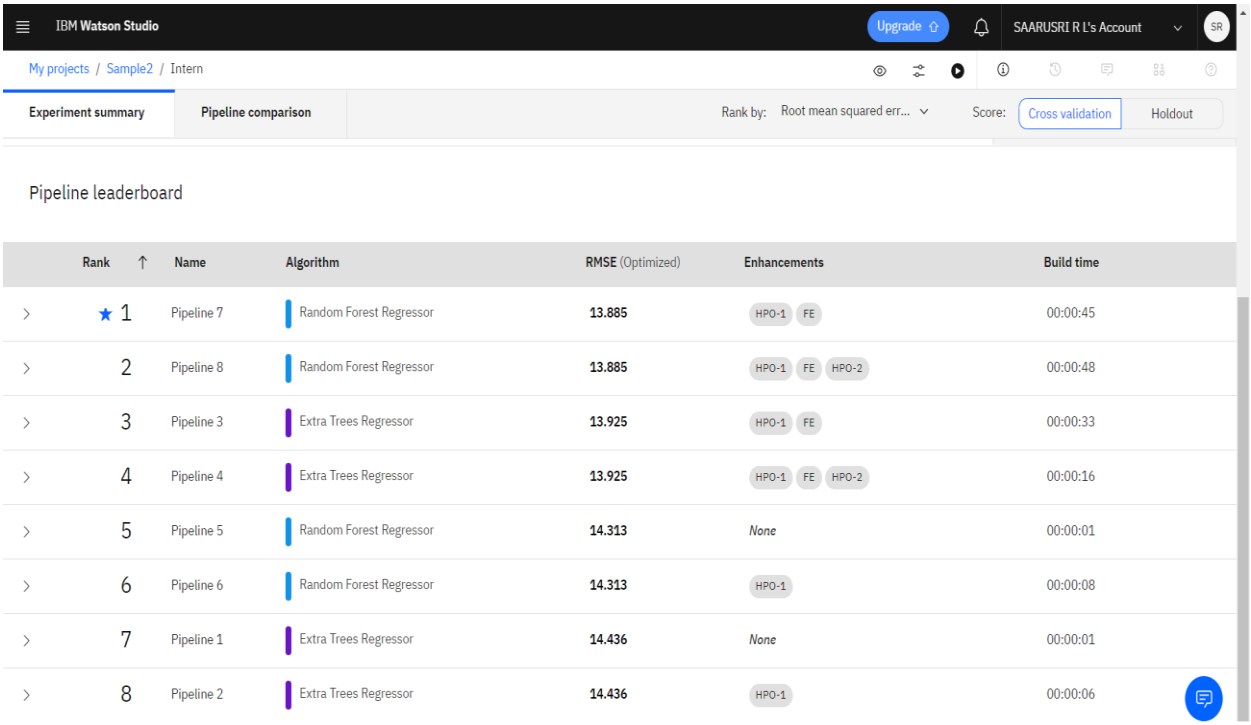
We have then imported the dataset in the name of copy.csv in the assests section.



Run the model and select the pipeline:

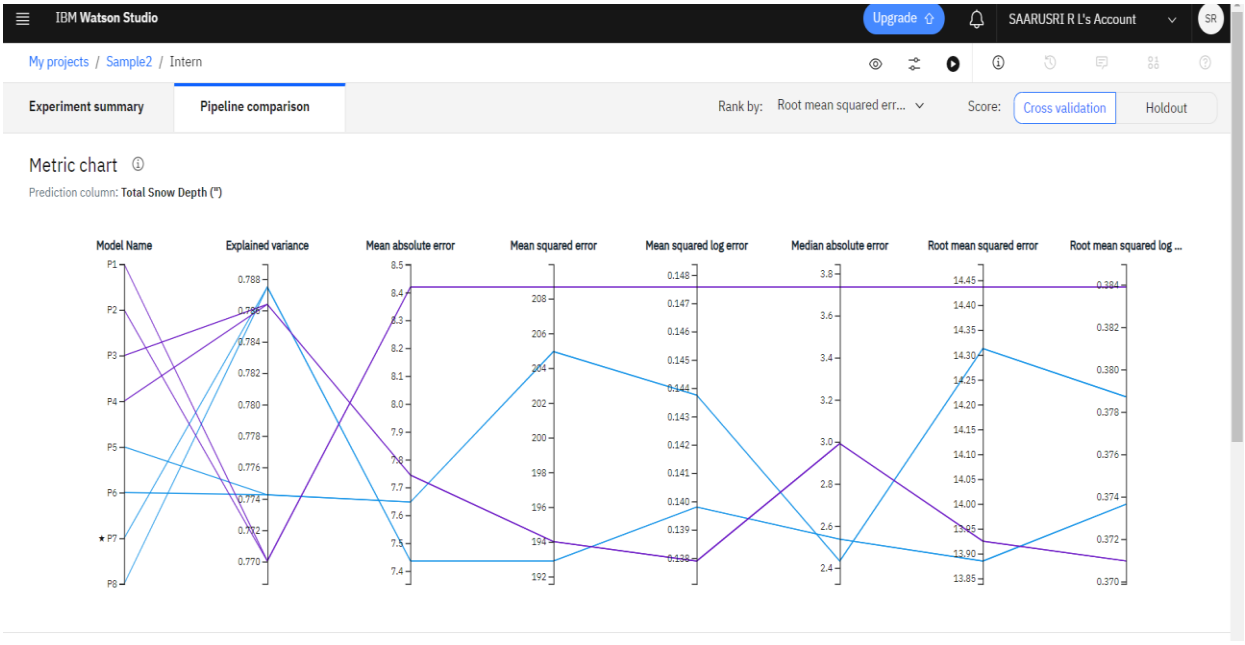
We then run the model, by choosing what to predict, the ratio of train and testing dataset and the number of pipelines to be used. We choose the default 90:10 ratio for training and testing and we chose 8 pipelines.

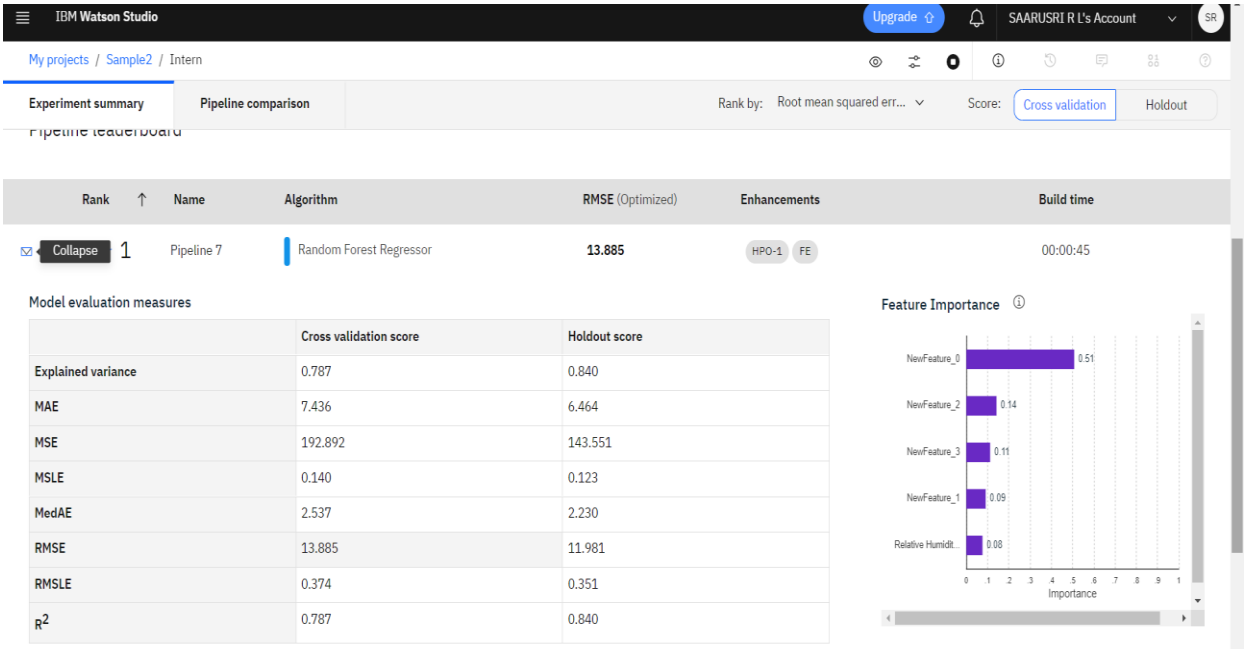




Rank	Name	Algorithm	RMSE (Optimized)	Enhancements	Build time
★ 1	Pipeline 7	Random Forest Regressor	13.885	HPO-1 FE	00:00:45
2	Pipeline 8	Random Forest Regressor	13.885	HPO-1 FE HPO-2	00:00:48
3	Pipeline 3	Extra Trees Regressor	13.925	HPO-1 FE	00:00:33
4	Pipeline 4	Extra Trees Regressor	13.925	HPO-1 FE HPO-2	00:00:16
5	Pipeline 5	Random Forest Regressor	14.313	None	00:00:01
6	Pipeline 6	Random Forest Regressor	14.313	HPO-1	00:00:08
7	Pipeline 1	Extra Trees Regressor	14.436	None	00:00:01
8	Pipeline 2	Extra Trees Regressor	14.436	HPO-1	00:00:06

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.

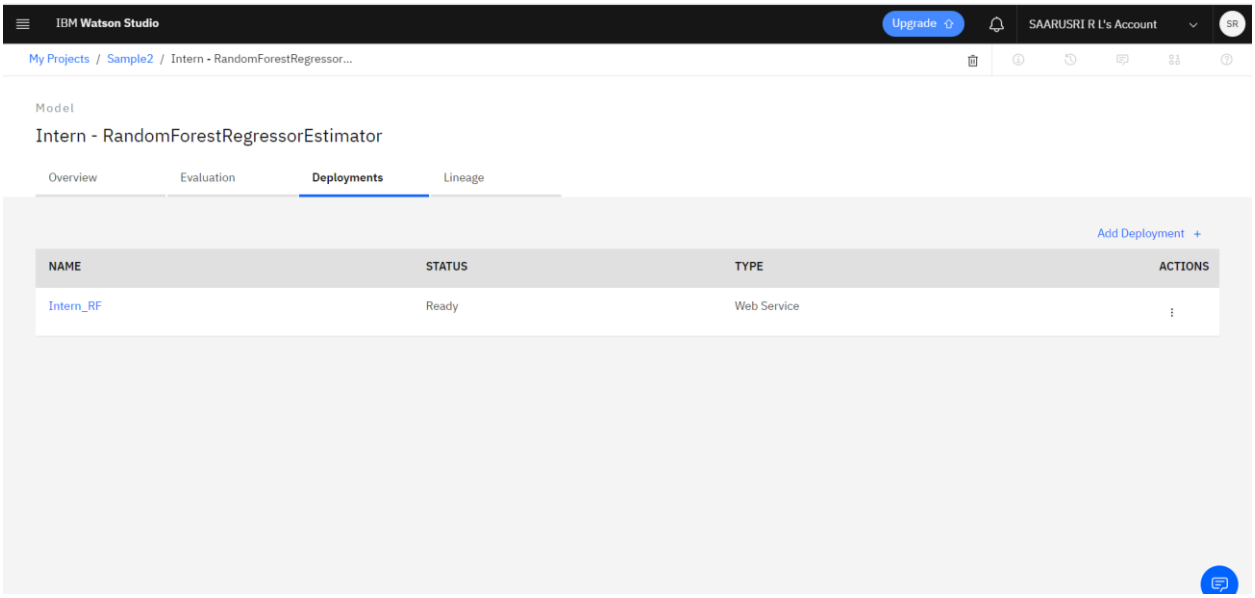




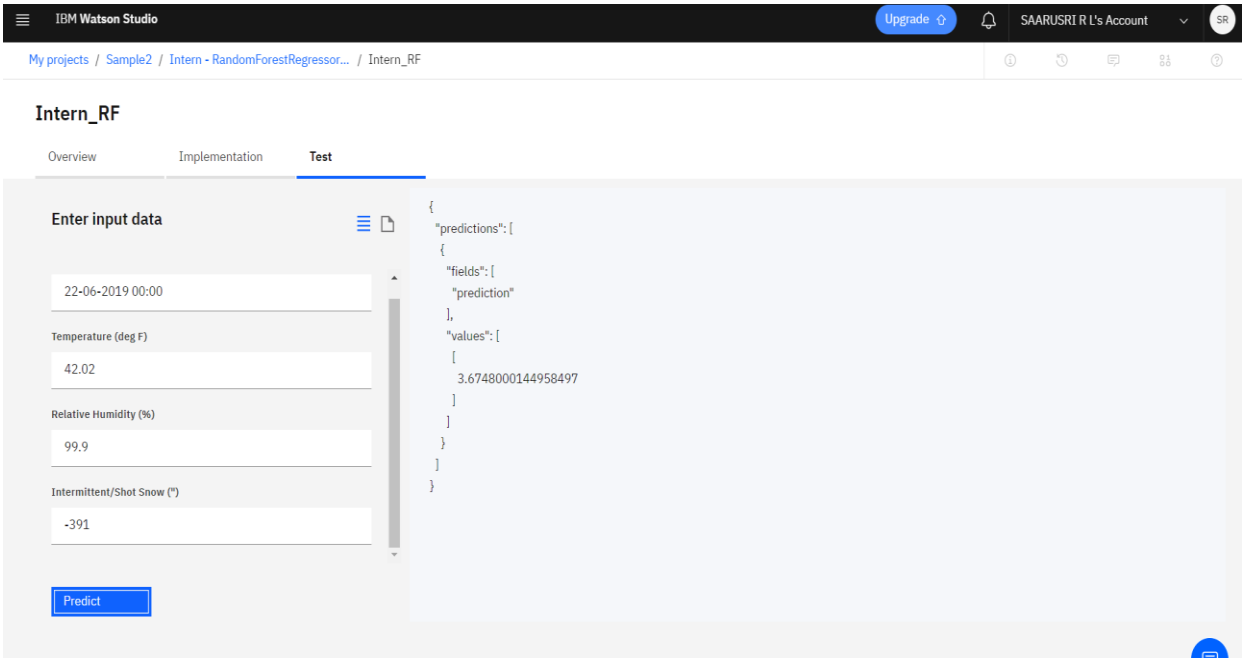
As we can see, the random forest algorithm has a maximum r2 score value of 78.7% in comparison to the other algorithms. We got the r2 value as 78.9 when we applied the same algorithm to our dataset in jupyter notebook. Hence these are verified.

Deploy and test the model in Watson Studio:

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



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:

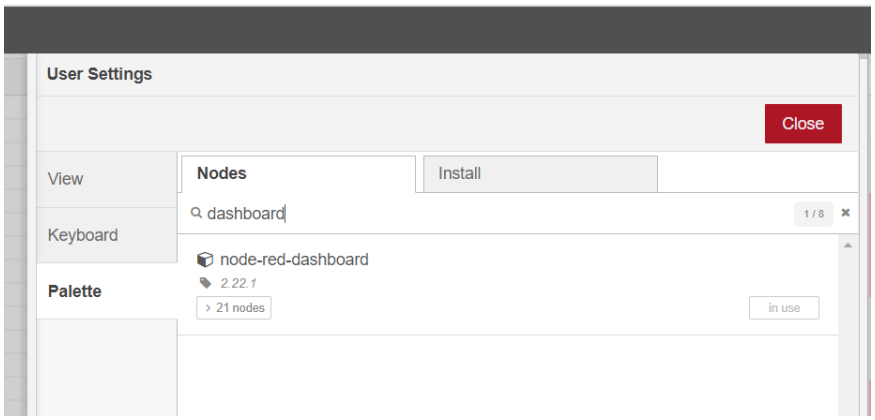
Create a Node Red service:

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.

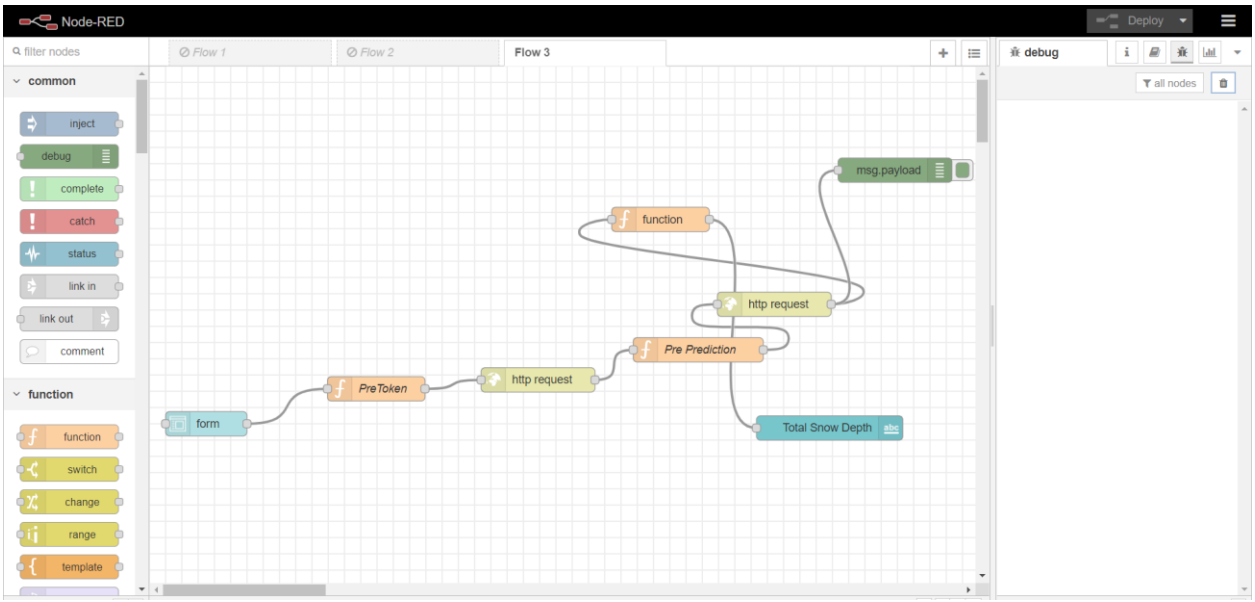
Install dashboard palette:

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

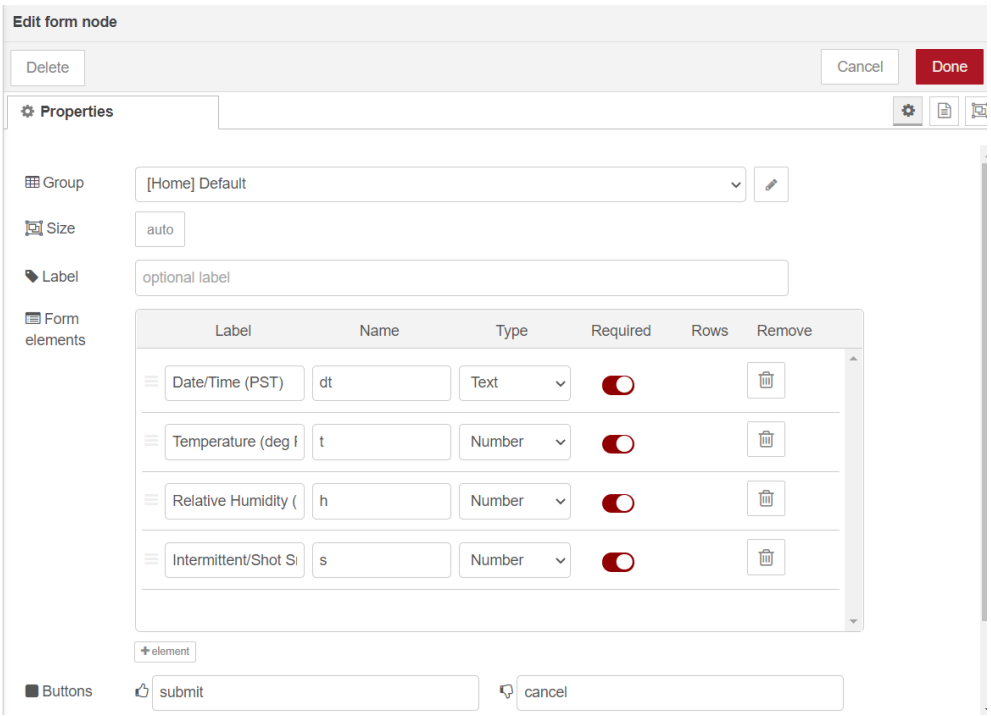


Building UI with Node Red:

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.



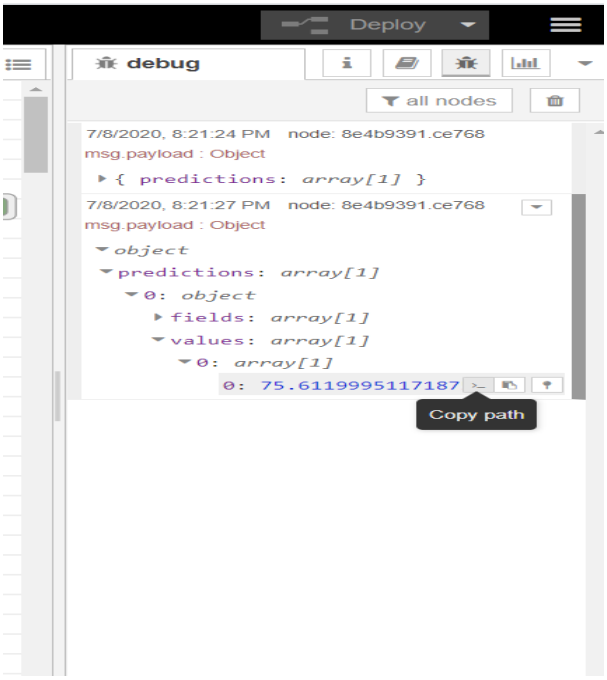
We have the data types as number for temperature, humidity and Intermittency. We have given the data type of date and time to be text.

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.

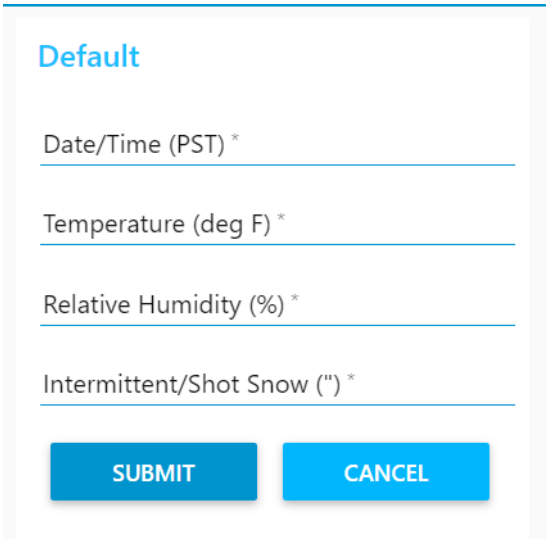


msg.payload : This node displays the value of our prediction (i.e) the depth of the snow in our case. This node displays selected message properties in the debug sidebar tab and optionally the runtime log. By default it displays

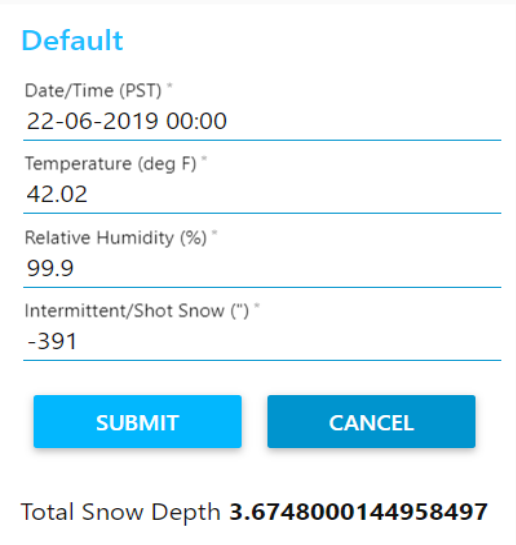
msg.payload, but can be configured to display any property, the full message or the result of a JSON data expression.

Deploy the app and run:

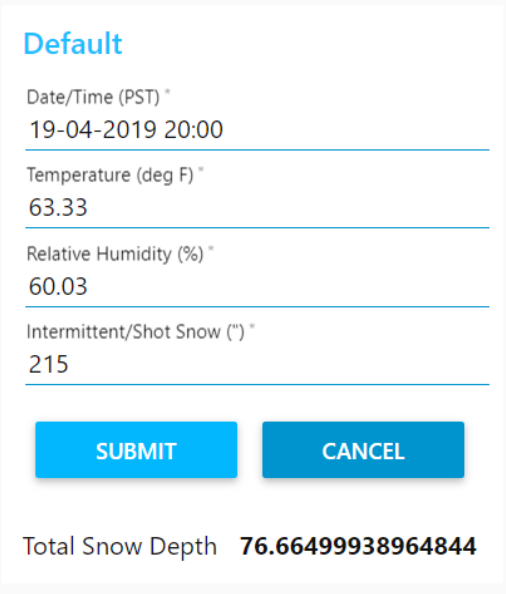
We then deploy our app and we load our ui web page.



This is our web page, we then enter the values to predict the total snow depth.



We have given all the input values and our app has predicted the total snow depth to be 3.67. This indicated less snow depth and hence avalanche is unlikely to occur.

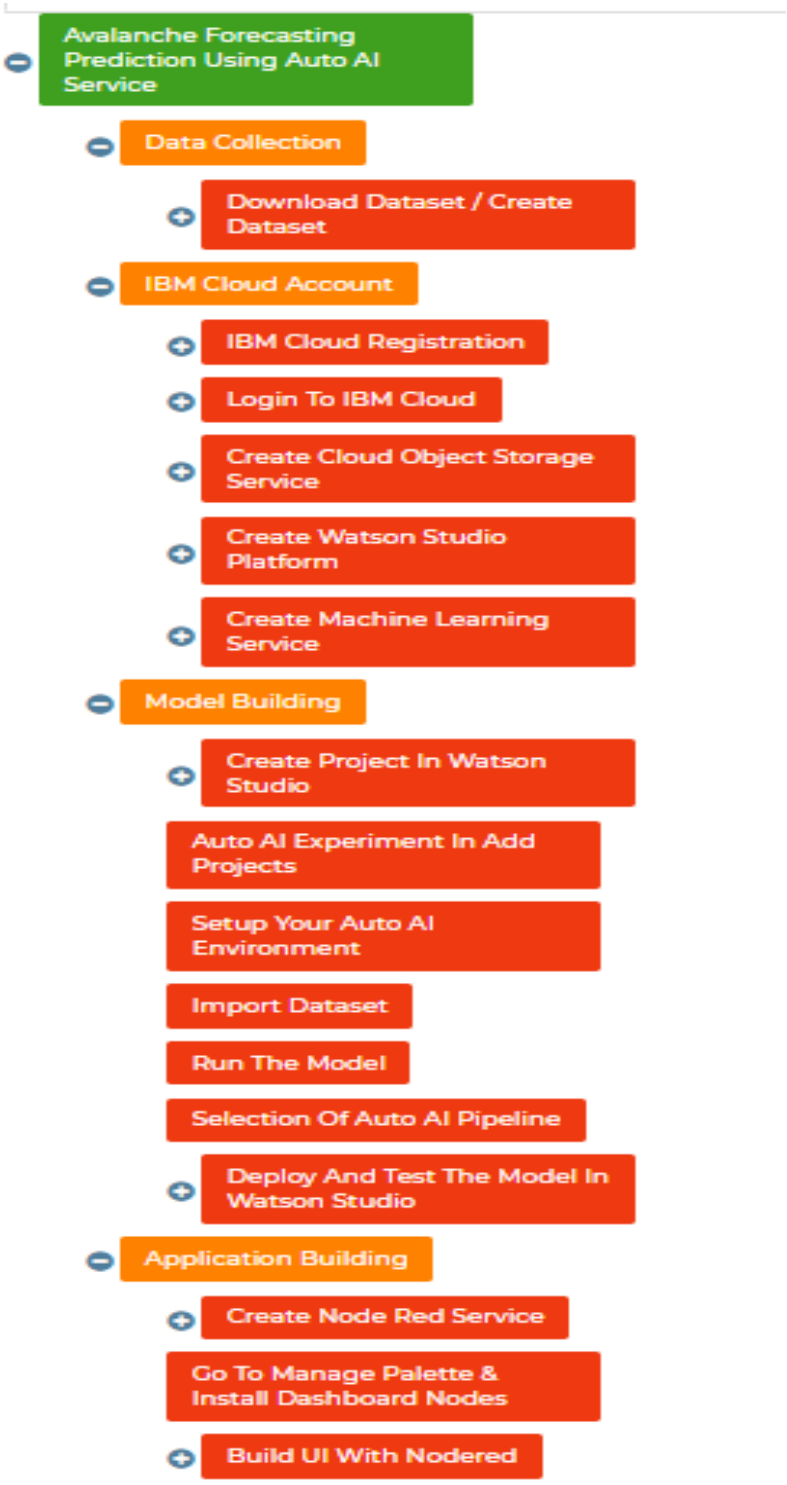


We changed our values and gave higher inputs and we can wee that the snow depth value has raised to be 76.66, indicating avalanche risk.

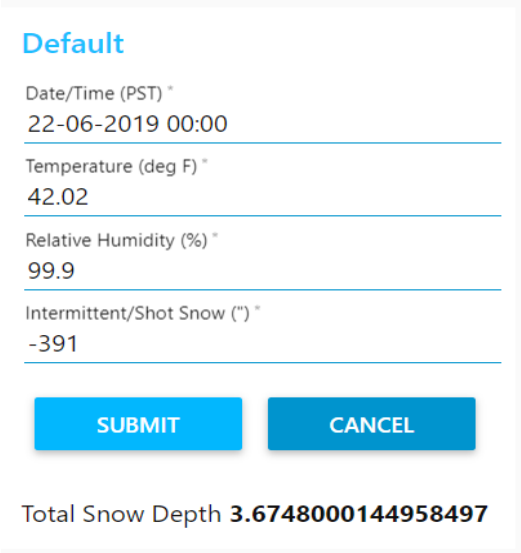
The url for our project is:

https://node-red-iaqyk.eu-gb.mybluemix.net/ui/#!/0?socketid=Sm0fPj0gJrgu2KR3AACJ

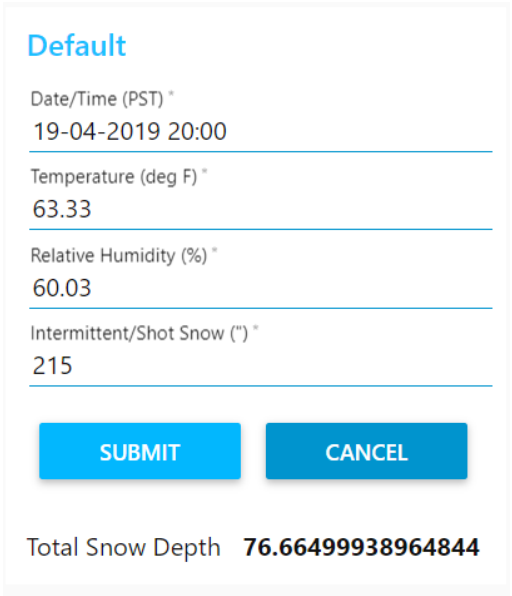
5.FLOW CHART


Avalanche Forecasting Prediction Using Auto AI Service
Data Collection
Download Dataset / Create Dataset
IBM Cloud Account
IBM Cloud Registration
Login To IBM Cloud
Create Cloud Object Storage Service
Create Watson Studio Platform
Create Machine Learning Service
Model Building
Create Project In Watson Studio
Auto AI Experiment In Add Projects
Setup Your Auto AI Environment
Import Dataset
Run The Model
Selection Of Auto AI Pipeline
Deploy And Test The Model In Watson Studio
Application Building
Create Node Red Service
Go To Manage Palette & Install Dashboard Nodes
Build UI With Nodered

6.RESULT



As we can see, our app gives the output depth of the snow if we enter the input values such as temperature, humidity and intermittency in a given date and time. If the snow depth is more than 8 inches there is a high risk of avalanche. If the snow depth is below that, avalanche is unlikely to occur.



As we change the values, the depth of the snow increases, indicating an alarming situation for avalanche occurrence. Thus, our app predicts the avalanche occurrence with great accuracy.

7. ADVANTAGES AND DISADVANTAGES

7.1 Advantages:

- The avalanche prediction helps the people to stay safe and aware of the situation.
- Our web app predicts the occurrence of avalanche 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 avalanche occurrence of their region with the help of the input parameters.
- The app gives results immediately without any delay.

7.2 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 avalanche.

8.APPLICATIONS

- This app can help the locals to predict the avalanche occurrence and be prepared for it.
- This might save the life of the surrounding people and also their animals.
- This app will also help the tourists who go for skiing to decide if to go or not depending upon the weather conditions.
- This app will create an awareness among the people regarding the depth of the avalanche and helps them act accordingly.
- It also helps the travellers to avoid taking the particular routes in case of high-risk factor.

9.CONCLUSION

Avalanche prediction helps the locals, tourists and travellers to stay safe and be prepared for the uncertain situations. More than 150 people die due to avalanches. This app might prevent avalanche deaths to a greater extent. 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. The r2 scores also match with the value we got in our jupyter notebook. The r2 score is 78.7% indicating a good accuracy of our model. This is the highest accuracy score in comparison to the other models. We can also see the comparison of various pipelines used in the IBM cloud. We can also see their error and r2 scores and find our why this algorithm is best suited for this approach. 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 weather such as temperature, humidity and the intermittency to predict the depth the snow and that helps us identify the risk of avalanche. This app helps us to predict the avalanche risk from any region provided we know the input parameters to enter. Thus, we have predicted the avalanche occurrence and it's strength with a great accuracy and also have deployed it as an app providing a better user interface.

10.FUTURE SCOPE

Our app can predict the avalanche occurrence and it's 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 avalanche occurrence has to be predicted. We can also link our app to the geographical locations of the user. This is will enable the app to get the weather details of the particular day and time directly without the user having to type them manually. This will simplify the process to a grater extent and will provide a much better user interface. We can also make the app automated to analyse the weather 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 mobile daily weather reports making it much easier for the users to be aware and safe.

11.BIBILOGRAPHY

https://gibbonswhistler.com/what-is-an-avalanche/
https://www.getprepared.gc.ca/cnt/hzd/vlchs-en.aspx
https://www.guidedolomiti.com/en/the-avalanches/