

---

# Project Report (Phase 3)

Department of Computer Science and Engineering

University at Buffalo, Buffalo, NY 14260

---

**Team:** Desireddy Sai Sankeerthana

[saisanke@buffalo.edu](mailto:saisanke@buffalo.edu)

Vamshi Jamalpur

[vamshija@buffalo.edu](mailto:vamshija@buffalo.edu)

## Prediction of sales

### A. Instructions to use the webapplication as follows:

If you haven't downloaded streamlit run "pip install streamlit" command in the terminal(All the files mentioned below are present in Phase-3 sub-directory).

- 1) Open and Run the main.ipynb file
- 2) Open and run model.ipynb file
- 3) Make sure that you're in the Phase 3 directory in the terminal before executing app.py
- 4) Open app.py
- 5) Open the terminal with correct directory where app.py is located and run the command " streamlit run app.py "
- 6) It will display the link to the local host as follows  
Local Url : " <http://localhost:8501> "  
Network Url : " <http://192.168.1.204:8501> "  
Note: the above Local host link is the one we got for our device it might vary by devices
- 7) If it doesn't open automatically you can click on hyperlink in terminal it will navigate to webpage
- 8) Now you're in the web application. On the left, you can find the side bar with predict sales on it and on the right side you can see the main page with the title sales prediction app.

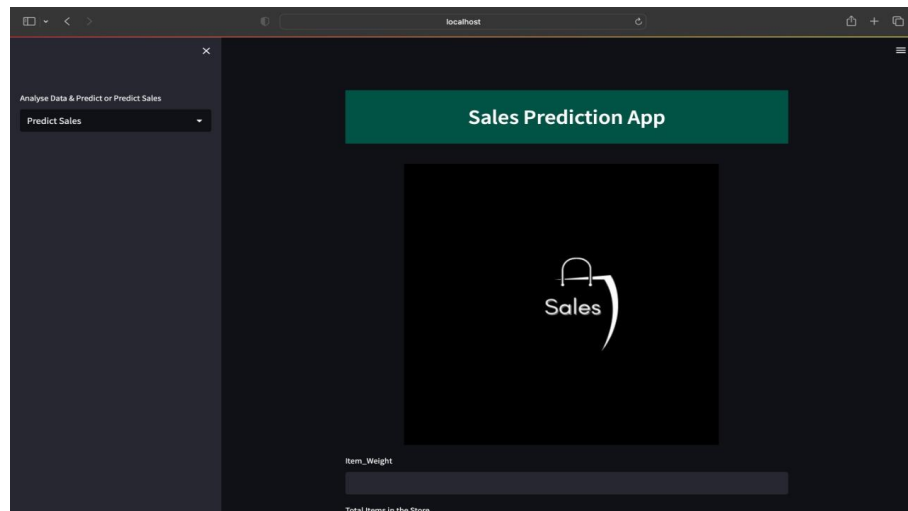


Fig 1

When you go side bar options you can find predict sales,analyse data & predict as options.

- 9) Click on predict sales option and enter item\_weight,total items in the store(it means total items present in retailer store),Item quantities in the store(the quantity of item present in the store ),Item\_Type (Item which he wants to predict),Item MRP(price of the item),Outlet\_establishment\_year(the year which the store was established),Outlet\_Type(Type of store),Outlet\_location(location in which the store in present),Outlet\_size(size of the store) and click on predict button.it will predict the expected number of sales for given item as input.

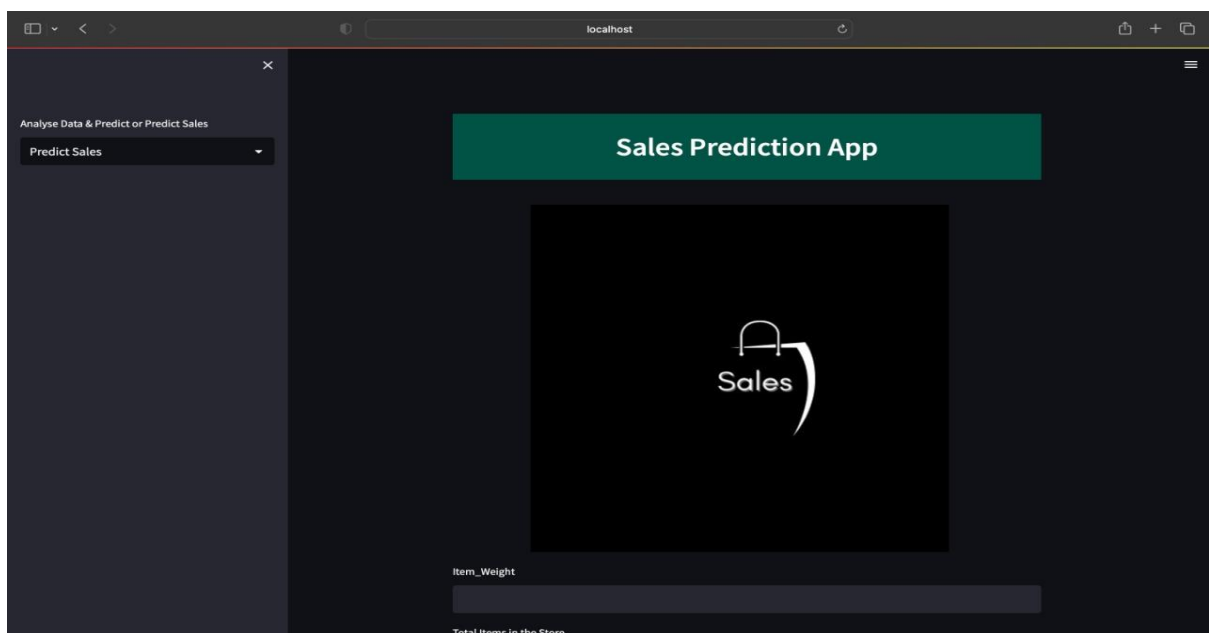


Fig2

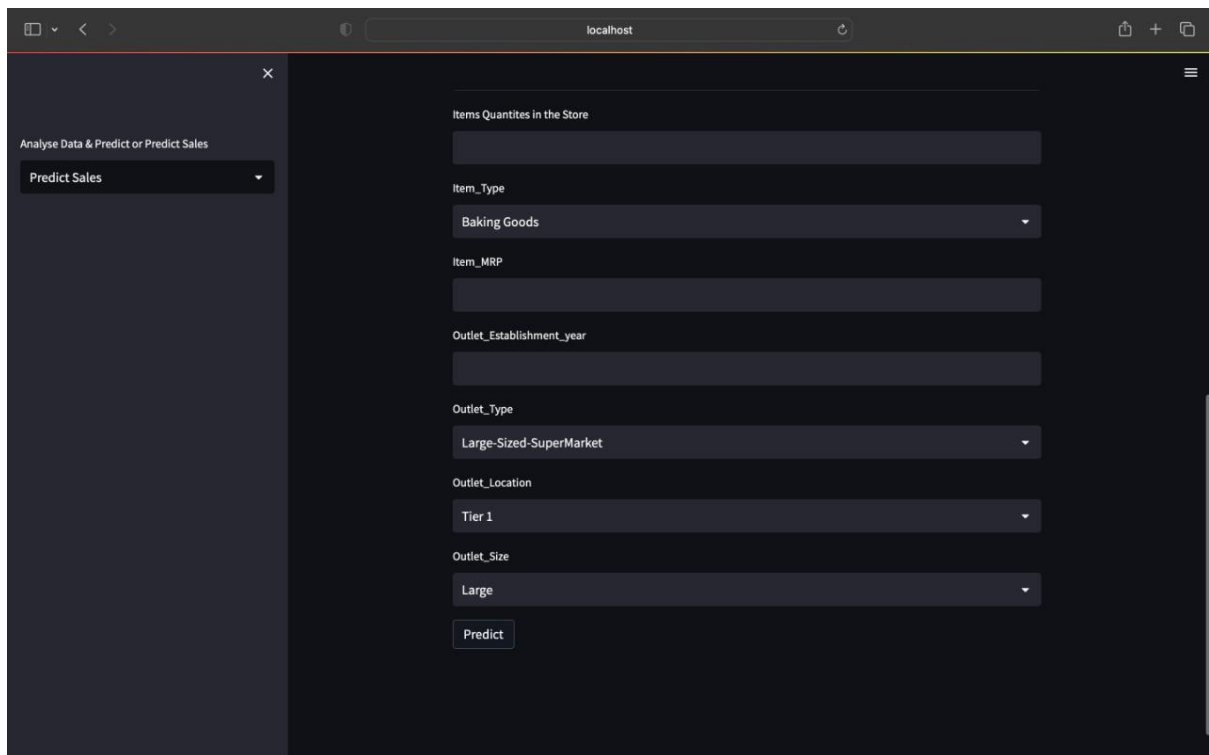


Fig 3

- 10) click on the sidebar and choose analyse data &predict option this will open analyse data & predict page

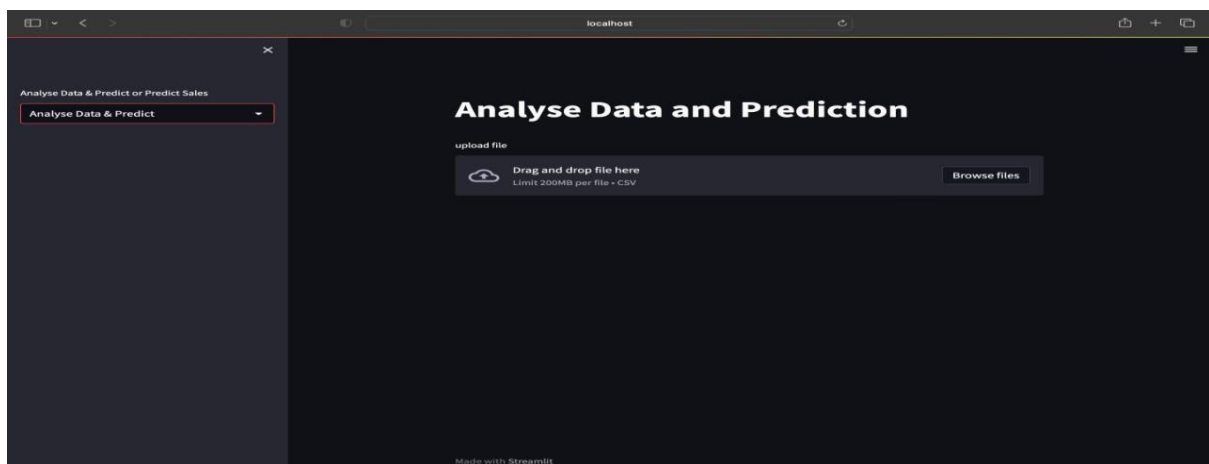


Fig 4

- 11) click on browse file option to upload your file(we have provided some sample .csv files in “sample csv files to test in ui” directory in Phase 3 Sub-directory) and when you upload your file you can find a new select box with default option as show visualization and when you scroll down you can find visualization of your

data. Note: the dataset should be in .csv file and it should not contain any null values.it should only have

columns :Item\_Weight , Item\_Visibility , Item\_Type , Item\_MRP ,  
Outlet\_size,Outlet\_Location\_Type,Outlet\_Type,  
Outlet\_Establishment\_year,Item\_Outlet\_sales.

12) On tapping on predict box you can predict the data

The screenshot shows a web application titled "Analyse Data and Prediction" running on localhost. The interface is dark-themed. On the left, there is a sidebar with a toggle for "Analyse Data & Predict or Predict Sales" and a dropdown menu currently set to "Analyse Data & Predict". The main content area has a header "Analyse Data and Prediction". Below this, there is an "upload file" section with a "Drag and drop file here" instruction, a "Limit 200MB per file • CSV" note, and a "Browse files" button. A file named "r1.csv" (2.7KB) is shown as uploaded. Below the upload section, there is a "Choose" dropdown menu set to "Predict Sales". Underneath, there are several input fields: "Item\_Weight", "Total Items in the Store", "Items Quantities in the Store", "Item\_Type" (set to "Baking Goods"), and "Item\_MRP".

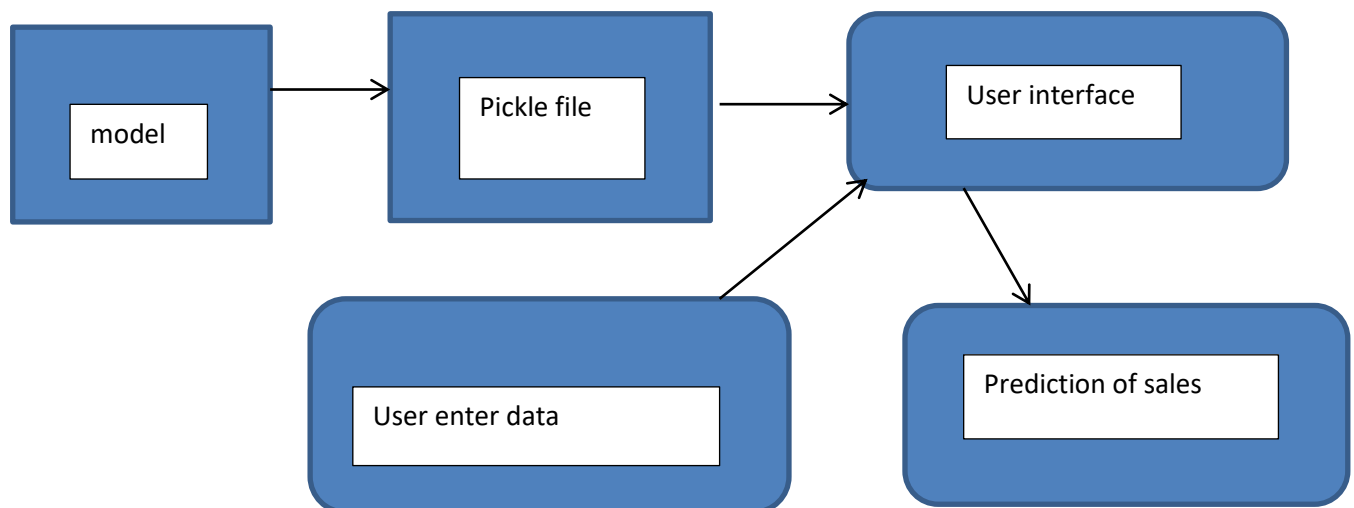
13) After predicting the data in analyse data &predict page when you go to predict sales page from sidebar all the fields will be filled with data that user had provided in analyse data & predict and by clicking on predict user can get end results of expected sales with web application dataset.

## **EXPLANATION FOR FINISHED DATA PRODUCT & UI:**

After using different models in Phase2 such as KNN(K-Nearest Neighbour) Regression , Linear Regression, Random Forest Regression, LASSO(Least Absolute Shrinkage and Selection Operator),Ridge Regression ,Decision Tree Regression.

We have achieved effective results for RMSE (Root mean Squared Error) ,MAE(Mean Absolute value),R2\_Score in Random Forest Regression and Decision Tree Regression. Hence, we finalized to use Random Forest Regression from Phase2 to Phase3 . In Phase 2, Random Forest has provided us with accuracy of 62%.Which is much efficient when compared to other algorithms.we have passes number estimators( $n\_estimators = 400$  ), number of jobs( $n\_jobs=4$ ), maximum depth( $max\_depth=7$ ) ,number of leafs( $min\_sample\_leaf=100$ ) as parameters for Randomforest Regression. Then we fit the model by passing values and store them in a pickle file and used that pickle file to load data in app.py.

We have created the web application that helps us in prediction of sales by using streamlit. It's a open source framework for machine learning. We have used random forest regression model in this phase to split the data into and made them fit into it .A pickle file is used to save the model and reduce length of re-training the model. Now we store the values into a pickle file named (saved\_file.pkl) in model.ipynb . Then we use that pickle file in app.py to do the front end work.



Predict Sales page:

In this UI page , end user can enter values for Item\_Weight , Total items in the store, Quantities of the item present the store,Item\_type,Item\_MRP,Outlet\_establishment\_year,Outlet\_Type,Outlet\_Location,Outlet\_size after entering all the input values by clicking on the button predict it results in expected

number of sales for the values entered. we have also done validation for each input that is item weight,totat items in the store,item quantities ,item MRP should be in numerical values and outlet establishment year should be from 1950-current year.

Working Function for Predict sales Page:

All the data that has been entered by end user is stored in array form in predict\_sales.py .then we send this array of data to the model that we retrieved from the pickle file and then the model will predict and displays the respective results in the UI(user interface).

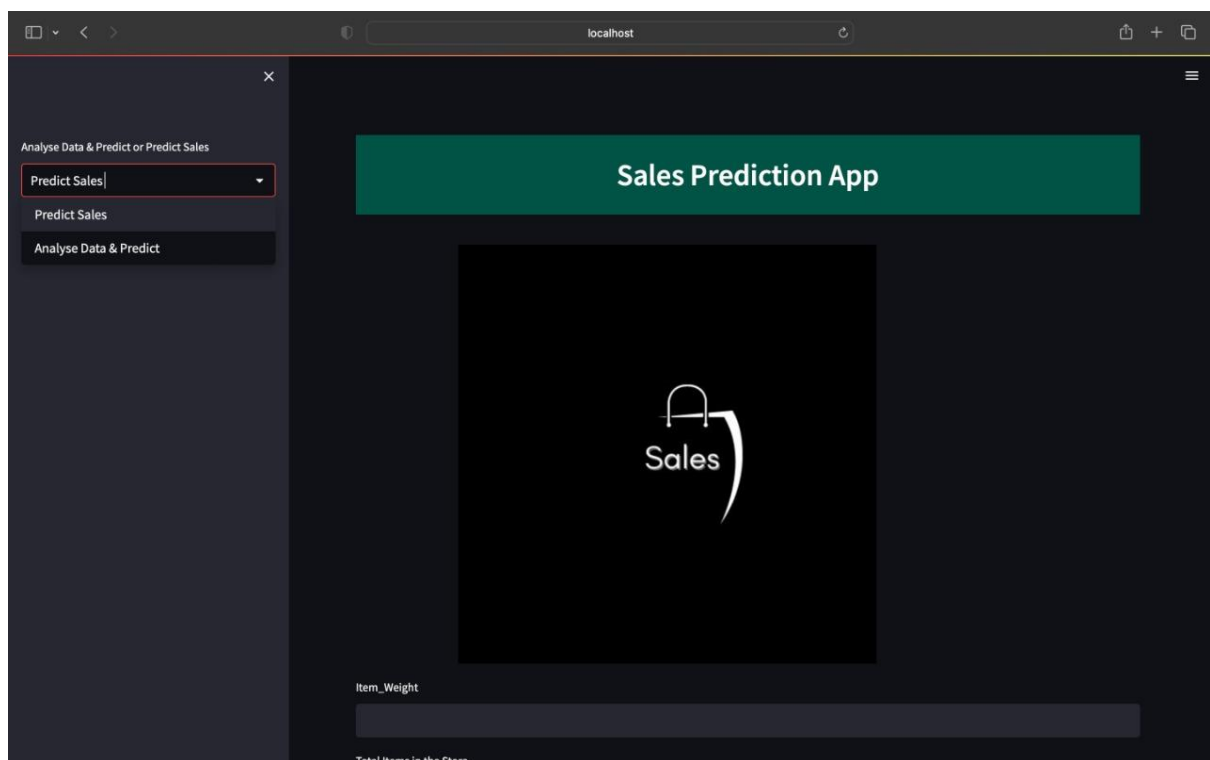


Fig1: prediction of sales page

Analyse Data & predict page:

Show visualisation:

In this UI page, end user can upload the dataset of their own.the dataset should be in .csv file and it should not contain any null values.it should only have columns :Item\_Weight ,

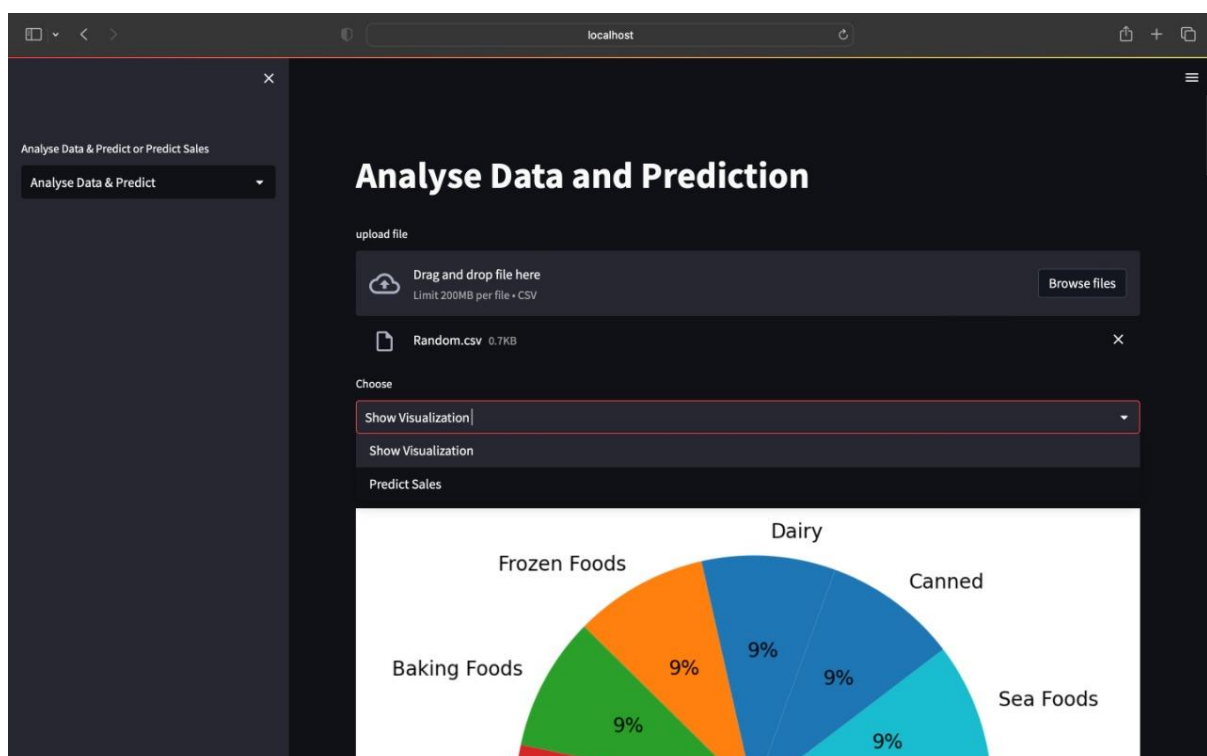
Item\_Visibility , Item\_Type , Item\_MRP , Outlet\_size,Outlet\_Location\_Type,Outlet\_Type, Outlet\_Establishment\_year,Item\_Outlet\_sales. After uploading the dataset their exist two options show visualisation and predict sales .By clicking on show visualisation ,end user can find some bar charts ,pie chart and relation between attributes for the dataset uploaded.

Predict Sales:

By clicking on predict sales , end user can predict the sales by entering values for Item\_weight, Total items in the store, Item Quantities in the store,Item\_type,Item\_MRP,Outlet\_establishment\_year,Outlet\_Type,Outlet\_Location,Outlet\_size after entering all the required inputs and clicking on predict button results in expected sales of product for respective results.

Working function for Analyse Data and predict page:

In the UI, there is another page present which is analysis and predict of data(Explore\_analysis.py). In this page, end user can upload the dataset of their own to see the viualization of the their dataset . In visualization ,we will displaying bar chart, pie chart and relation between the attributes. From this page, user can also predict sales for their own data by going to predict sales. In predict sales, after the end user enter the data the data is stored in an array and sent to model that is retrieved from the pickle file and then model will train the data and predicts the sales and displays the results in the UI





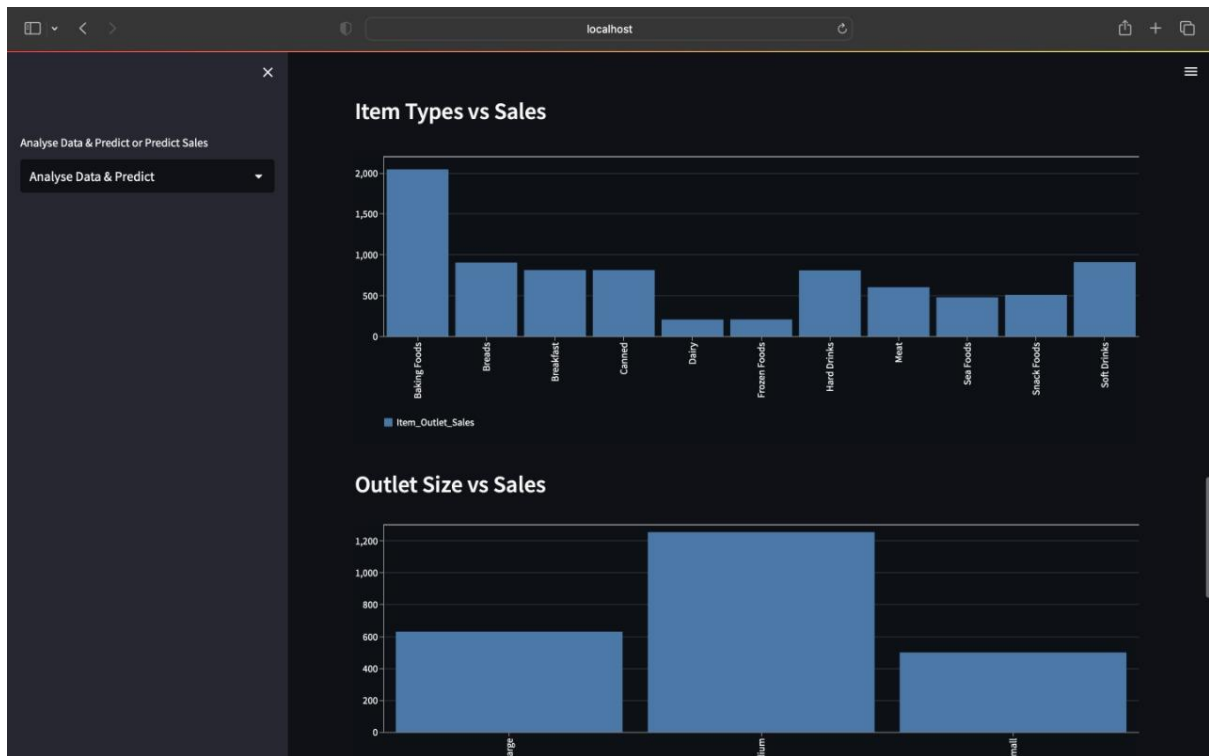


Fig 2: visualisation of data

The above figure represents bar chart, pie chart for uploaded dataset.

The figure shows a web application interface for data analysis and prediction. The interface includes a sidebar with navigation options, a main content area with a file upload section, and a prediction section with various input fields.

**File Upload Section:**

- Drag and drop file here (Limit 200MB per file • CSV)
- Browse files
- Random.csv 0.7KB

**Prediction Section:**

- Choose: Predict Sales
- Item\_Weight: [Input field]
- Total Items in the Store: [Input field]
- Items Quantities in the Store: [Input field]
- Item\_Type: Baking Goods
- Item\_MRP: [Input field]
- Outlet\_Establishment\_year: [Input field]
- Outlet\_Type: [Input field]

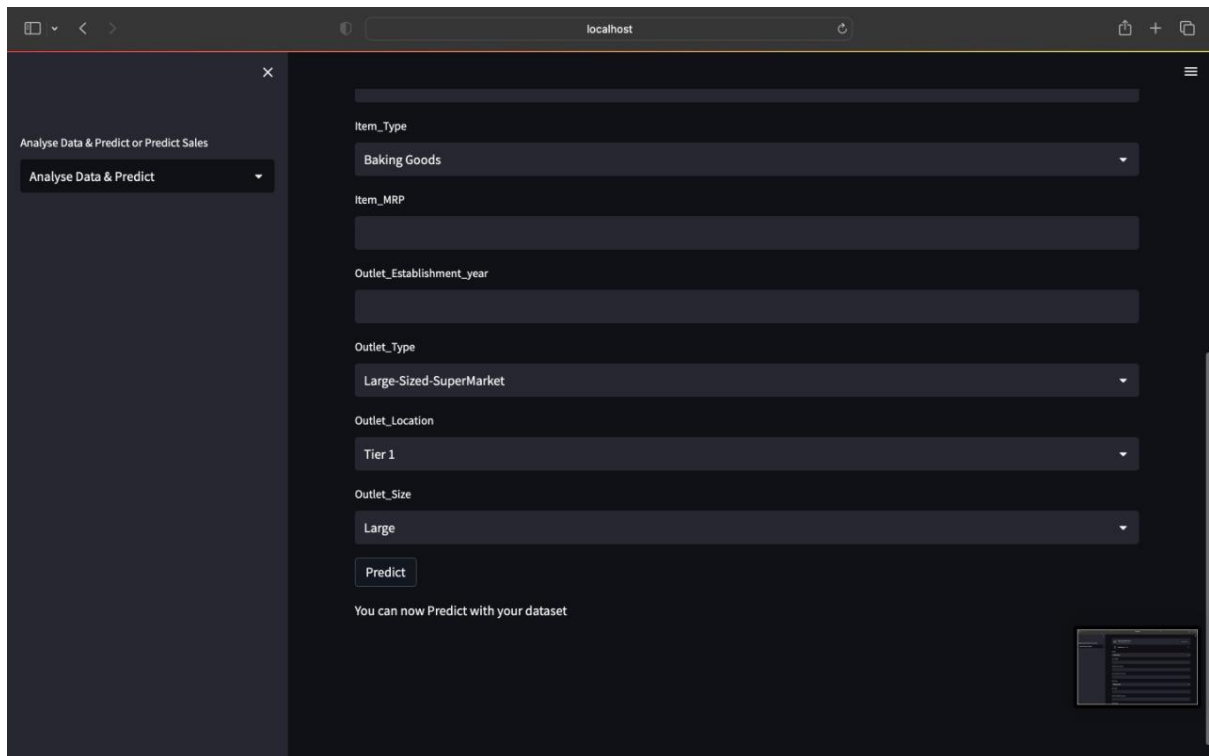
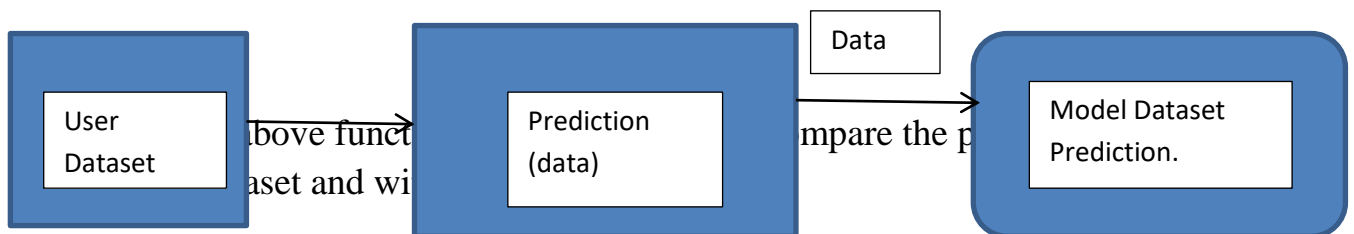


Fig 3: Analyse data & Predict Page

### Prediction of user data and web data:



The screenshot shows a web application running on localhost. On the left, a sidebar contains the text "Analyse Data & Predict or Predict Sales" and a dropdown menu with "Analyse Data & Predict" selected. The main area is titled "Items Quantities in the Store" and contains several input fields: a text field with "49", a dropdown for "Item\_Type" with "Breads" selected, a text field with "23", a text field with "2011", a dropdown for "Outlet\_Type" with "Medium-Sized-SuperMarket" selected, a dropdown for "Outlet\_Location" with "Tier 2" selected, and a dropdown for "Outlet\_Size" with "Medium" selected. Below these fields is a "Predict" button. A green box displays the result: "The Expected number of sales are 650.72". At the bottom, it says "You can now Predict with your dataset".

Fig4: Prediction with end user data

The screenshot shows the same web application interface as Fig4, but with the dropdown menu in the sidebar set to "Predict Sales". The input fields and the "Predict" button are the same. The green box now displays the result: "The Expected number of sales are 3842.34".

Fig5: prediction with web application data

By the prediction of both the data's end user can understand what kind of dataset he has entered or uploaded. In the fig4 prediction with end user data set,

User has entered Item\_weight as 10 ,total items in the store as 100,item quantities in the store as 49,Item\_type as breads,Item\_MRP as 23,Outlet\_Establishment\_year as 2011,Outlet\_Type as medium-sized supermarket,Outlet\_location as Tier 2,Outlet\_size as Medium by clicking predict it shows expected result as 650.72.In fig5 Prediction with web application data , user has entered same data as in fig4 and got result as 3842.34 for prediction with web application dataset.

### **C. RECOMMENDATIONS RELATED TO OUR PROBLEM STATEMENT:**

#### **Learn from product:**

Users who use our application can increase their sales by analysing attributes such as Item\_Weight-which helps the users to understand by how much should we increase the weight of our product so that the overall sales increase. Item\_MRP –which helps to understand how to place the cost so that sales might increase. Item\_Quantity – helps us to understand at what amount the item should be purchased so that sales will increase and they can minimize the purchase of odd items.By using our application, a user can learn how to purchase different products to get raise in sales.

#### **Help them to solve problems related to problem statement:**

The main issue we are trying to resolve in our model is that how can we improve the sales by taking into consideration of different attributes, that might help in improving our sales this in turn is the issue faced by all the retailer who can learn through this model by changing the product parameters. we can also develop web application by making user to customize his own attributes.

#### **Other Ideas to extend the project:**

In our model we have taken a smaller dataset for creating a robust inference because number of observations was only 8494. Gathering more data can yield more robust predictions. For instance, there are some attributes which may improve the model such as Outlet\_Location, Outlet\_Establishment\_year . we can gather more data from all kinds of stores so this will help us to understand to add some more different attributes which might be effective . we can make the stores to gather all the data for every 6 months and gathering the data from different kinds of stores and different kinds of locations will help to understand what product is sold most and which product has less sales .This will also helps the new retailer who wants to enter industry and analyse what kind of items are being sold at their respective locations. Gathering huge data will also help other retailer owner to recognize what items are getting more sales. we can also improve it by getting data of each unique item.this will also help users to predict sales of each unique different item for example if we have the data for all different kinds of noodles sold so that when the user specifically want to predict the sales of one noodle item then it will be really helpful to know the sales of that noodle item.

