Team

Project Report on

IMPACT OF COVID-19 ON FOOD SECURITY-VISUALIZATION DASHBOARD

Submitted

By

Sam Clastine Jesumuthu

Hariharan P

Chapter No **INDEX**

1. PROBLEM DESCRIPTION

The COVID-19 pandemic is a health and human crisis threatening the food security and nutrition of millions of people around the world. Hundreds of millions of people were already suffering from hunger and malnutrition before the virus hit and, unless immediate action is taken, we could see a global food emergency. In the longer term, the combined effects of COVID-19 itself, as well as corresponding mitigation measures and the emerging global recession could, without large-scale coordinated action, disrupt the functioning of food systems. Such disruption can result in consequences for health and nutrition of a severity and scale unseen for more than half a century.

Alarmed by a potential rise in food insecurity during the COVID-19 pandemic, many countries and organizations are mounting special efforts to keep agriculture safely running as an essential business, markets well supplied in affordable and nutritious food, and consumers still able to access and purchase food despite movement restrictions and income losses.

2. SOLUTION

Create a machine learning model to predict supply –food security level of people and provide a dashboard to show lockdown situations and Supply Demand can impact the food security of people.

3. THEORETICAL FRAMEWORK DETAILS

3.1 Project Requirement

1. **Identifying The Variables**
2. we had identified the following variables Variable that need to be predicted is **Food Security Level in percent.**
3. Feature Variables are as follows
4. malnutrition
5. disability
6. no_insurance
7. access_to_food
8. severely_housing_cost_burdened
9. median_income
10. below_poverty
11. unemployment
12. children
13. seniors
14. migrants
15. rural_population
16. Supply&Demand
17. lockdown_days

3.2 Data acquisition

1. Collecting the data source

    We had collected the data from various sources are listed below

    1.https://knoema.com/atlas/India/topics/Food-Security
    2.https://data.nal.usda.gov/dataset/international-food-security-0
    3.https://www.ifpri.org/blog/addressing-covid-19-impacts-agriculture-

food-security-and-livelihoods-india
4.https://www.ers.usda.gov/data-products/food-environment-atlas/data-access-and-documentation-downloads/
5. https://github.com/IBM/visualize-food-insecurity

3.3 Software Requirement

We had used the following tools mentioned below

1. Jupyter Notebook
2. IBM Watson Studio
3. Machine Learning Studio
4. Dashboard
5. Auto AI
6. Notebook

3.4 Data Preprocessing

Preparation

Preparation is the manipulation of data into a form suitable for further analysis and processing. Raw data cannot be processed and must be checked for accuracy. Preparation is about constructing a data set from one or more data sources to be used for further exploration and processing. Analyzing data that has not been carefully screened for problems can produce highly misleading results that are heavily dependent on the quality of data prepared.

Input

Inputis the task where verified data is coded or converted into machine readable form so that it can be processed through an application. Data entry is done through the use of a keyboard, scanner, or data entry from an existing source. This time-consuming process requires speed and accuracy. Most data need to follow a formal and strict syntax since a great deal of

processing power is required to breakdown the complex data at this stage. Due to the costs, many businesses are resorting to outsource this stage.

Processing

Processing is when the data is subjected to various means and methods of powerful technical manipulations using Machine Learning and Artificial Intelligence algorithms to generate an output or interpretation about the data. The process may be made up of multiple threads of execution that simultaneously execute instructions, depending on the type of data.

3.5 Creating Machine learning Model

Usually, machine learning models require a lot of data in order for them to perform well. Usually, when training a machine learning model, one needs to collect a large, representative sample of data from a training set. Data from the training set can be as varied as a corpus of text, a collection of images, and data collected from individual users of a service. Overfitting is something to watch out for when training a machine learning model.

Creating the machine learning model using Watson studio Auto AI tool

1. IBM Watson Studio is a collaborative environment with AI tools that you and your team can use to collect and prepare training data, and to design, train, and deploy machine learning models.
2. Ranging from graphical tools you can use to build a model in minutes, to tools that automate running thousands of experiment training runs and hyperparameter optimization, Watson Studio AI tools support popular frameworks, including: TensorFlow, Caffe, PyTorch, and Keras.

3.6 Watson Machine Learning

AutoAI automatically preprocesses your data, selects the best estimator for the data, and then generates model candidate pipelines for you to review and compare. Deploy the best performing pipeline as a machine learning model.

1. Using IBM Watson Machine Learning, you can build analytical models and neural networks, trained with your own data, that you can deploy for use in applications.

2. Watson Machine Learning provides a full range of tools and services so you can build, train, and deploy Machine Learning models. Choose from tools that fully automate the training process for rapid prototyping to tools that give you complete control to create a model that matches your needs.

As you can see here I have passed the following parameters in '**train_test_split':**

1. x and y that we had previously defined
2. test_size: This is set 0.2 thus defining the test size will be 20% of the dataset
3. random_state: it controls the shuffling applied to the data before applying the split. Setting random_state a fixed value will guarantee that the same sequence of random numbers are generated each time you run the code.

When splitting a dataset there are two competing concerns:

1. If you have less training data, your parameter estimates have greater variance.
2. And if you have less testing data, your performance statistic will have greater variance.
3. The data should be divided in such a way that neither of them is too high, which is more dependent on the amount of data you have. If your data is too small then no split will give you satisfactory variance so you will have to do cross-validation but if your data is huge then it doesn't really matter whether you choose an 80:20 split or a 90:10 split (indeed you may choose to use less training data as otherwise, it might be more computationally intensive).

3.7 Splitting The Dataset, Hyperparameters, Cross-Validation, Fit And Tune Models

1. If you evaluate your model on the same data you used to train it, your model could be very overfit and you wouldn't even know! A model should be judged on its ability to predict new, unseen data.
2. Hyperparameters express "higher-level" structural settings for algorithms.

   e.g: strength of the penalty used in regularized regression

1. Cross-validation is a method for getting a reliable estimate of model performance using only your training data.

2. At the end of this process, you will have a cross-validated score for each set of hyperparameter values... for each algorithm.
3. At the end of this process, you will have a cross-validated R2score for each set of hyperparameter values for each algorithm.

3.8 Creating visualization dashboard

Data Visualization and better define the scope and meaning of interactiveness as intended in this article.

Finally, I will showcase some concrete examples for all I have blabbered about, mostly referring to personal projects of mine, and improvements I obtained when relying on these interactive bits. This final part is exactly to demonstrate the capabilities of such tools on an already more than impressive framework like Jupyter. It is all about pushing you to try out for yourself on your projects and spread the word.

We had Created a Visualization Dashboard using Cognos Dashboard in Watson Studio.

3.9 Deployment

Deploy using NodeRed:

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways. It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click.

Node-RED provides a browser-based flow editor that makes it easy 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. Java Script 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.

We had used the following Node are as follows

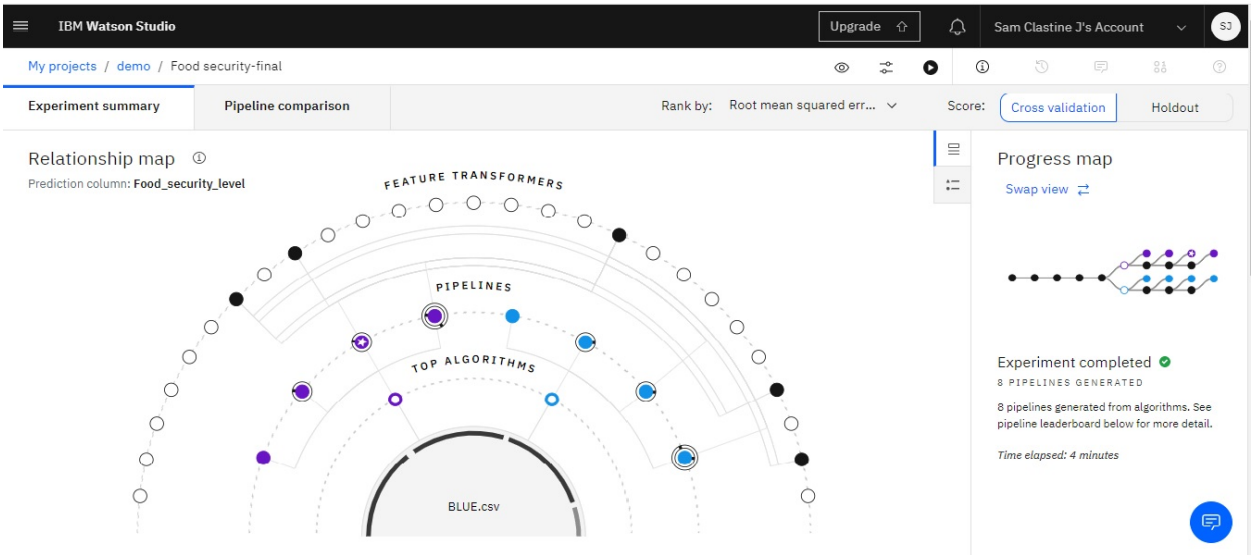
1. Inject

2. Debug
3. Function
4. Chart
5. Timestamp
6. Gauge

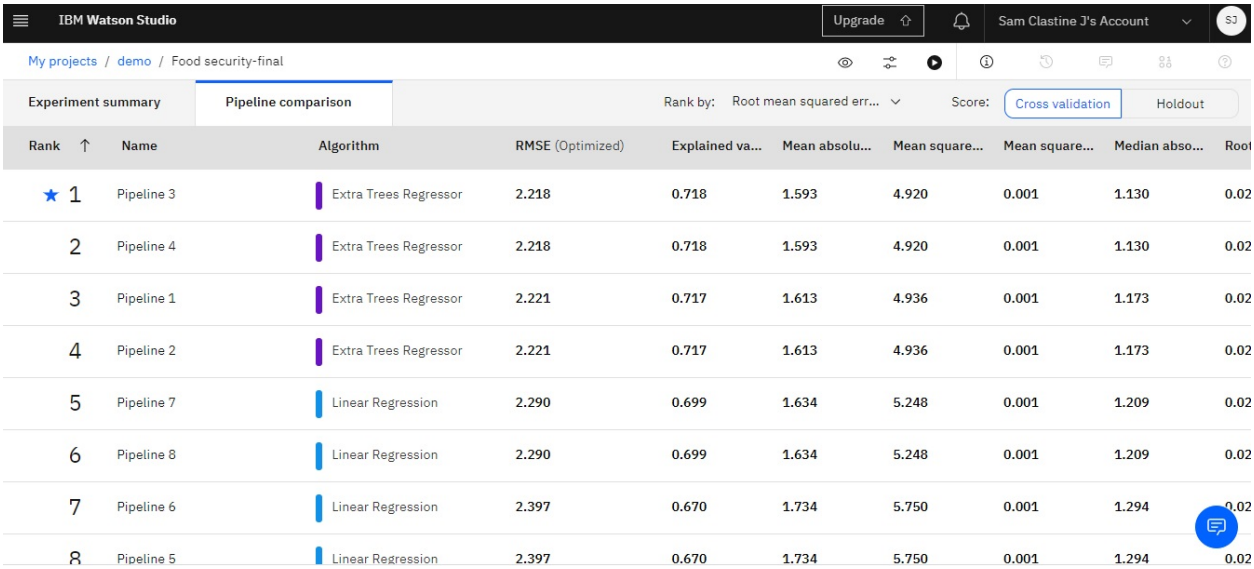
After Deploying the ML Model, we had built a Webpage and redirected the Node Red UI.

4. Project steps (Screenshots)

Creating ML model in Watson Studio Auto AI

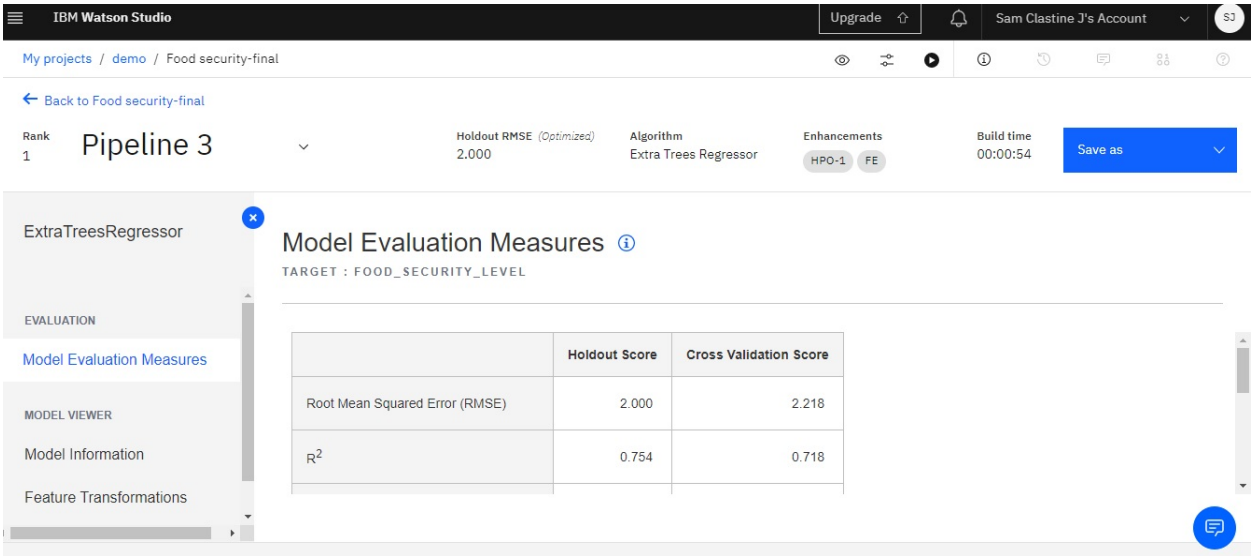


ML model Pipeline Comparison



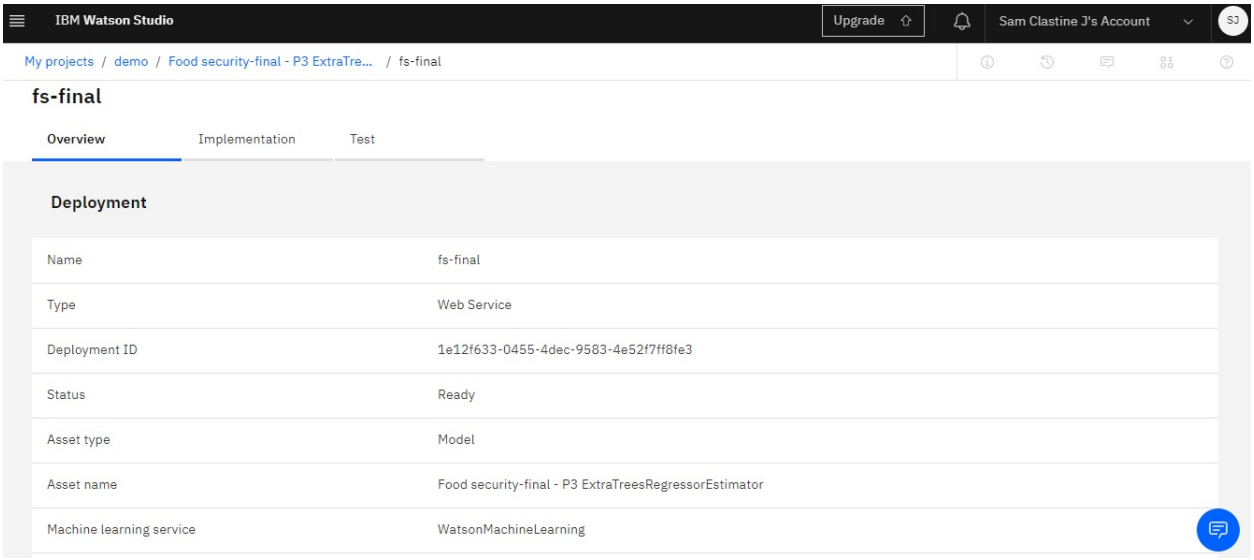
Rank	Name	Algorithm	RMSE (Optimized)	Explained va...	Mean absolu...	Mean square...	Mean square...	Median abso...	Root
★ 1	Pipeline 3	Extra Trees Regressor	2.218	0.718	1.593	4.920	0.001	1.130	0.02
2	Pipeline 4	Extra Trees Regressor	2.218	0.718	1.593	4.920	0.001	1.130	0.02
3	Pipeline 1	Extra Trees Regressor	2.221	0.717	1.613	4.936	0.001	1.173	0.02
4	Pipeline 2	Extra Trees Regressor	2.221	0.717	1.613	4.936	0.001	1.173	0.02
5	Pipeline 7	Linear Regression	2.290	0.699	1.634	5.248	0.001	1.209	0.02
6	Pipeline 8	Linear Regression	2.290	0.699	1.634	5.248	0.001	1.209	0.02
7	Pipeline 6	Linear Regression	2.397	0.670	1.734	5.750	0.001	1.294	0.02
8	Pipeline 5	Linear Regression	2.397	0.670	1.734	5.750	0.001	1.294	0.02

Selecting the Best Pipeline for the model

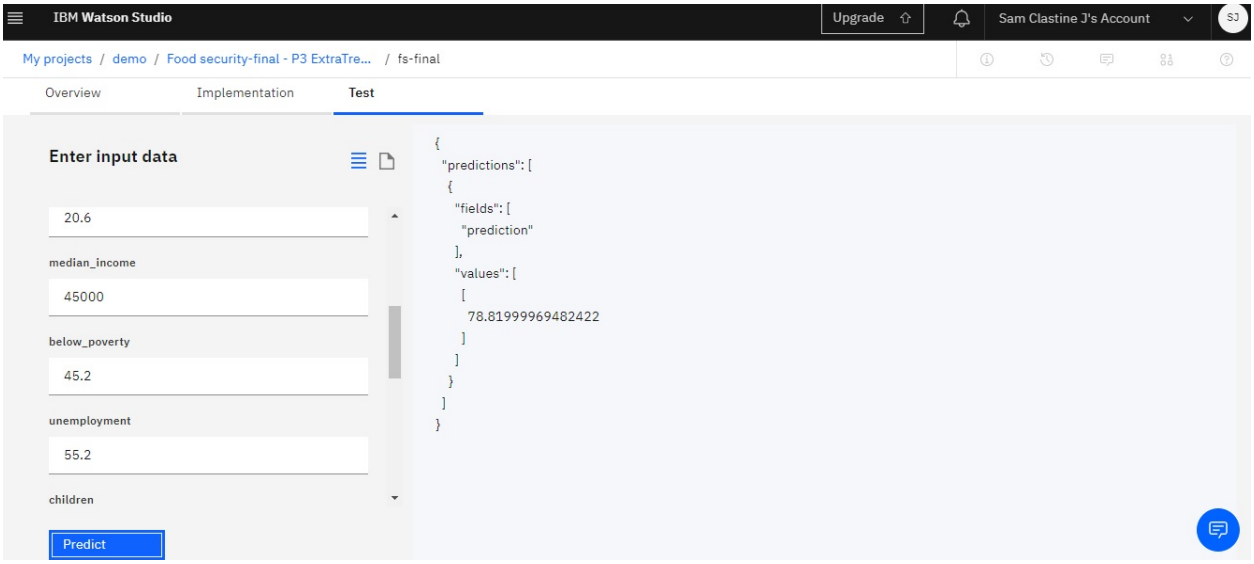


	Holdout Score	Cross Validation Score
Root Mean Squared Error (RMSE)	2.000	2.218
R^2	0.754	0.718

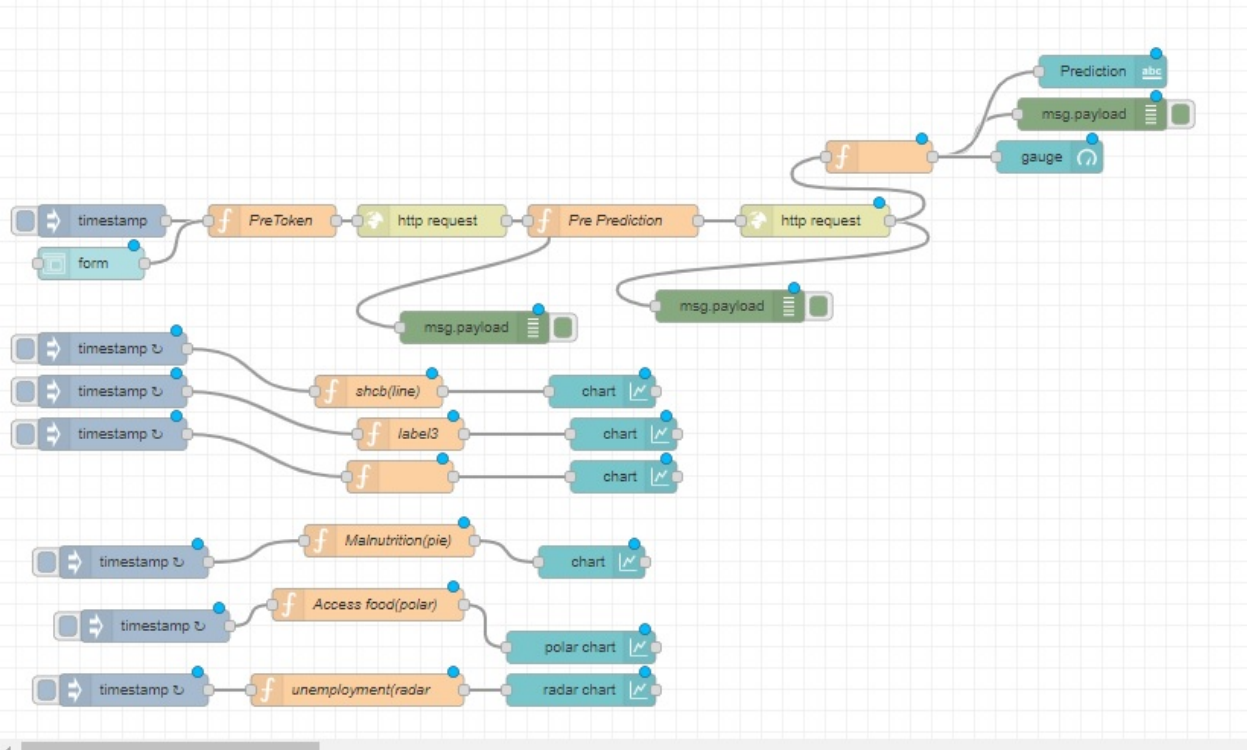
Deploying the Model



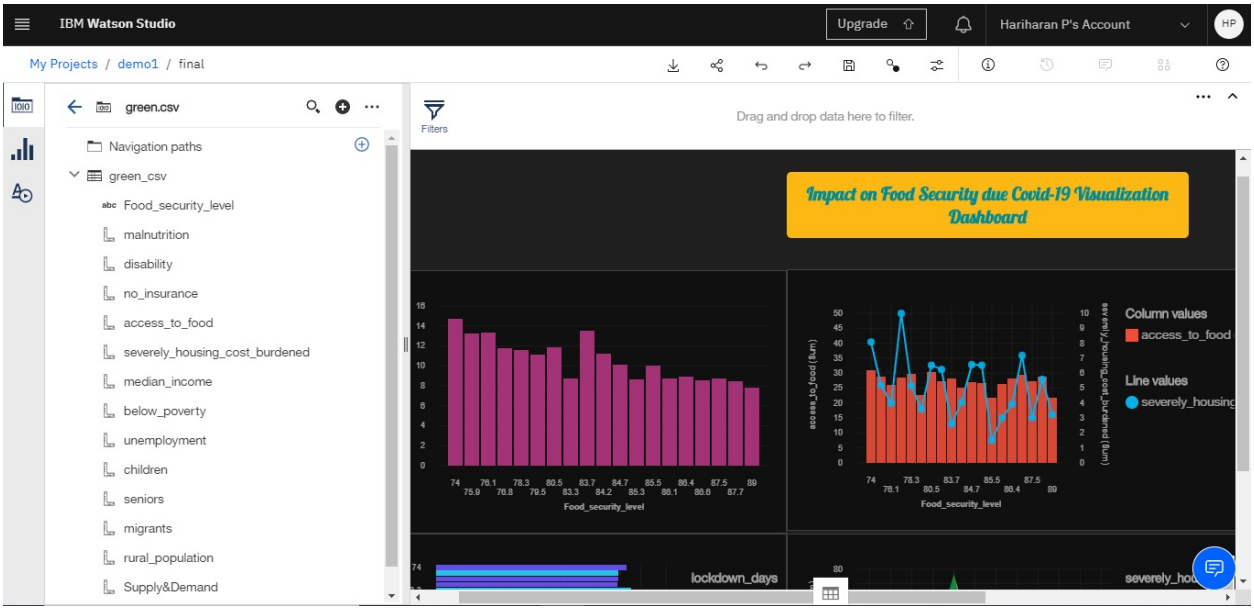
Testing the Model



Creating the Node Red Flow and Deploying the ML model



Creating a Visualization Dashboard using Watson studio



Deploying Visualization Dashboard and Food security level Prediction models in a

Webpage

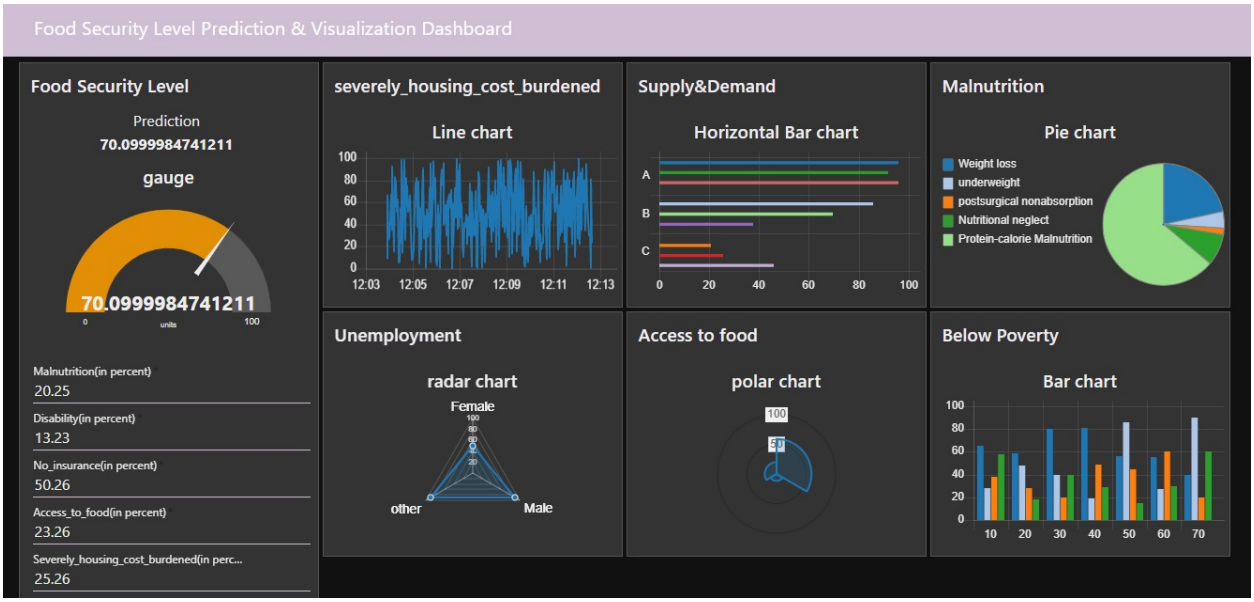
5. RESULT

1. We had deployed the **Food Security prediction level in Node RED** and **Visualization Dashboard using Cognos Dashboard.**
2. And both this UI link is implemented in a Webpage.

A. **Webpage**

B. Visualization Dashboard(using Cognos dashboard Watson Studio)

C. Food security level Prediction



6. PROJECT LINKS

1. Visualization dashboard
2. Food Security level Prediction
3. Webpage

7. CONCLUSION

This project work has provided us an excellent opportunity and experience, to use our limited knowledge we gained more practical knowledge and experience regarding, planning, designing, visualizing and deploying while doing this project work.

We are proud that we have completed the work within the time successfully. **“Impact on Food Security due to COVID-19- Visualization dashboard”** Is working with satisfactory condition. We have developed our ability and skills by making use of available facilities.