murali Krishna chintha

Backorder prediction

Table of Contents

[1. Introduction 2](#_Toc42644595)

[1.1. Data 2](#_Toc42644596)

[1.2. Technologies Used: 3](#_Toc42644597)

[2. Process Flow 4](#_Toc42644598)

[2.1. Retraining the model 5](#_Toc42644599)

[2.1.1. Data Validation Check 5](#_Toc42644600)

[2.1.2. Data Splitting 6](#_Toc42644601)

[2.1.3. Data Pre-processing 6](#_Toc42644602)

[2.1.4. Training Different Models 8](#_Toc42644603)

[2.1.5. Choosing the best model based on the metric. 8](#_Toc42644604)

[2.1.6. Saving the model. 8](#_Toc42644605)

[2.2. Bulk Prediction 9](#_Toc42644606)

[2.2.1. Data Validation Check 10](#_Toc42644607)

[2.2.2. Data Pre-processing 10](#_Toc42644608)

[2.2.3. Loading the model and predicting using the model 11](#_Toc42644609)

[2.3. Single Value Prediction 11](#_Toc42644610)

[2.3.1. Data Validation Check 13](#_Toc42644611)

[2.3.2. Loading the model and predicting the data 13](#_Toc42644612)

[3. Concepts 13](#_Toc42644613)

[3.1. Random Forest Classifier 13](#_Toc42644614)

[3.2. XGBoost Classifier 14](#_Toc42644615)

[3.3. GridSearchCV 14](#_Toc42644616)

# Introduction

Backorder is a common supply chain problem, impacting an inventory system service level and effectiveness. Identifying parts with the highest chances of shortage prior its occurrence can present a high opportunity to improve an overall company's performance. **Backorder management** forms an important subset of the inventory management analysis. The percentage of items backordered and the number of backorder days are important measures of the quality of a company's customer service and the effectiveness of its inventory management. A company that consistently sees items in backorder, might be running too lean. Such a company might be losing out on business by not providing the products demanded by its customers. This is because when an item is on backorder, a customer might look somewhere else for a substitute product, especially if the expected waiting time until the product becomes available is considerably long. This can provide an opportunity for once loyal customers to try other companies' products and potentially switch. Thus, it is a huge disadvantage in a highly competitive industry, where the number of alternatives available are huge or in an industry where brand loyalty is very low. Difficulties in proper inventory management leading to enormous back orders, can eventually lead to loss of market share as customers become frustrated with the company's lack of product availability. Moreover, customers waiting for their backordered products must receive shipping notifications on the shipping date they’ve been provided. This means that customers must be kept in the loop in case of any delays or changes, to avoid a storm of complaints and help maintain a good reputation for the company.

## Data

The dataset has 23 columns including 22 features and one target column.

|  |  |
| --- | --- |
| **Features** | **Description** |
| **sku** | Random ID for the product |
| **national\_inv** | Current inventory level for the part |
| **lead\_time** | Transit time for product (if available) |
| **in\_transit\_qty** | Amount of product in transit from source |
| **forecast\_3\_month** | Forecast sales for the next 3 months |
| **forecast\_6\_month** | Forecast sales for the next 6 months |
| **forecast\_9\_month** | Forecast sales for the next 9 months |
| **sales\_1\_month** | Sales quantity for the prior 1 month time period |
| **sales\_3\_month** | Sales quantity for the prior 3 month time period |
| **sales\_6\_month** | Sales quantity for the prior 6 month time period |
| **sales\_9\_month** | Sales quantity for the prior 9 month time period |
| **min\_bank** | Minimum recommend amount to stock |
| **potential\_issue** | Source issue for part identified |
| **pieces\_past\_due** | Parts overdue from source |
| **perf\_6\_month\_avg** | Source performance for prior 6 month period |
| **perf\_12\_month\_avg** | Source performance for prior 12 month period |
| **local\_bo\_qty** | Amount of stock orders overdue |
| **deck\_risk** | Part risk flag |
| **oe\_constraint** | Part risk flag |
| **ppap\_risk** | Part risk flag |
| **stop\_auto\_buy** | Part risk flag |
| **rev\_stop** | Part risk flag |
| **went\_on\_backorder** | Product actually went on backorder. This is the target value. |

## Technologies Used:

|  |  |
| --- | --- |
| IDE | PyCharm |
| Database | MySQL |
| Frontend | HTML5, CSS3, Bootstrap |
| Integration | Flask |
| Deployment | Google Cloud Platform |

# Process Flow

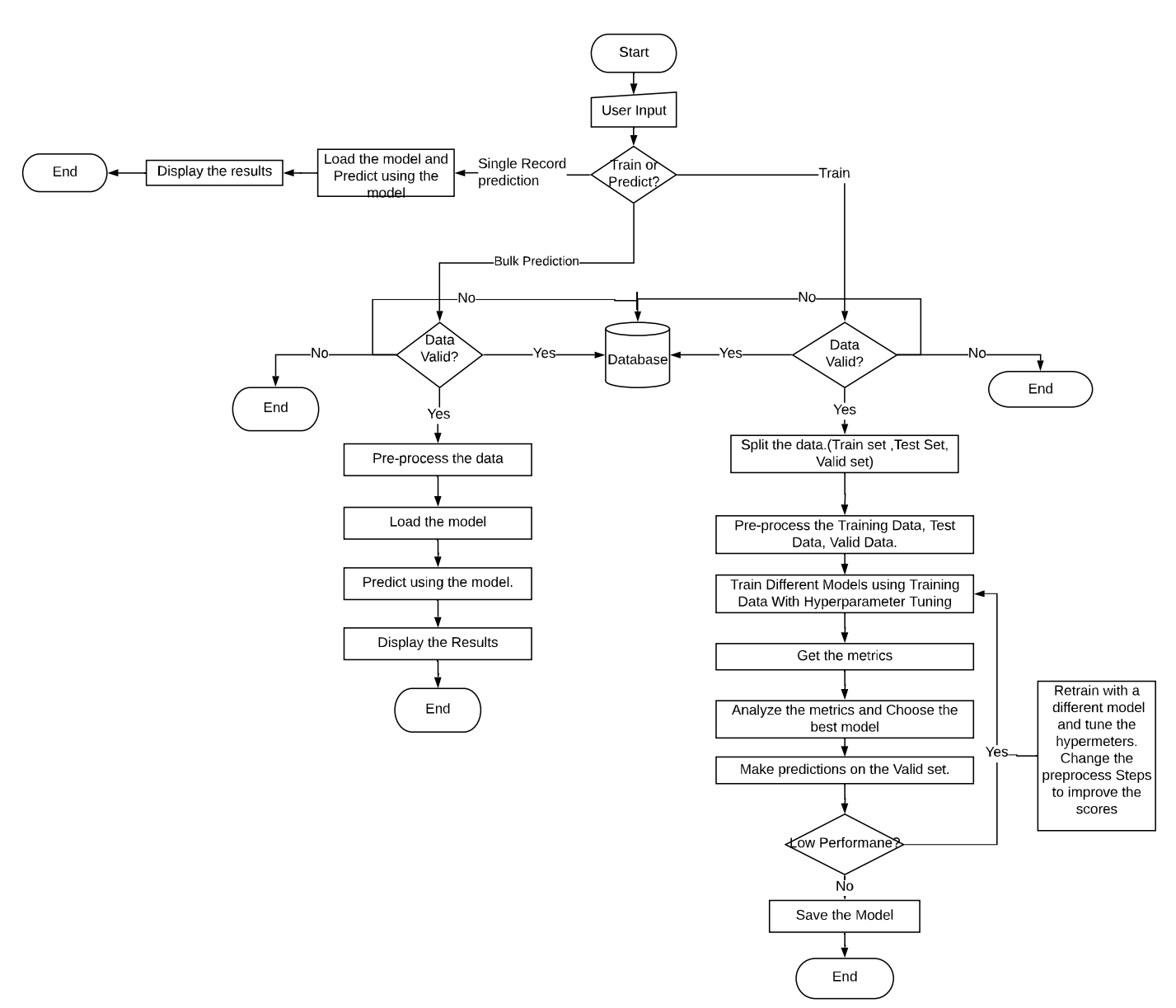


Fig 1: Flow chart explaining the process.

The Flow chart above explains the process flow of the entire solution. There are three paths to the flow based on the user input –

1. Retraining the model
2. Bulk Prediction
3. Single Value Prediction.

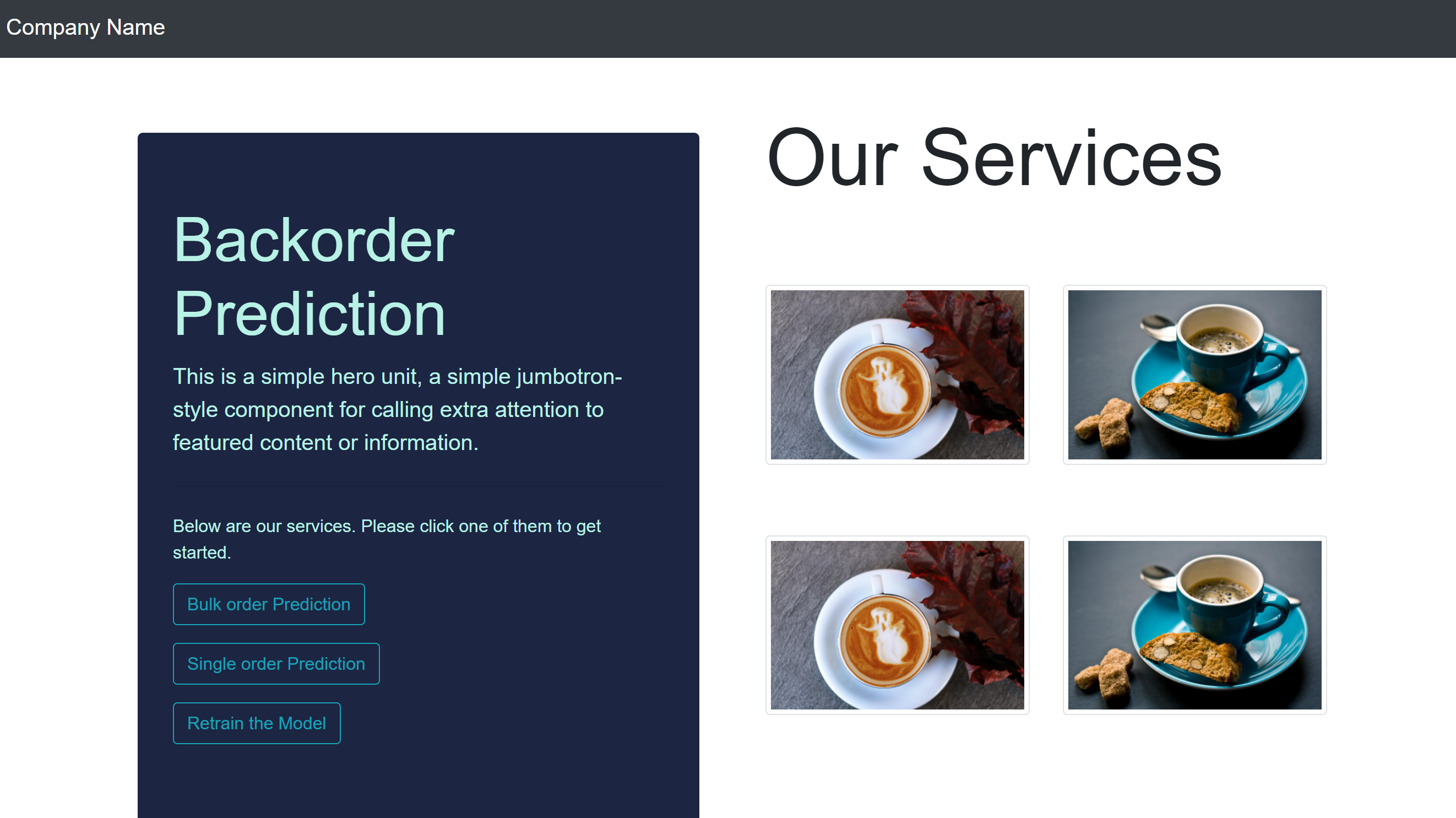


Fig 2: Home Page for user input.

## Retraining the model

The steps involved in retraining the model are

1. Data Validation Check
2. Data Splitting
3. Data Pre-processing
4. Training Different Models
5. Choosing the best model based on the metric.
6. Saving the model.

### Data Validation Check

Data Validation is performed to check if the data provided is valid or not. The main tasks in data validation check are –

1. Checking the number of columns agreed as per SLA
2. Checking the datatypes of each column agreed as per SLA.
3. Checking the column names agreed as per SLA.
4. Checking if any of the columns have more than 75% null values. In this case, data will be considered as invalid.

If the data meets all the four conditions then the data is considered to be Valid data. If it does not meet the conditions, data will be considered as invalid data.

Both Valid and Invalid data are pushed into the database.

### Data Splitting

The data for retraining the model will be split into three unequal sets in the ratio 70:15:15.

1. Training set.
2. Validation set.
3. Test set.

Training and Validation sets will be used for training different models and choosing the best one among them. Test set will be used to validate the chosen model.

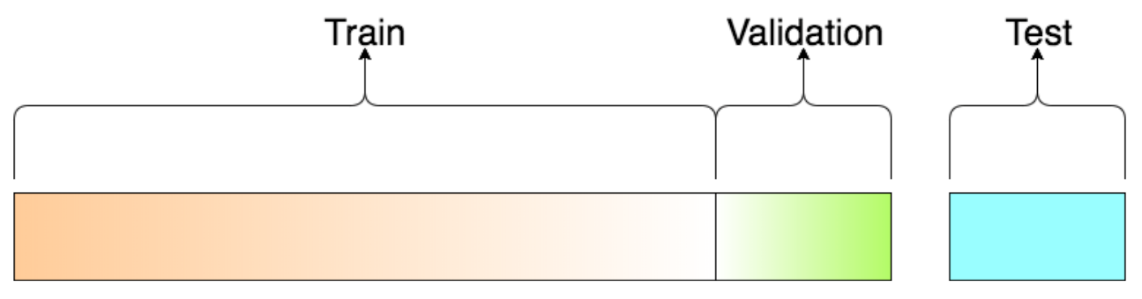


Fig 3: Diagram of Train Validation Test splitting of the data.

### Data Pre-processing

Data pre-processing is the most important step in training the model. In this step, we will prepare the data to be fed the model for training. Pre-processing includes –

1. Dealing with null values – Null values in columns ‘perf\_6\_month’ and ‘perf\_12\_month’ have been replaced with median. Since the null value count in rest of the columns is very less. Dropping the null values does not affect the distribution of the data. So, the null values in other columns have been dropped.

Note: If the null values in any column are greater than 50% of the total data. The data will be considered as invalid.

1. Converting categorical variables – Categorical columns are converted into numerical columns. In this dataset the categorical variables are
2. 'potential\_issue',
3. 'deck\_risk',
4. 'oe\_constraint',
5. 'ppap\_risk',
6. 'stop\_auto\_buy',
7. 'rev\_stop',
8. 'went\_on\_backorder'

Since all the categorical columns are binary columns i.e., there are only two unique values in all the categorical columns. Map function has been used to map

‘Yes’ – 1

‘No’ - 0

1. Feature Extraction – Only features that explains the variance in the dataset are used. All the other columns will be dropped. These columns have been considered to drop after checking the correlation between the features. The dropped columns in this data are

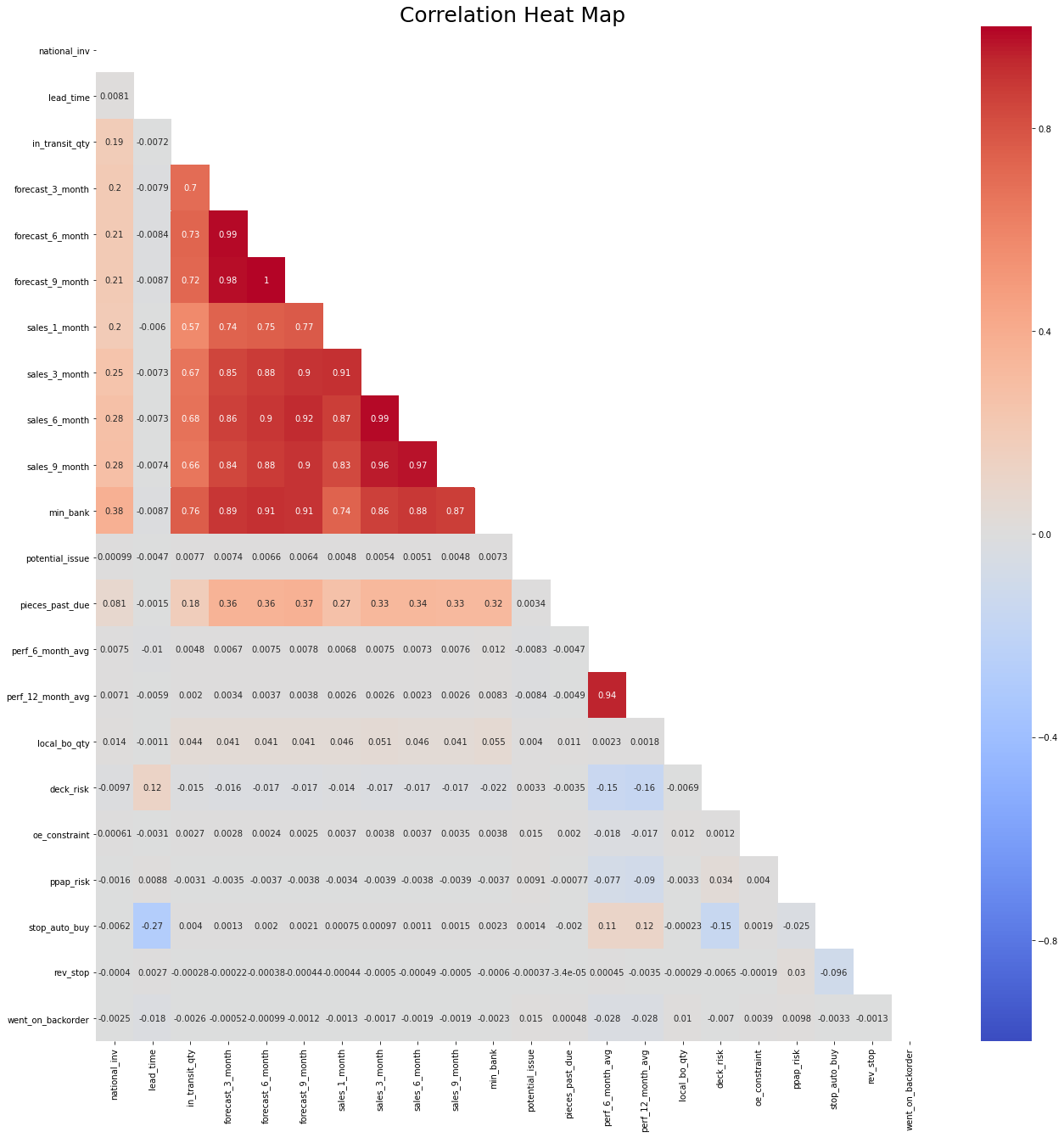


Fig 4: Correlation Matrix

* 1. 'in\_transit\_qty',
  2. 'forecast\_3\_month',
  3. 'forecast\_6\_month',
  4. 'forecast\_9\_month',
  5. 'sales\_3\_month',
  6. 'sales\_6\_month',
  7. 'sales\_9\_month',
  8. 'min\_bank',
  9. 'potential\_issue',
  10. 'perf\_12\_month\_avg'

1. Data Standardization – Since most of the features have different units and are measured in different scales, data standardization should be done to bring all the features to a same scale.

### Training Different Models

After pre-processing, the imbalance in the data should be handled by upsampling the minority class and downsampling the majority class in the data. After imbalance in the data is handled, data can be used to train different models. The different models used are

1. Random Forest Classifier
2. XGBoost Classifier

GridSearchCV has been used for hyperparameter optimization of the models. Each model will be trained on the pre-processed data using GridSearchCV.

### Choosing the best model based on the metric.

After training different and tuning the respective hyperparameters, best model will chose based on the best f1-score. The metric used here is the F1-Score. The model with the highest F1-score will be chosen.

### Saving the model.

After the model has been chosen. The existing model will be deleted and the new model will be dumped into a pickle file which can be used for loading the model any number of times.

## Bulk Prediction



Fig 4: Page for Bulk predictions

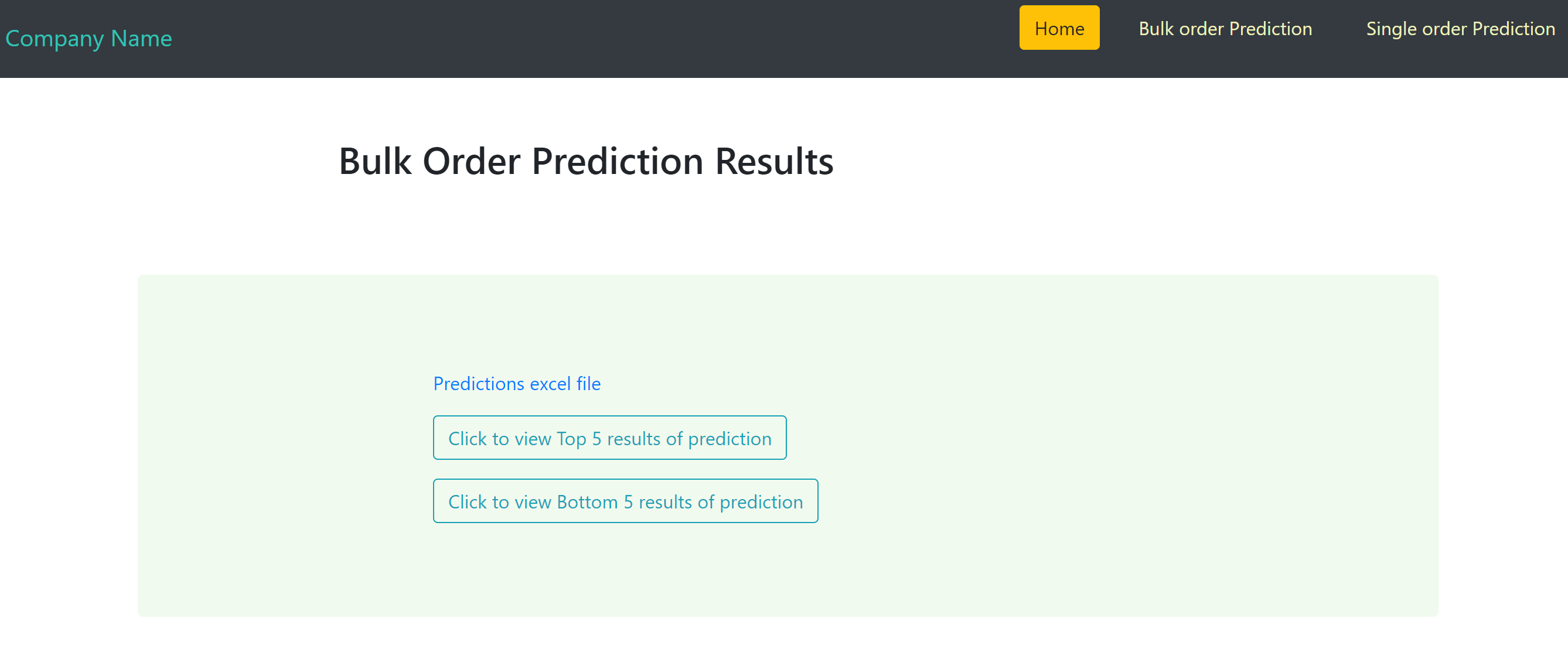


Fig 5: Bulk prediction results page

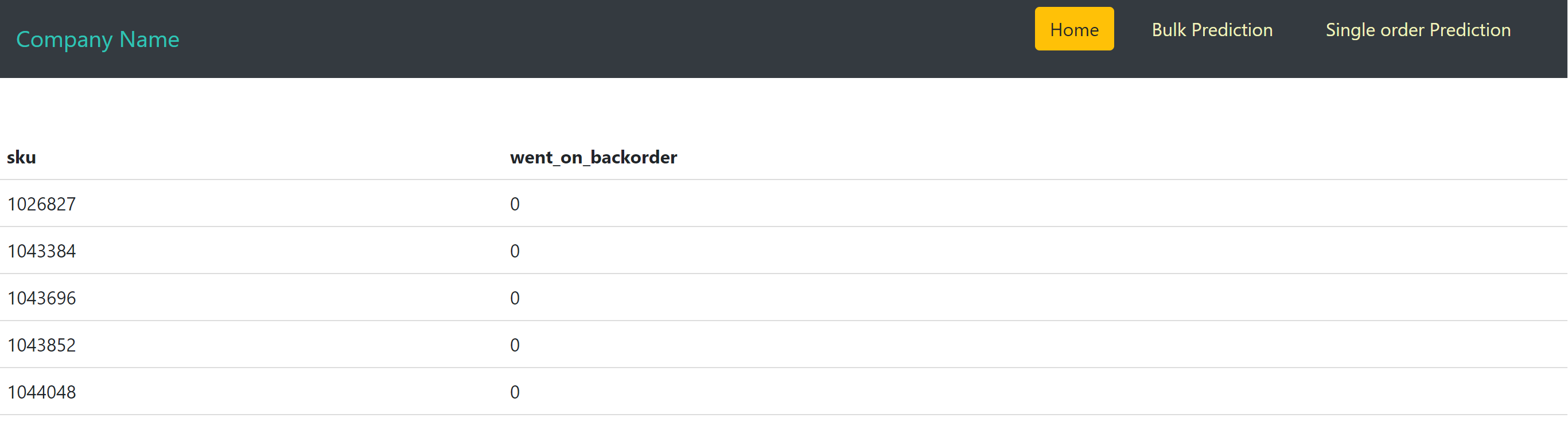


Fig 6: Page showing the top 5 rows of the bulk prediction.

The steps involved in predicting the model are

1. Data Validation Check
2. Data Pre-processing
3. Loading and Predicting using the model

### Data Validation Check

Data Validation is performed to check if the data provided is valid or not. The main tasks in data validation check are –

1. Checking the number of columns agreed as per SLA
2. Checking the datatypes of each column agreed as per SLA.
3. Checking the column names agreed as per SLA.
4. Checking if any of the columns have more than 75% null values. In this data will considered as invalid.

If the data meets all the three conditions then the data is considered to be Valid data. If it does not meet the conditions, data will be considered as invalid data.

### Data Pre-processing

Data pre-processing is the most important step in training the model. In this step, we will prepare the data to be fed the model for training. Pre-processing includes –

1. Dealing with null values – Null values in columns ‘perf\_6\_month’ and ‘perf\_12\_month’ have been replaced with median. Since the null value count in rest of the columns is very less. Dropping the null values does not affect the distribution of the data. So, the null values in other columns have been dropped.

Note: If the null values in any column are greater than 50% of the total data. The data will be considered as invalid.

1. Converting categorical variables – Categorical columns are converted into numerical columns. In this dataset the categorical variables are
2. 'potential\_issue',
3. 'deck\_risk',
4. 'oe\_constraint',
5. 'ppap\_risk',
6. 'stop\_auto\_buy',
7. 'rev\_stop',
8. 'went\_on\_backorder'

Since all the categorical columns are binary columns i.e., there are only two unique values in all the categorical columns. Map function has been used to map

‘Yes’ – 1

‘No’ - 0

1. Feature Extraction – Only features that explains the variance in the dataset are used. All the other columns will be dropped. These columns have been considered to drop after checking the correlation between the features. The dropped columns in this data are

* 1. 'in\_transit\_qty',
  2. 'forecast\_3\_month',
  3. 'forecast\_6\_month',
  4. 'forecast\_9\_month',
  5. 'sales\_3\_month',
  6. 'sales\_6\_month',
  7. 'sales\_9\_month',
  8. 'min\_bank',
  9. 'potential\_issue',
  10. 'perf\_12\_month\_avg'

1. Data Standardization – Since most of the features have different units and are measured in different scales, data standardization should be done to bring all the features to a same scale.

### Loading the model and predicting using the model

After pre-processing the data for prediction, the saved model will be loaded from the pickle file and the prediction data will be given to the model as input. The output can be downloaded.

## Single Value Prediction

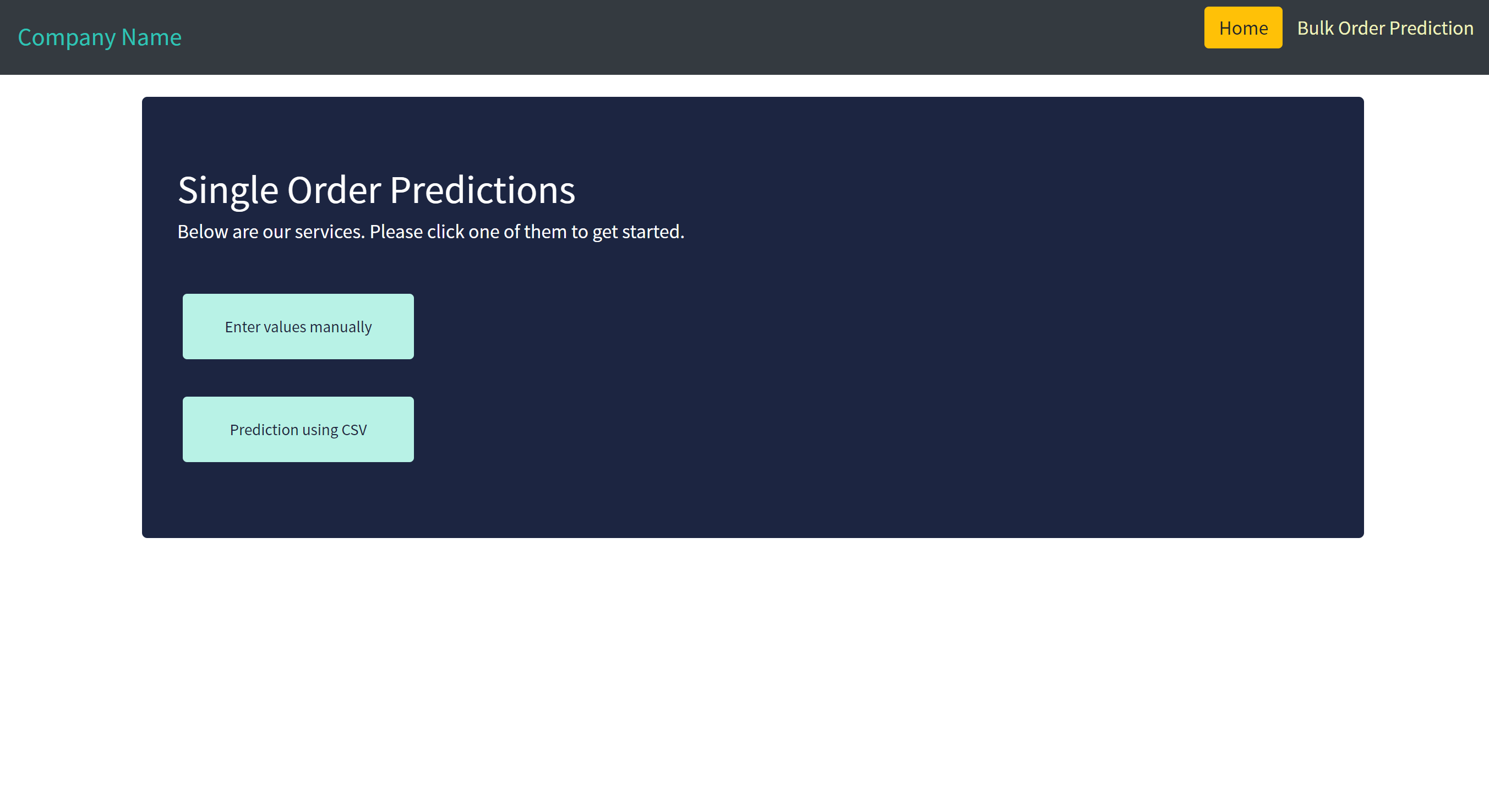


Fig 4: Single Value Prediction page.

The user can choose to enter the values manually or can upload the CSV file containing a single record.

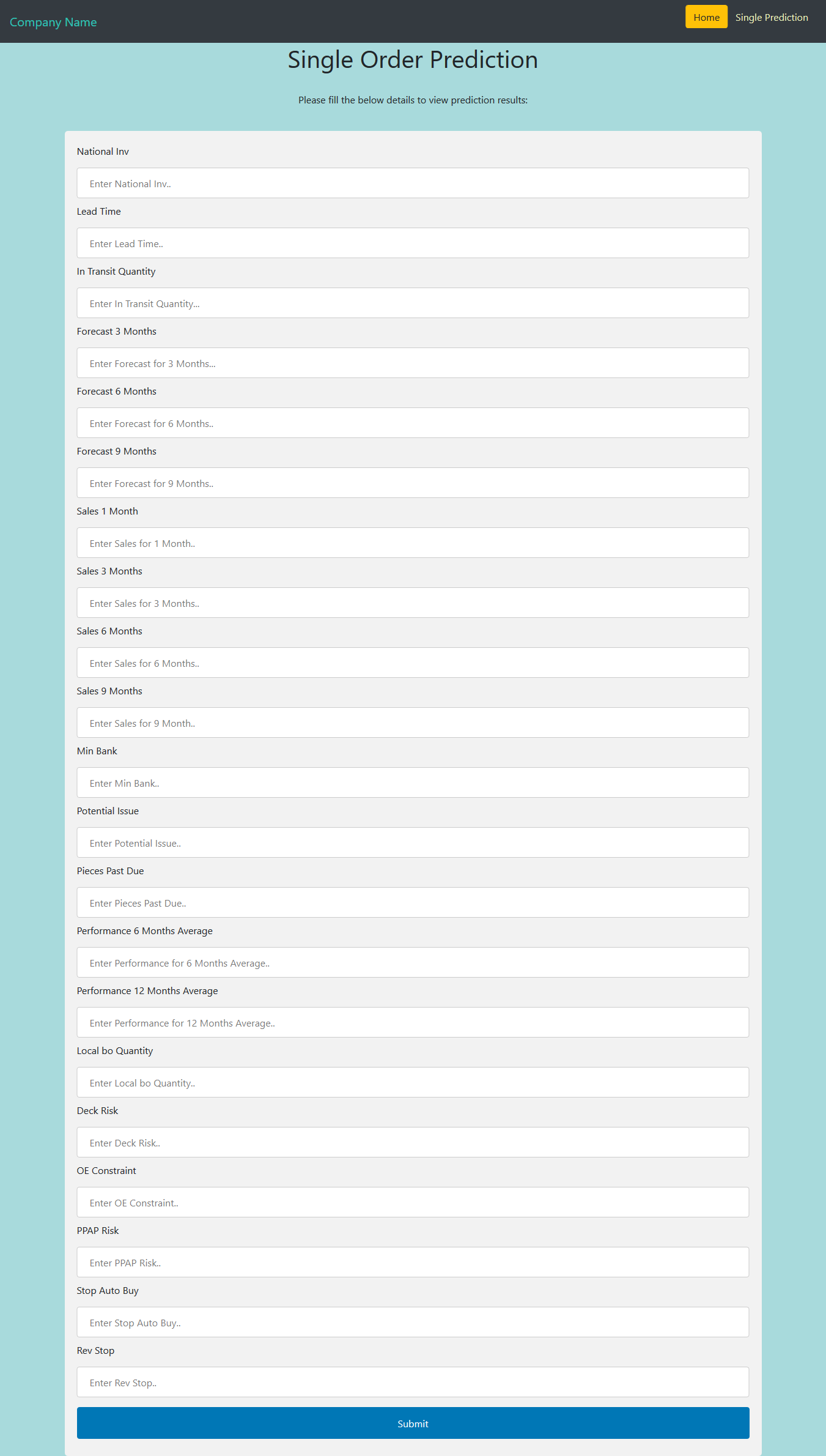


Fig: Page for entering values manually

The Steps for prediction if the user chose to enter the values manually are

1. Data Validation Check.
2. Loading the model and predicting the data.

### Data Validation Check

Data Validation is performed to check if the data provided is valid or not. The main tasks in data validation check are –

1. Checking the data types of each of the variables entered by the user.
2. Checking if any of the entered values are null values.

If the data meets all the conditions then the data is considered to be Valid data. If it does not meet the conditions, data will be considered as invalid data.

### Loading the model and predicting the data

After validating the data for single value prediction, the saved model will be loaded from the pickle file and the prediction data will be given to the model as input. The output can be downloaded.

\*\*\*\* Add a pic after the results of single value prediction\*\*\*\*

# Concepts

## Random Forest Classifier

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. In random forests, each tree in the ensemble is built from a sample drawn with replacement from the training set. Also, when splitting a node during the construction of the tree, the split that is chosen is no longer the best split among all features. Instead, the split that is picked is the best split among a random subset of the features. As a result of this randomness, the bias of the forest usually slightly increases but, due to averaging, its variance also decreases, usually more than compensating for the increase in bias, hence yielding an overall better model. In our model, the number of estimators used are 5 and we have considered ‘Entropy’ as a measure of the quality of a split.

## XGBoost Classifier

Extreme Gradient Boosting (XGBoost) is built on the principles of gradient boosting framework. 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. XGBoost uses a more regularized model formalization to control over-fitting, which gives it better performance. In our model, the number of estimators used are 100. The model internally uses log-linear classifier for regularizing the model with λ = 1.

## GridSearchCV

GridSearchCV is a library function that is a member of sklearn’s model\_selection package. It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, you can select the best parameters from the listed hyperparameters.

In addition to that, you can specify the number of times for the cross-validation for each set of hyperparameters.