

Cereal Analysis Based On Ratings By Using Machine Learning Techniques With Ibm Watson

Project Mentor: Shivani Kapoor

Submitted By

Aditya Roshan 19BCE0911

Mayank Tiwary 19BCE0918

Aniket Mishra 19BCE2062

1. INTRODUCTION

1.1 Overview

Breakfast is the most important meal of the day by eating a nutritious breakfast you better chances of reaching the recommended five servings of fruits and vegetables a day and you're more likely to get all the nutrients you need. Usually a customer expects to consume dietary cereals with high proteins, fiber and low sugars, fats. Predicting a brand with high dietary cereals became a big issue.

The project objective is to find the high dietary food that is predicted on the basis of rating of the food.

1. To find which quantities are showing more impact on the rating of food.
2. To show the food which is impacting less on the rating of food?

We use machine learning algorithms to predict the food with a high beneficiary diet. The model can predict the rating of the food more accurately by giving the inputs which are the cereals and ingredients present in the food. Thus a customer can get high dietary food by the rating of the food given to it from the cereals and ingredients present. The rating is predicted using the neural networks model.

1.2 Purpose

The best breakfast cereals are rich in fiber, something most of us don't get enough of. Sitting down to a healthy, high fiber diet could be the key to maintaining or losing weight. So it is crucial to select the best cereal which has the most nutritious value , in this project we implement regression prediction models which would help us in selecting the most nutritious cereal.

2. LITERATURE SURVEY

2.1 Existing Problems

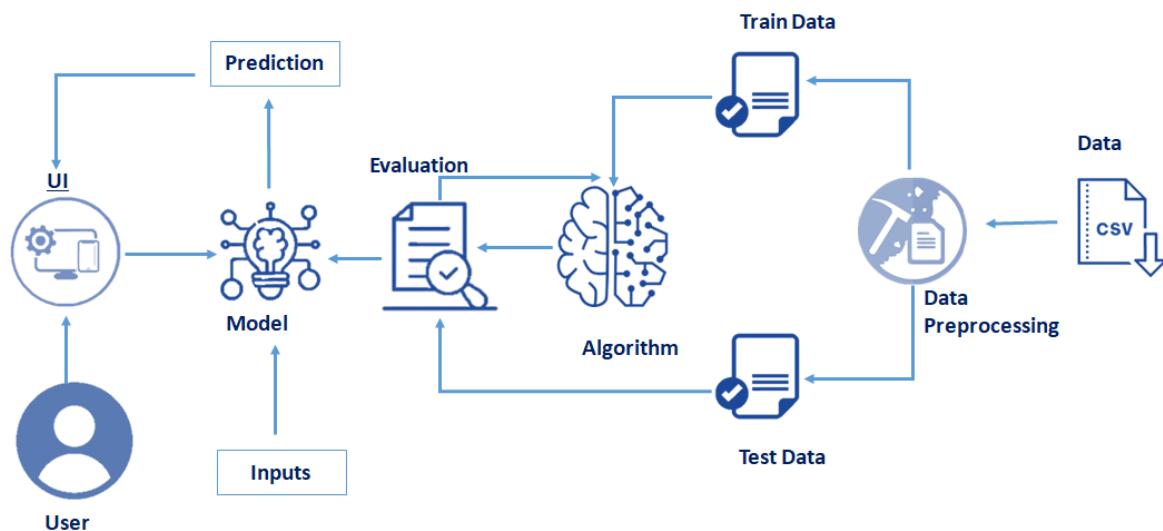
Different classifier models have been used in various classifications . In [1], Seng divided the work into two parts. In the first part they presented review on viticulture technology. In the second part they collected the dataset for image classification. They also applied different classifiers like SVM, KNN, and CNN on two different dataset. In [2], Gang proposed improved KNN algorithm. They pointed out the limitation of KNN i.e. it get affected by rating in music system. KNN is slow in real time as it have to keep track of all training data and find the neighbor nodes, whereas LR can easily extract output from the tuned θ coefficients. Neural networks need large training data and have huge computation time , making these not ideal for cereal analysis which has not much computations and does not a have a large data set.

2.2 Proposed solution

We have implemented linear regression (LR) in our project. Linear Regression is a regression model, it takes features and predict a continuous output, giving us a linear curve. LR is found to be much faster than KNN and CNN and they need less training dataset as compared to CNN. As for our requirements LR would be a better model to use as our dataset and computations are small.

3. THEORITICAL ANALYSIS

3.1 Block diagram



Here, the data from data-set is pre-processed before splitting it into training and testing data. Then linear regression classifier model is applied on both sets , then we make an interactive UI where user can give all the information(input) required to give the accuracy.

3.2 Hardware / Software designing

Hardware Requirements-

- Memory and disk space required per user: 1GB RAM + 1GB of disk + .5 CPU core.
- Server overhead: 2-4GB or 10% system overhead, .5 CPU cores, 10GB disk space.

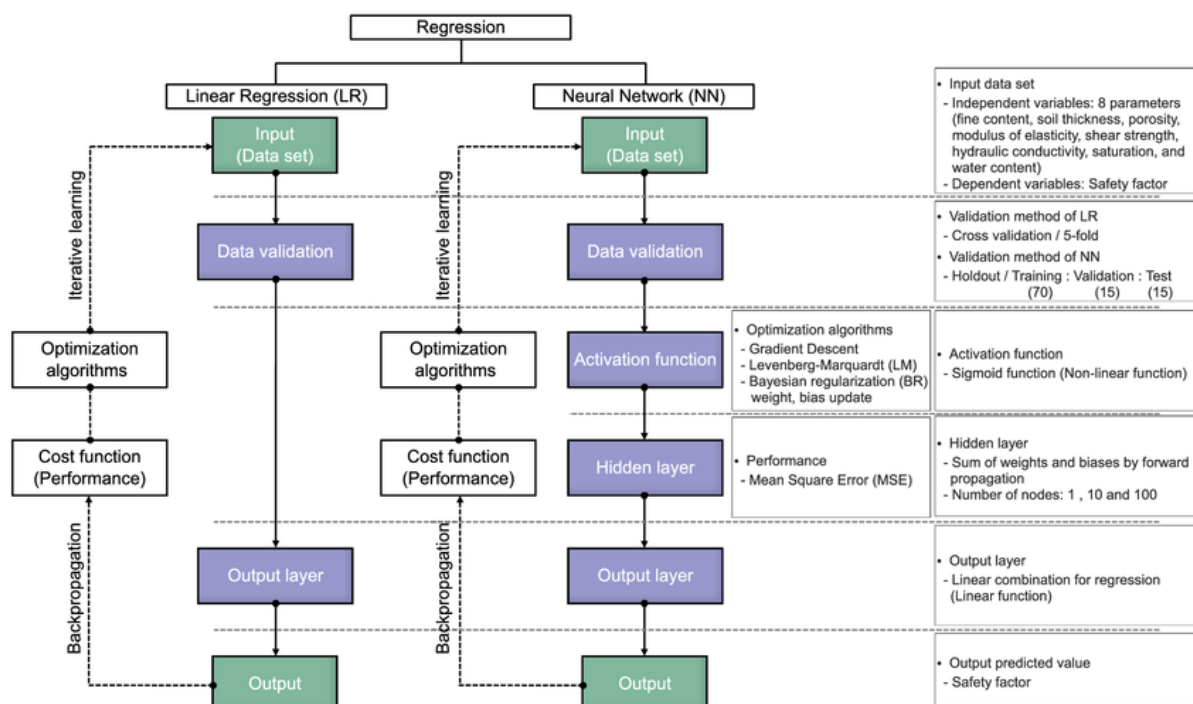
Software Requirements-

- Jupyter
- Mozilla Firefox/Google Chrome / Edge.

4. Experimental Investigations

When applying LabelEncoder the data was categorised efficiently and correlations found in the data was accurate. After pre-processing data was visualised in various plots such as heatmap , pairplot and boxplot which were produced in less time. Then data was transformed into training and testing sets , firstly linear regression was implemented on training set and later on the testing set. We found that the computations and data visualisations were done efficiently and quickly. And we serialised it so that it can be used directly as a .pkl file which is further used and implemented in the UI.

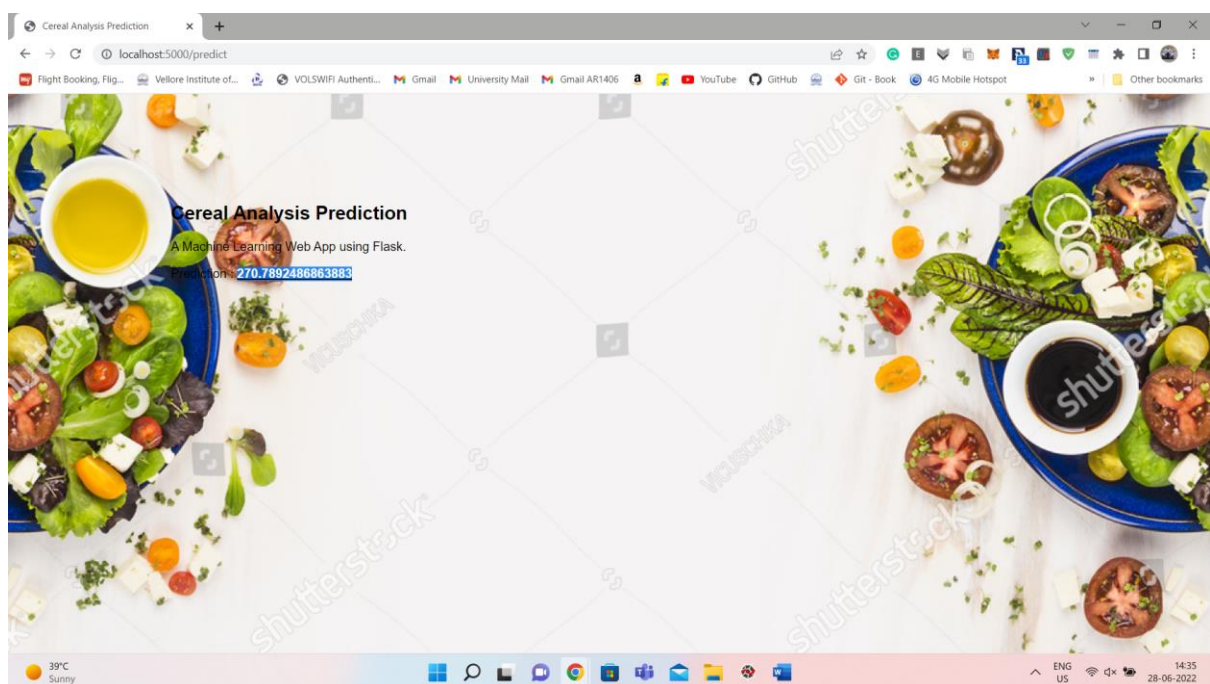
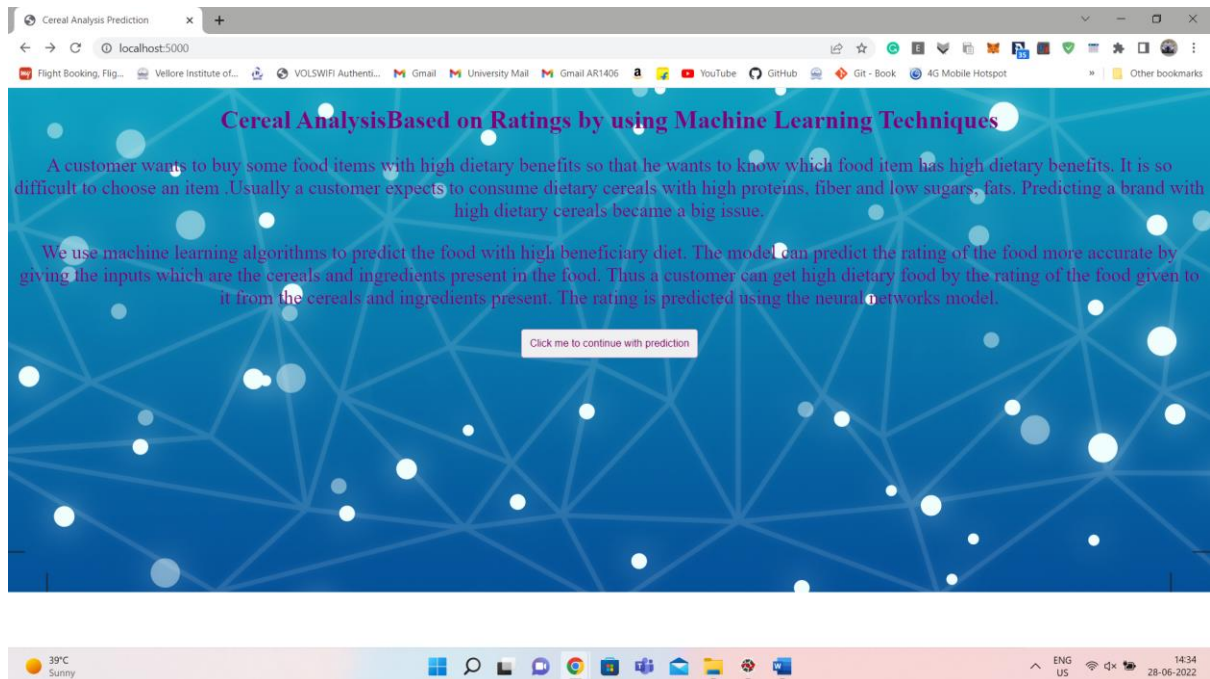
5. Flowchart

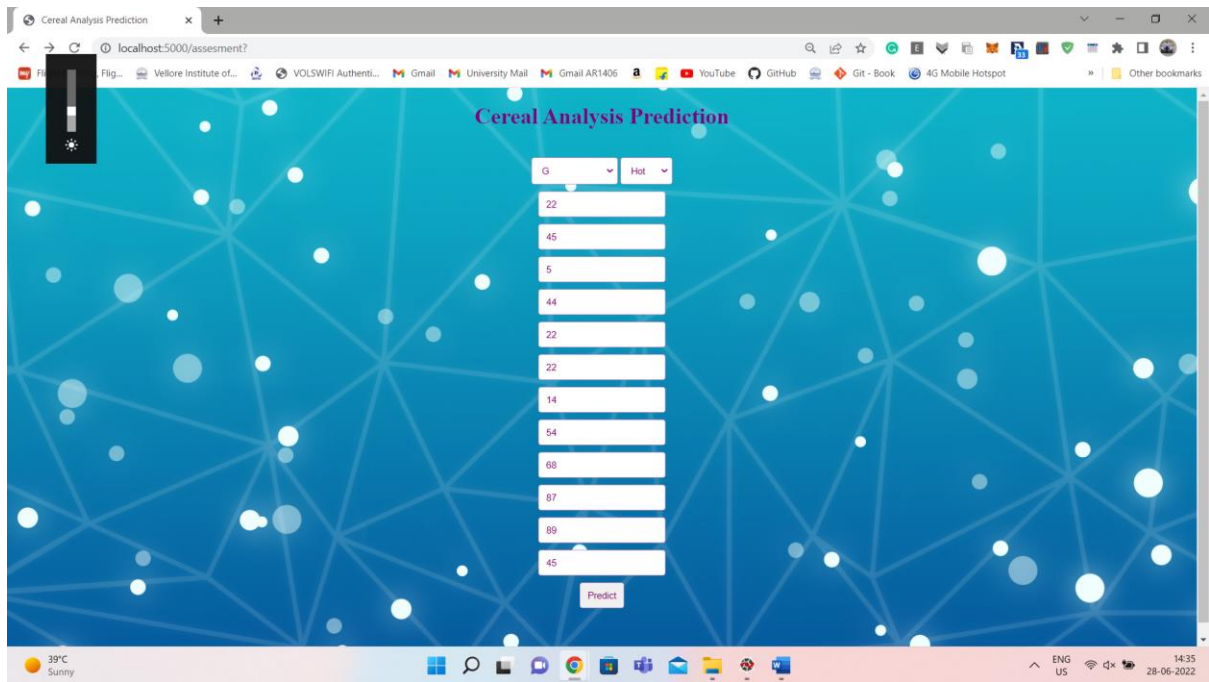


6. Result

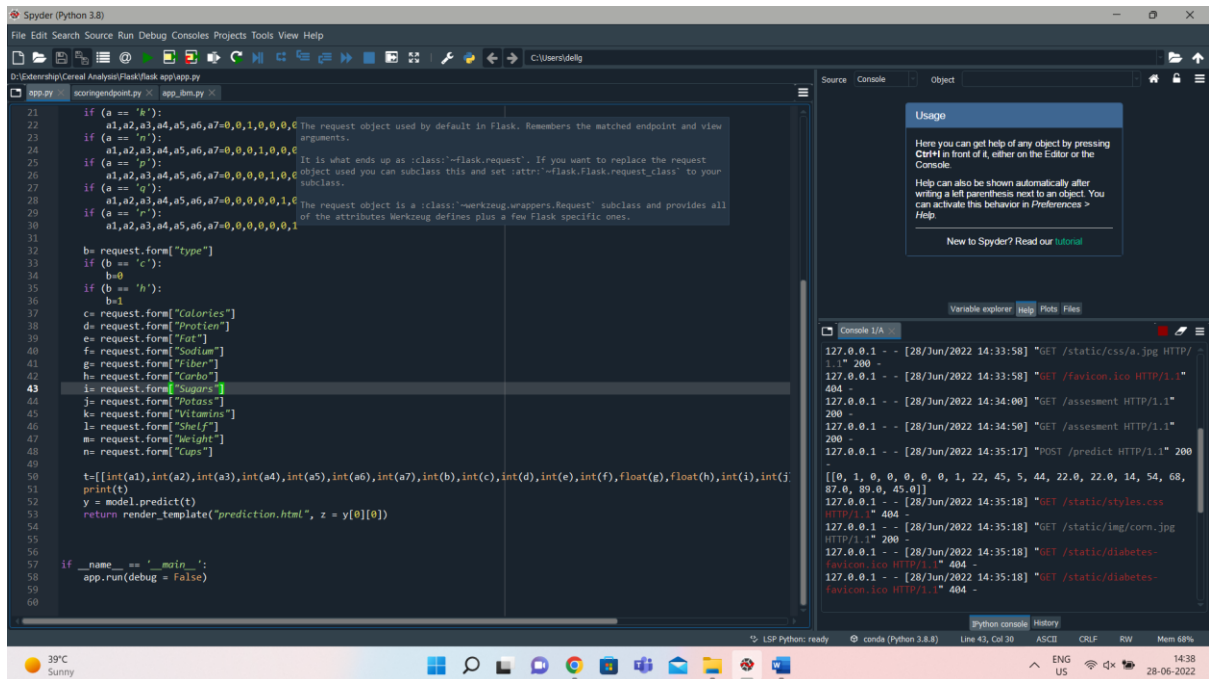
The program works by taking various aspects of the cereal the user wants as input and running it through machine learning algorithm that predicts the food with high beneficiary diet. The model can predict the rating of the food more accurately if the user gives inputs

on the ingredients present. The rating is predicted using at the neural network model.

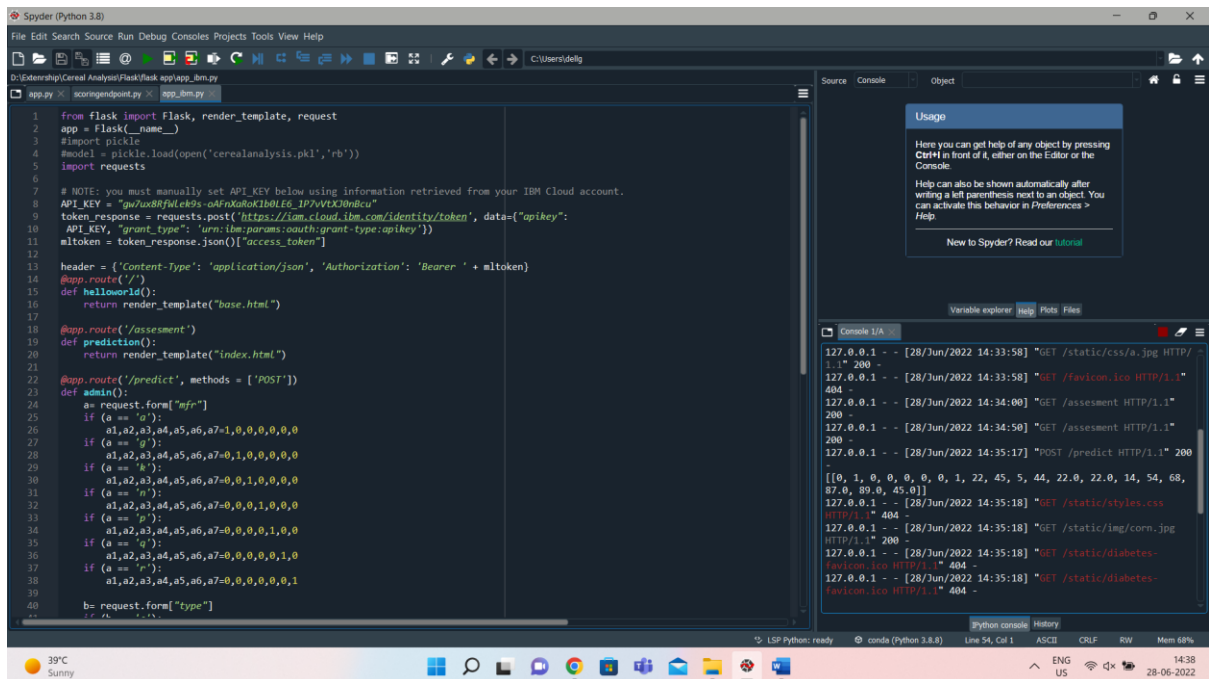




FLASK APP



FLASK APP FOR IBM

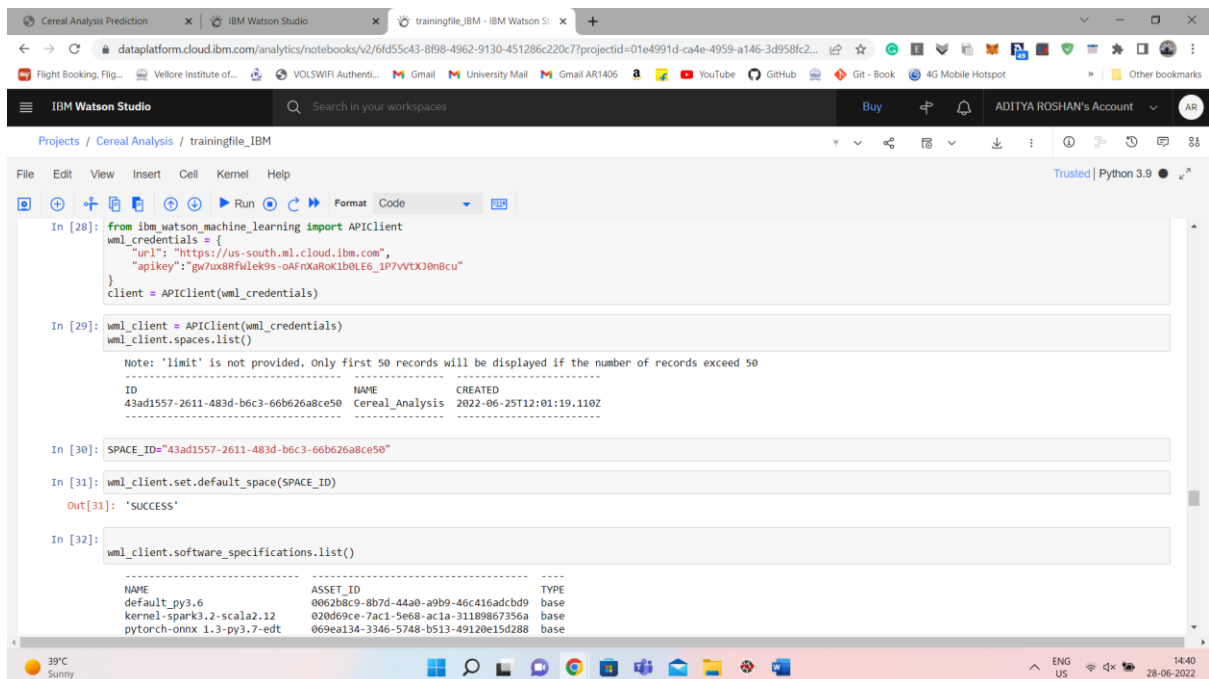


```
1 from flask import Flask, render_template, request
2 app = Flask(__name__)
3 #import pickle
4 model = pickle.load(open('cerealanalysis.pkl', 'rb'))
5 import requests
6
7 # NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
8 API_KEY = "gw7ux8Rfulek9s-oAFnXaRokIb0LE6_1P7vVtXJ0n8Cu"
9 token_response = requests.post("https://iam.cloud.ibm.com/identity/token", data={"apikey":
10 API_KEY, "grant_type": "urn:ibm:params:oauth:grant-type:apikey"})
11 mtoken = token_response.json()["access_token"]
12
13 header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mtoken}
14 @app.route('/')
15 def helloworld():
16     return render_template("base.html")
17
18 @app.route('/assessment')
19 def prediction():
20     return render_template("index.html")
21
22 @app.route('/predict', methods = ['POST'])
23 def admin():
24     a= request.form["mfr"]
25     if (a == 'a'):
26         a1,a2,a3,a4,a5,a6,a7=1,0,0,0,0,0,0
27     if (a == 'g'):
28         a1,a2,a3,a4,a5,a6,a7=0,1,0,0,0,0,0
29     if (a == 'h'):
30         a1,a2,a3,a4,a5,a6,a7=0,0,1,0,0,0,0
31     if (a == 'n'):
32         a1,a2,a3,a4,a5,a6,a7=0,0,0,1,0,0,0
33     if (a == 'p'):
34         a1,a2,a3,a4,a5,a6,a7=0,0,0,0,1,0,0
35     if (a == 'q'):
36         a1,a2,a3,a4,a5,a6,a7=0,0,0,0,0,1,0
37     if (a == 'r'):
38         a1,a2,a3,a4,a5,a6,a7=0,0,0,0,0,0,1
39
40     b= request.form["type"]
41     c= b, a1, a2, a3, a4, a5, a6, a7
```

Console output:

```
127.0.0.1 - - [28/Jun/2022 14:33:58] "GET /static/css/a.jpg HTTP/1.1" 200 -
127.0.0.1 - - [28/Jun/2022 14:33:58] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [28/Jun/2022 14:34:00] "GET /assessment HTTP/1.1" 200 -
127.0.0.1 - - [28/Jun/2022 14:34:50] "GET /assessment HTTP/1.1" 200 -
127.0.0.1 - - [28/Jun/2022 14:35:17] "POST /predict HTTP/1.1" 200 -
[[0, 1, 0, 0, 0, 0, 0, 1, 22, 45, 5, 44, 22.0, 22.0, 14, 54, 68, 87.0, 89.0, 45.0]]
127.0.0.1 - - [28/Jun/2022 14:35:18] "GET /static/styles.css HTTP/1.1" 404 -
127.0.0.1 - - [28/Jun/2022 14:35:18] "GET /static/img/corn.jpg HTTP/1.1" 200 -
127.0.0.1 - - [28/Jun/2022 14:35:18] "GET /static/diabetes-favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [28/Jun/2022 14:35:18] "GET /static/diabetes-favicon.ico HTTP/1.1" 404 -
```

DEPLOYMENT USING IBM



```
In [28]: from ibm_watson_machine_learning import APIClient
wml_credentials = {
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey": "gw7ux8Rfulek9s-oAFnXaRokIb0LE6_1P7vVtXJ0n8Cu"
}
client = APIClient(wml_credentials)

In [29]: wml_client = APIClient(wml_credentials)
wml_client.spaces.list()

Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50
-----
ID NAME CREATED
-----
43ad1557-2611-483d-b6c3-66b626a8ce50 Cereals_Analysis 2022-06-25T12:01:19.110Z

In [30]: SPACE_ID="43ad1557-2611-483d-b6c3-66b626a8ce50"

In [31]: wml_client.set_default_space(SPACE_ID)
Out[31]: 'SUCCESS'

In [32]: wml_client.software_specifications.list()

-----
NAME ASSET_ID TYPE
-----
default_py3.6 0062b8c9-8b7d-44a0-a9b9-46c416adcbd9 base
kernel-spark3.2-scala2.12 020d09ce-7ac1-5e68-ac1a-31189867356a base
pytorch-onnx 1.3-py3.7-edt 069ea134-3346-5748-b513-49120e15d288 base
```


The screenshot shows the IBM Watson Studio web interface. The browser address bar displays a URL from dataplatform.cloud.ibm.com. The interface includes a top navigation bar with 'IBM Watson Studio' and a search bar. Below this, a breadcrumb trail shows 'Projects / Cereal Analysis / trainingfile_IBM'. The main area contains a Jupyter Notebook with a code cell 'In [36]: model_details' and an output cell 'Out[36]:'. The output is a JSON object representing model details, including fields like 'entity', 'label_column', 'software_spec', 'name', 'training_data_references', 'endpoint_url', 'secret_access_key', 'id', 'location', and 'schema'. The 'schema' field contains a list of 16 fields, each with a name and a type of 'float'. The bottom status bar shows '39°C Sunny' and the date '28-06-2022'.

```
In [36]: model_details
Out[36]: {'entity': {'hybrid_pipeline_software_specs': [],
  'label_column': 'l0',
  'software_spec': {'id': '12b83a17-24d8-5082-900f-0ab31fbfd3cb',
    'name': 'runtime-22.1-py3.9'},
  'training_data_references': [{'connection': {'access_key_id': 'not_applicable',
    'endpoint_url': 'not_applicable',
    'secret_access_key': 'not_applicable'},
    'id': '1',
    'location': {}},
    'schema': {'fields': [{'name': 'f0', 'type': 'float'},
      {'name': 'f1', 'type': 'float'},
      {'name': 'f2', 'type': 'float'},
      {'name': 'f3', 'type': 'float'},
      {'name': 'f4', 'type': 'float'},
      {'name': 'f5', 'type': 'float'},
      {'name': 'f6', 'type': 'float'},
      {'name': 'f7', 'type': 'float'},
      {'name': 'f8', 'type': 'float'},
      {'name': 'f9', 'type': 'float'},
      {'name': 'f10', 'type': 'float'},
      {'name': 'f11', 'type': 'float'},
      {'name': 'f12', 'type': 'float'},
      {'name': 'f13', 'type': 'float'},
      {'name': 'f14', 'type': 'float'},
      {'name': 'f15', 'type': 'float'},
      {'name': 'f16', 'type': 'float'}]}
```

DEPLOYED USING IBM

The screenshot shows the IBM Watson Studio web interface with a Jupyter Notebook. The code cell 'In [40]:' contains a deployment command: `# Deploy deployment = wml_client.deployments.create(artifact_uid=model_uid, meta_props=deployment_props)`. The output cell shows the deployment process: 'Synchronous deployment creation for uid: '450fe645-df77-4b99-892d-6e6497a14e5a' started', followed by 'initializing', a note about 'online_url' being deprecated, and 'ready'. The final output is 'Successfully finished deployment creation, deployment_uid='024830c7-499c-436e-92a2-edaceb44c487''. The bottom status bar shows '39°C Sunny' and the date '28-06-2022'.

```
In [40]: # Deploy
deployment = wml_client.deployments.create(
    artifact_uid=model_uid,
    meta_props=deployment_props
)

#####

Synchronous deployment creation for uid: '450fe645-df77-4b99-892d-6e6497a14e5a' started
#####

initializing
Note: online_url is deprecated and will be removed in a future release. Use serving_urls instead.
ready

-----
Successfully finished deployment creation, deployment_uid='024830c7-499c-436e-92a2-edaceb44c487'
```

SCREENSHOTS OF THE WEBSITE DEPLOYED IS SAME AS ABOVE.

7. ADVANTAGES & DISADVANTAGES

Advantages :

- Easy and simple implementation.
- Takes less space than other models.
- Fast training.
- Value of θ coefficients gives an assumption of feature significance.

Disadvantages :

- Applicable only if the solution is linear. In many real life scenarios, it may not be the case.
- Algorithm assumes the input residuals (error) to be normal distributed, but may not be satisfied always.
- Algorithm assumes input features to be mutually independent(no co-linearity).

8.Applications

- Linear regressions can be used in business to evaluate trends and make estimates or forecasts. For example, if a company's sales have increased steadily every month for the past few years, by conducting a linear analysis on the sales data with monthly sales, the company could forecast sales in future months.
- Linear regression can also be used to analyze the marketing effectiveness, pricing and promotions on sales of a product. For instance, if company XYZ, wants to know if the funds that they have invested in marketing a particular brand has given them substantial return on investment, they can use linear regression.

- Linear Regression can be also used to assess risk in financial services or insurance domain. For example, a car insurance company might conduct a linear regression to come up with a suggested premium table using predicted claims to Insured Declared Value ratio.
- Predicting crop yields based on the amount of rainfall. Yield is a dependent variable while the measure of precipitation is an independent variable.

9. Conclusion

The assignment of the data to training and test set is done using random sampling. We perform random sampling on R using `sample()` function. We have used `set.seed()` to generate same random sample everytime and maintain consistency. We will use the index variable while fitting neural network to create training and test data sets. Then we fit a neural network on our data. We use `neuralnet` library for the analysis. The first step is to scale the cereal dataset. The scaling of data is essential because otherwise a variable may have large impact on the prediction variable only because of its scale. Using unscaled may lead to meaningless results. The common techniques to scale data are: min-max normalization, Z-score normalization, median and MAD, and tan-h estimators. The min-max normalization transforms the data into a common range, thus removing the scaling effect from all the variables. Unlike Z-score normalization and median and MAD method, the min-max method retains the original distribution of the variables. We use min-max normalization to scale the data. A Neural network is inspired from biological nervous system. Similar to nervous system the information is passed through layers of processors. The significance of variables is represented by weights of each connection. The article provides basic understanding of back propagation algorithm, which is used to assign these weights. The aim is to predict the rating of cereals using information such as calories, fat, protein etc. After constructing the

neural network we evaluate the model for accuracy and robustness. We compute RMSE and perform cross-validation analysis. In cross validation, we check the variation in model accuracy as the length of training set is changed. We show that model accuracy increases when training set is large. Before using the model for prediction, it is important to check the robustness of performance through cross validation.

10. Future Scope

For future work we hope to implement Structural Equation Modelling (SEM) which offers various pros when compared to multiple linear regression and helps overcome some of the disadvantages we faced.

There are two main differences between regression and structural equation modelling. The first is that SEM allows us to develop complex path models with direct and indirect effects. This allows us to more accurately model causal mechanisms we are interested in. The second key difference is to do with measurement. In SEM we assume that our actual variables are indicators of underlying constructs (for example, 'I like school' is an indicator of attitudes to school), and we can incorporate that measurement model directly into SEM. This again means that we are more accurately modelling the phenomena we want to explain.

11.Bibliography

11.1 References

1. K.P. Seng, Member, IEEE, L.M. Ang, Senior Member, IEEE, Leigh M. Schmidtke, Suzy Y. Rogiers, "Computer Vision and Machine Learning for Viticulture Technology", Accepted manuscript for IEEE Access, October 2018.
2. G. Li and J. Zhang, "Music personalized recommendation system based on improved KNN algorithm," 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, 2018, pp. 777-781.
3. Suhartanto, D.; Helmi Ali, M.; Tan, K.H.; Sjahroeddin, F.; Kusdibyo, L. Loyalty toward Online Food Delivery Service: The Role of E-Service Quality and Food Quality. J. Foodserv. Bus. Res. 2019, 22, 81–97. [CrossRef]
4. Ara, J.; Hasan, M.T.; Al Omar, A.; Bhuiyan, H. Understanding Customer Sentiment: Lexical Analysis of Restaurant Reviews. In Proceedings of the 2020 IEEE Region 10 Symposium (TENSYP), Dhaka, Bangladesh, 5–7 June 2020; pp. 295–299.
5. Parliament of Australia. Population and Migration Statistics in Australia. Available online: https://www.aph.gov.au/About_Parliament/Parliamentary_Departments/Parliamentary_Library/pubs/rp/rp1819/Quick_Guides/PopulationStatistics (accessed on 10 May 2022).