Logistics Demand Forecasting using Artificial Neural Networks

Harsh Sunil Langade

20th January 2022

Abstract

Businesses face different inventory challenges when they are dealing with supply chains. One such challenge is Demand Forecasting. Logistics demand forecasting is a way for companies to accurately anticipate the demand for products and shipments throughout the supply chain, even under uncontrollable conditions or circumstances. It is a fundamental instrument to evaluate and optimize the supply chain.

AI-Enabled demand forecasting boosts logistics with tools that provide real action plans and support decision-making based on broad data analytics. Artificial Intelligence can compare the planning decisions made by humans and algorithms and use this data for. In some way, it will improve the demand forecasting and help to make accurate decisions during the logistic process.

In this report, I have proposed an idea of using Artificial Intelligence (specifically deep learning) techniques for Logistics Demand Forecasting which may result in more accurate demand prediction as compared to various traditional statistical methods.

1.Problem Statement

To deploy Artificial Intelligence (Machine Learning) techniques to improve Demand Forecasting results/predictions for small Logistics service providers. Supply chain experts have relied on the traditional forecasting models and tools to derive demand forecast for their goods and services. While it has helped them reduce the demand-supply mismatch to some extent, there's lot of catching up to do to improve the forecasting models.

Use of Artificial Intelligence (Machine Learning) techniques and models may substantially increase the chances of predicting demand with smaller margin of error.

2.Business Need Assessment

The statistical forecast algorithms used generally, work on best fit model selection which doesn't work well for the high volume/volatility Stock-keeping units (SKU's). These algorithms don't have external variables like PMI index, inflation, social media sentiment, competitor pricing, weather forecast, demand sensing, to name a few factored into their logic. Failure to accurately predict demand forecast leads to higher inventory costs, lost sales, lower customer satisfaction levels, reduced margins and lot of other similar repercussions.

AI-Enabled demand forecasting boosts logistics with tools that provide real action plans and support decision-making based on broad data analytics. Artificial intelligence generates advanced forecasts based on real-time data based on internal and external influences such as demographics, weather, the performance of similar products, and online and social media reviews, allowing supply chain managers to take on more strategic tasks.

It is evident that development of highly efficient AI-based models for demand forecasting has become one of the most important needs for small-scale logistics service provider companies.

3.Target Specifications and Characterization

The main target, of developing and deploying efficient Artificial Intelligence based predictive models, is to replace or assist various statistical and survey-based models that are currently used to forecast demand of products by various companies that provide Logistics service.

To increase the accuracy of predictions of product demand which may further benefit in the following ways:

1.Reduce production and inventory risks with accurate demand prediction.

2.Reduce cargo charges by optimizing number of goods carrier vehicles which further can be done by accurate demand forecasting.

3.It will help to save time for the employees. In case the employees have more time and ready decision in planning the logistic activities, they can be more efficient in other departments of logistics companies.

4.External Search (information sources/references)

Some of the Information sources used are:

Importance of Demand Forecasting in Supply Chain

https://www.demandsolutions.com/ds-blog/abstracts/importance-of-demand-forecasting-in-supply-chain/

The Importance of Forecasting in Logistics Planning

https://blog.flexis.com/the-importance-of-forecasting-in-logistics-planning

What is Demand Forecasting? Importance and Benefits of Forecasting Customer Demand

hipbob.com/blog/demand-forecasting/

AI in Logistics for Demand Forecasting: Forecasting Benefits & Importance

https://datenwissen.com/blog/ai-in-logistics-for-demand-prediction/

References:

https://redstagfulfillment.com/what-is-demand-forecasting/

https://www.economicsdiscussion.net/demand-forecasting/techniques-of-demand-forecasting-survey-and-statistical-methods/3611

https://www.birlasoft.com/articles/how-machine-learning-is-reshaping-the-future-of-demand-forecasting

https://amconsoft.com/logistics-demand-forecasting-the-benefits-of-ai-how-to-implement-it/

https://www.transmetrics.ai/blog/logistics-demand-forecasting/

https://research.aimultiple.com/demand-forecasting/

5.Benchmarking:

Various demand forecasting methods already exist which are used extensively for this purpose. Some of them are:

SURVEY METHODS:

1.Delphi Method:

The Delphi method, or Delphi technique, leverages expert opinions on the market forecast. This method requires engaging outside experts and a skilled facilitator.It starts with sending a questionnaire to a group of demand forecasting experts. You create a summary of the responses from the first round and share it with the panel. This process is repeated through successive rounds. The answers from each round, shared anonymously, influence the next set of responses. The Delphi method is complete when the group comes to a consensus.

2.Market Experimentation:

Involves collecting necessary information regarding the current and future demand for a product. This method carries out the studies and experiments on consumer behaviour under actual market conditions. In this method, some areas of markets are selected with similar features, such as population, income levels, cultural background, and tastes of consumers. The market experiments are carried out with the help of changing prices and expenditure, so that the resultant changes in the demand are recorded. These results help in forecasting future demand.

STATISTICAL METHODS:

1.Econometric Method:

The econometric method requires some number crunching. This technique combines sales data with information on outside forces that affect demand. Then a mathematical formula is created to predict future demand. The econometric demand forecasting method accounts for relationships between economic factors.

2.Trend Projection Method:

In this method, sales forecasts are made through analysis of past data taken from previous year's books of accounts. In case of new organizations, sales data is taken from organizations already existing in the same industry. This method uses time-series data on sales for forecasting the demand of a product.

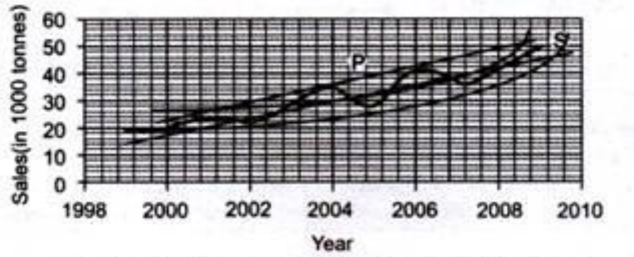


Figure-13: An Example of Graphical Method

Table-1: Time Series Data on Sales of XYZ Organization	
Year	**Sales (In 1000 tones)**
2000	20
2001	24
2002	22
2003	30
2004	36
2005	28

Table-1: Time Series Data on Sales of XYZ Organization	
Year	**Sales (In 1000 tones)**
2006	40
2007	36
2008	42
2009	50

Some other statistical methods are: Time Series Analysis, Index Number method.

6.Applicable Patents:

1.WO2014075108A2   WIPO (PCT)

Forecasting system using machine learning and ensemble methods

Techniques for determining forecast information for a resource using learning algorithms are disclosed. The techniques can include an ensemble of machine learning algorithms. The techniques can also use latent states to generate training data. The techniques can identify actions for managing the resource based on the forecast information.

2.U.S. Pat. No. 7,080,026. It discloses systems and methods for demand forecasting that enable multiple-scenario comparisons and analyses by letting users create forecasts from multiple history streams (for example, shipments data, point-of-sale data, customer order data, return data, etc.) with various alternative forecast algorithm theories.

7.Applicable Regulations:

1. Government Regulations for small logistics service providers.
2. Various Employment Laws.
3. Laws related to data collection from other service providers, so as to have large amount of data which would help in making better prediction models.
4. Models need to be developed and optimised individually, else the patents like those mentioned above might claim the technology used.
5. Review of various work authority regulations.

8.Applicable Constraints:

1. Use of cloud storage platforms to store the data gathered from various sources.
2. Need of good GPUs for better performance of deep learning algorithms and neural networks.
3. Continual Data maintenance.
4. Data Collection – Neural networks perform better when large amount of data is available.
5. Making logistics service providers aware of, and proving the advantages of use of AI in demand forecasting. Convincing them to implement these techniques.
6. Good data management.
7. Cost factor – Large Neural Networks increase the total cost of the process.

9.Business Opportunity:

Many small logistics companies still use traditional statistical and survey-based methods to forecast demand.

In the logistics industry, the majority of business growth is dependent on demand and supply combination. The more accurately they predict the demand of a product the more growth they see in their business. AI is one of the great technologies through which demand prediction or forecasting can be made accurate and fast which leads to business growth and makes it easy to compete with competitors.

So, it is clear that there is a huge business opportunity as every company would want to grow its business and increase their overall revenue(profit).

10.Concept Generation:

The AI product requires machine learning and deep learning models to be created as per the needs of the product. Classical machine learning models are modified rather than building a new one from scratch. The product basically performs a regression task for predicting product demand. There are various machine learning models that can be used for this task. Neural Networks are proved to be better than classical machine learning algorithms like Linear Regression. Neural Networks perform better as they can deal with non-linearities with good performance unlike other algorithms.

Also, logistics service provider companies generate a large volume of data everyday and data can be taken from other sources as well. Neural Networks work much better than any other algorithm when amount of available training data is large.

11.Concept Development:

The goal is that logistics companies should get accurate demand forecast. Considering advantages of Neural Networks, the next step is to ideate the product. Using already available data on the demand of various products, the model will predict the demand for future. The data is fed to the model on which it trains. It is then validated using data for cross-validation and testing. Once the model is ready, it is used for demand forecasting in real world scenario.

Various steps involved in Concept/Product development are as follows:

1.Data Collection and Pre-processing:

Data can be collected from various sources. This data needs to be cleaned and properly pre-processed according the neural network model. Feature Engineering can be done to introduce new features which would provide useful information to the model. Filling any missing values is very important. The model works best when properly structured data is provided to it.

2.Data Analysis:

Available data needs to be analysed well. Using python libraries for analysis, a deep insight into the data can be taken.

3.Data Visualization:(If applicable)

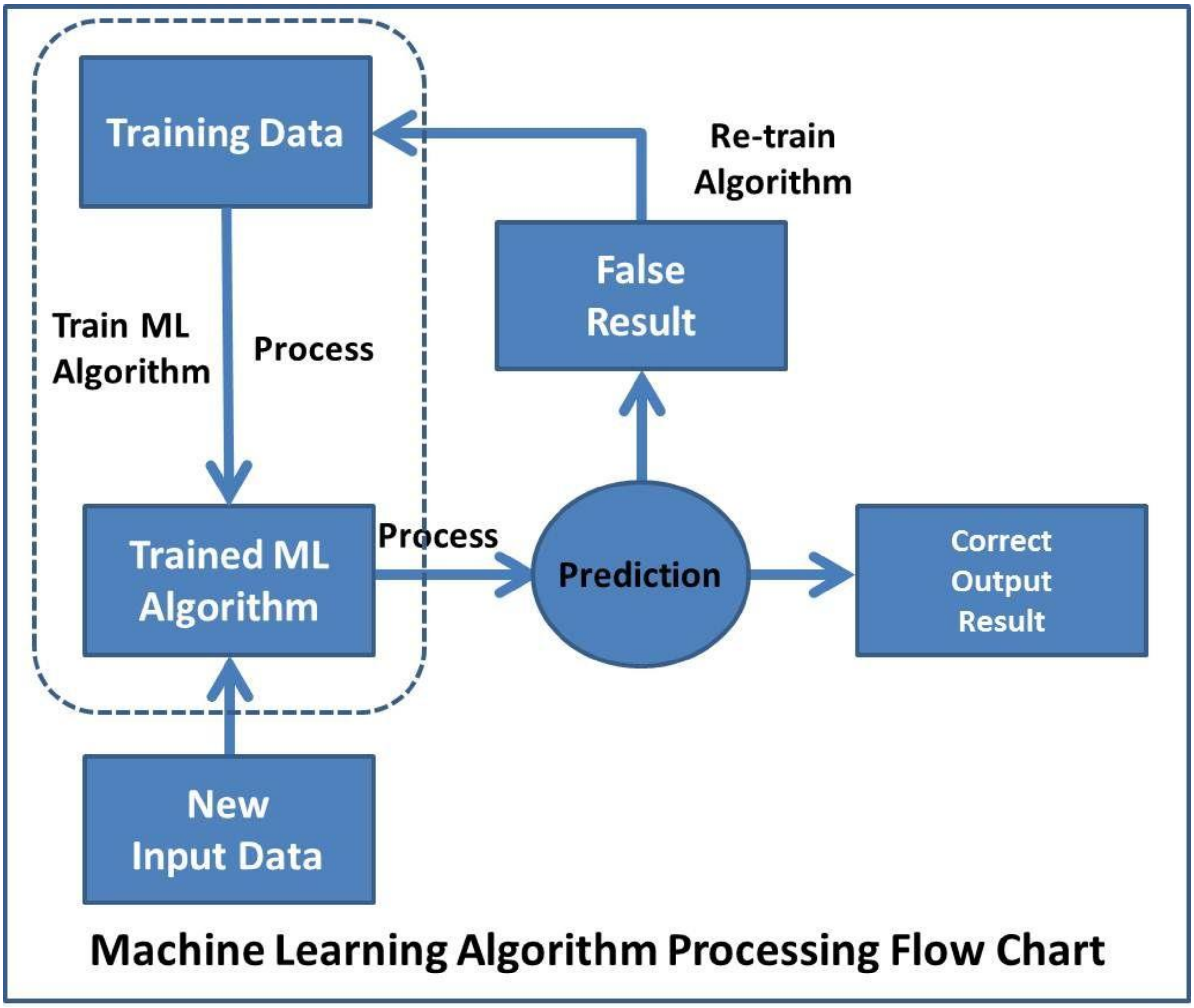
Using python libraries, seaborn and matplotlib, data visualization can be done for better understanding of data.

4.Model Building:

For the product development, Neural Networks are used. Hyperparameter tuning needs to be done well for better performance. This may include number of hidden layers, number of units in each layer, optimization algorithm, activation functions etc.

5.Model Evaluation:

Performance of the model is evaluated using various evaluation metrics.

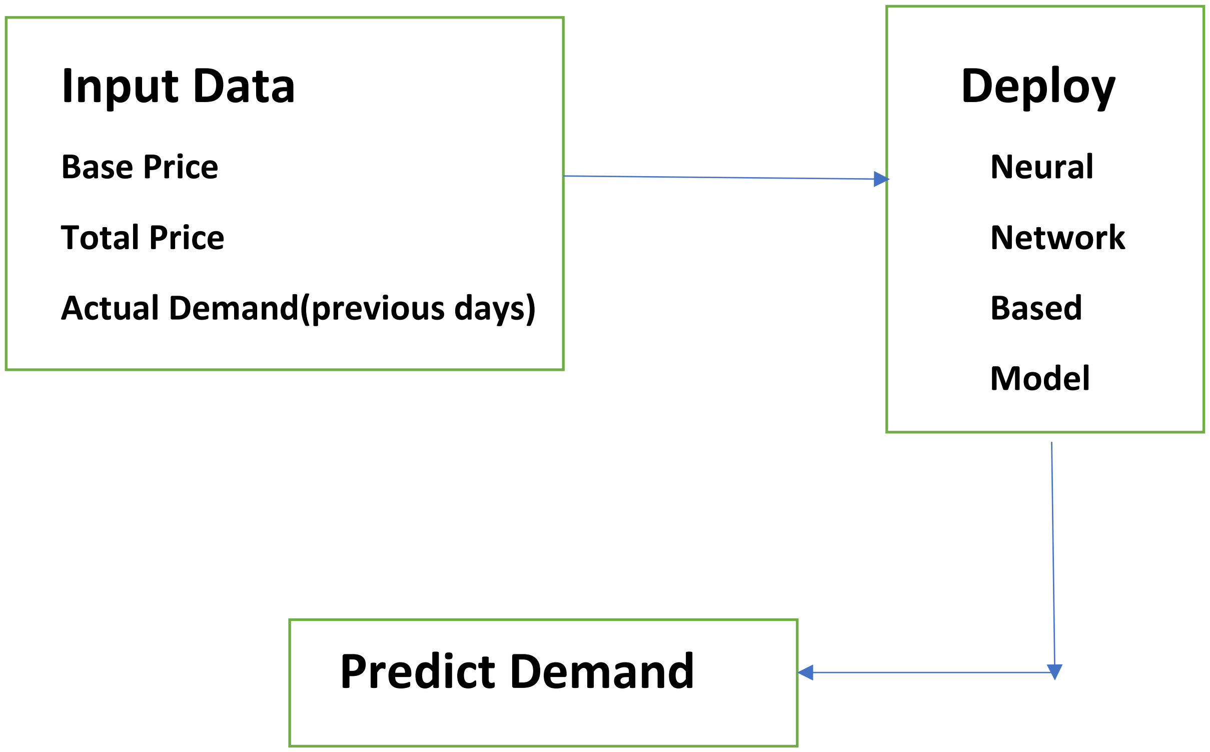


12.Final Product Prototype with Schematic Diagram:

The product is an Artificial Intelligence system that will be available as a software or an application for Logistics companies for demand forecasting.

The system will take input data like total price, base price, actual demand, date. It will ask for actual demand for specific number of days. The system is developed using Neural Network. It will predict the demand for a future.

A rough schematic is as follows:



13.Product Details:

13.1.How does it work?

The model takes input data including base price, total price, actual demand of previous specified number of days, date. Some of this data may not be useful in predicting the demand by regression technique. Using this data, the model predicts the demand for future. This prediction, combined with other methods can be used by logistics company for planning and optimizing various aspects of logistics services.

13.2.Data Sources:

The dataset is taken from a GitHub repository. Link to the dataset:

https://raw.githubusercontent.com/shreyas-jk/Product-Demand-Forecasting-Using-ML/main/data.csv

13.3Algorithms, Frameworks, Packages and Libraries, Software used:

1.Algorithms: Deep Neural Network, Adam-Optimization algorithm.

2.Frameworks: TensorFlow, Sci-Kit Learn

3.Packages and Libraries: NumPy, Pandas, Matplotlib, Seaborn

4.Sotware: Any Integrated Development Environment (IDE). Here, Jupyter Notebook is used.

13.4.Team required to develop:

1.Data Researcher.

2.Data Analyst/Data Scientist.

3.Machine Learning Engineer.

4.Software Developer.

5.Supply Chain Management Professional (excelling at Demand Forecasting).

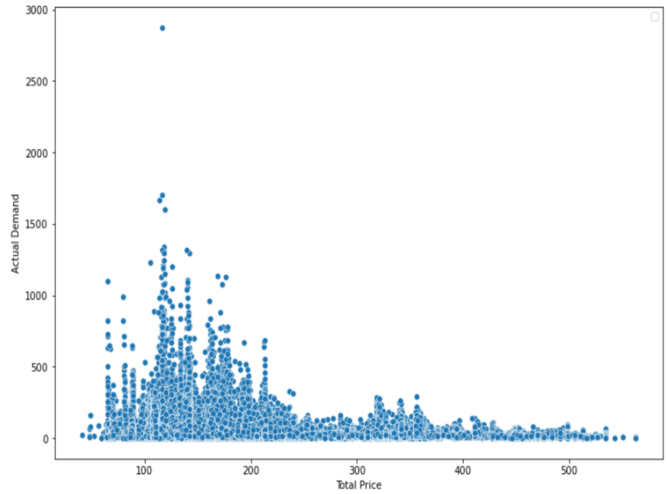
14.Code Implementation:

14.1Visualizations:

Data Visualization

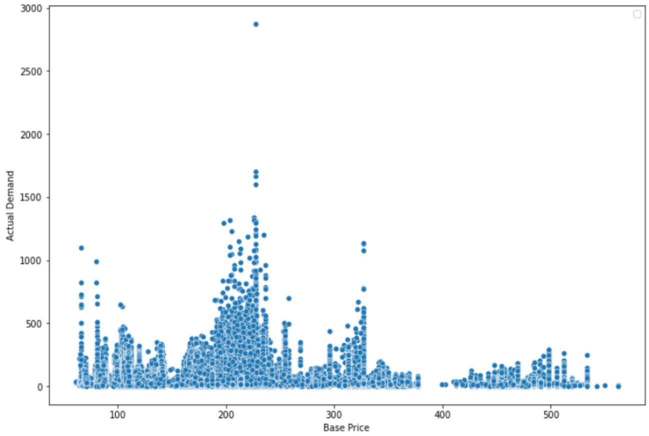
In [64]:
```
plt.figure(figsize=(12,8))
sns.scatterplot(x='total_price',y='demand',data=data)
plt.xlabel('Total Price')
plt.ylabel('Actual Demand')
plt.legend()
plt.show()
```

```
No handles with labels found to put in legend.
```



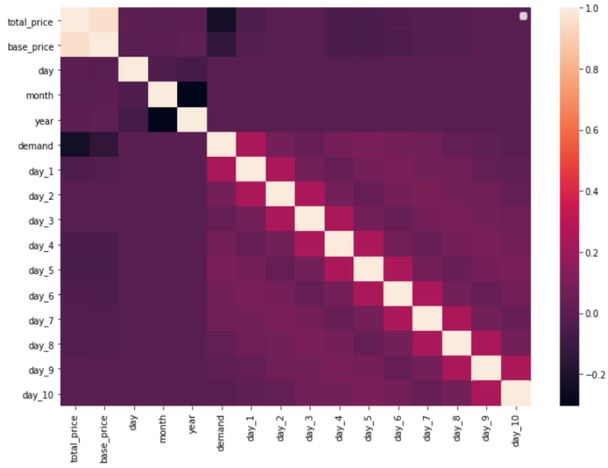
In [65]:
```
plt.figure(figsize=(12,8))
sns.scatterplot(x='base_price',y='demand',data=data)
plt.xlabel('Base Price')
plt.ylabel('Actual Demand')
plt.legend()
plt.show()
```

```
No handles with labels found to put in legend.
```



In [66]:
```
plt.figure(figsize=(12,8))
sns.heatmap(data=data.corr())
plt.legend()
plt.show()
```

```
No handles with labels found to put in legend.
```



14.2.Exploratory Data Analysis(EDA):

In [46]:
```
data = pd.read_csv('https://raw.githubusercontent.com/shreyas-jk/Product-Demand-Forecasting-Using-ML/main/data.csv')
```

Exploratory Data Analysis

In [47]:
```
data.head()
```

Out[47]:

	record_ID	week	store_id	sku_id	total_price	base_price	is_featured_sku	is_display_sku	units_sold
0	1	17/01/11	8091	216418	99.0375	111.8625	0	0	20
1	2	17/01/11	8091	216419	99.0375	99.0375	0	0	28
2	3	17/01/11	8091	216425	133.9500	133.9500	0	0	19
3	4	17/01/11	8091	216233	133.9500	133.9500	0	0	44
4	5	17/01/11	8091	217390	141.0750	141.0750	0	0	52

In [48]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150150 entries, 0 to 150149
Data columns (total 9 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   record_ID        150150 non-null  int64
 1   week             150150 non-null  object
 2   store_id         150150 non-null  int64
 3   sku_id           150150 non-null  int64
 4   total_price      150149 non-null  float64
 5   base_price       150150 non-null  float64
 6   is_featured_sku  150150 non-null  int64
 7   is_display_sku   150150 non-null  int64
 8   units_sold       150150 non-null  int64
dtypes: float64(2), int64(6), object(1)
memory usage: 10.3+ MB
```

In [49]:
```
data.describe()
```

Out[49]:

	record_ID	store_id	sku_id	total_price	base_price	is_featured_sku	is_display_sku	units_sold
count	150150.000000	150150.000000	150150.000000	150149.000000	150150.000000	150150.000000	150150.000000	150150.000000
mean	106271.555504	9199.422511	254761.132468	206.626751	219.425927	0.095611	0.133200	51.674206
std	61386.037861	615.591445	85547.306447	103.308516	110.961712	0.294058	0.339792	60.207904
min	1.000000	8023.000000	216233.000000	41.325000	61.275000	0.000000	0.000000	1.000000
25%	53111.250000	8562.000000	217217.000000	130.387500	133.237500	0.000000	0.000000	20.000000
50%	106226.500000	9371.000000	222087.000000	198.075000	205.912500	0.000000	0.000000	35.000000
75%	159452.750000	9731.000000	245338.000000	233.700000	234.412500	0.000000	0.000000	62.000000
max	212644.000000	9984.000000	679023.000000	562.162500	562.162500	1.000000	1.000000	2876.000000

In [50]:
```
data.isnull().sum()
```

Out[50]:
```
record_ID          0
week               0
store_id           0
sku_id             0
total_price        1
base_price         0
is_featured_sku    0
is_display_sku     0
units_sold         0
dtype: int64
```

In [51]:
```
null_index = data[data['total_price'].isnull()].index.tolist()
```

In [52]:
```
null_index
```

Out[52]:
```
[136949]
```

In [53]:
```
print(data.iloc[[136949]])
```

```
        record_ID      week  store_id  sku_id  total_price  base_price  \
136949     193915  23/04/13      9436  245338          NaN    469.5375

        is_featured_sku  is_display_sku  units_sold
136949                0               0           1
```

In [54]:
```
data.drop([136949],inplace=True)
```

In [55]:
```
data = data.drop(['record_ID','store_id','sku_id','is_featured_sku', 'is_display_sku'],axis=1)
```

In [56]:
```
data
```

Out[56]:

	week	total_price	base_price	units_sold
0	17/01/11	99.0375	111.8625	20
1	17/01/11	99.0375	99.0375	28
2	17/01/11	133.9500	133.9500	19
3	17/01/11	133.9500	133.9500	44
4	17/01/11	141.0750	141.0750	52
...	...	...	...	...
150145	09/07/13	235.8375	235.8375	38
150146	09/07/13	235.8375	235.8375	30
150147	09/07/13	357.6750	483.7875	31
150148	09/07/13	141.7875	191.6625	12
150149	09/07/13	234.4125	234.4125	15

In [57]:

```
data[["day", "month", "year"]] = data["week"].str.split("/", expand = True)
data = data.drop(['week'],axis=1)
```

In [58]:

```
data['day'] = pd.to_numeric(data['day'],errors='coerce')
data['month'] = pd.to_numeric(data['month'],errors='coerce')
data['year'] = pd.to_numeric(data['year'],errors='coerce')
data
```

Out[58]:

	total_price	base_price	units_sold	day	month	year
0	99.0375	111.8625	20	17	1	11
1	99.0375	99.0375	28	17	1	11
2	133.9500	133.9500	19	17	1	11
3	133.9500	133.9500	44	17	1	11
4	141.0750	141.0750	52	17	1	11
...	...	...	...	...	...	...
150145	235.8375	235.8375	38	9	7	13
150146	235.8375	235.8375	30	9	7	13
150147	357.6750	483.7875	31	9	7	13
150148	141.7875	191.6625	12	9	7	13
150149	234.4125	234.4125	15	9	7	13

150149 rows × 6 columns

In [59]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 150149 entries, 0 to 150149
Data columns (total 6 columns):
 #   Column       Non-Null Count   Dtype  
---  ------       --------------   -----  
 0   total_price  150149 non-null  float64
 1   base_price   150149 non-null  float64
 2   units_sold   150149 non-null  int64  
 3   day          150149 non-null  int64  
 4   month        150149 non-null  int64  
 5   year         150149 non-null  int64  
dtypes: float64(2), int64(4)
memory usage: 8.0 MB
```

In [60]:

```
data['demand'] = data['units_sold']
data = data.drop(['units_sold'],axis=1)
```

In [61]:

```
data['day_1'] = data['demand'].shift(-1)
data['day_2'] = data['demand'].shift(-2)
data['day_3'] = data['demand'].shift(-3)
data['day_4'] = data['demand'].shift(-4)
data['day_5'] = data['demand'].shift(-5)
data['day_6'] = data['demand'].shift(-6)
data['day_7'] = data['demand'].shift(-7)
data['day_8'] = data['demand'].shift(-8)
data['day_9'] = data['demand'].shift(-9)
data['day_10'] = data['demand'].shift(-10)
```

In [103]:

```
data
```

Out[103]:

	total_price	base_price	day	month	year	demand	day_1	day_2	day_3	day_4	day_5	day_6	day_7	day_8	day_9	day_10
0	99.0375	111.8625	17	1	11	20	28.0	19.0	44.0	52.0	18.0	47.0	50.0	82.0	99.0	120.0
1	99.0375	99.0375	17	1	11	28	19.0	44.0	52.0	18.0	47.0	50.0	82.0	99.0	120.0	40.0
2	133.9500	133.9500	17	1	11	19	44.0	52.0	18.0	47.0	50.0	82.0	99.0	120.0	40.0	68.0
3	133.9500	133.9500	17	1	11	44	52.0	18.0	47.0	50.0	82.0	99.0	120.0	40.0	68.0	87.0
4	141.0750	141.0750	17	1	11	52	18.0	47.0	50.0	82.0	99.0	120.0	40.0	68.0	87.0	186.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
150145	235.8375	235.8375	9	7	13	38	30.0	31.0	12.0	15.0	NaN	NaN	NaN	NaN	NaN	NaN
150146	235.8375	235.8375	9	7	13	30	31.0	12.0	15.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
150147	357.6750	483.7875	9	7	13	31	12.0	15.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
150148	141.7875	191.6625	9	7	13	12	15.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
150149	234.4125	234.4125	9	7	13	15	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

150149 rows × 16 columns

In [104]:

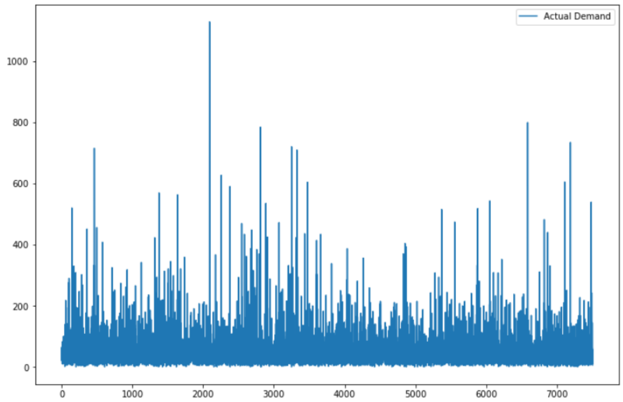
```
data = data.dropna()
```

14.3.ML Modelling:

Train -Test Split

In [162]:
```
from sklearn.model_selection import train_test_split
X_train, X_cv, y_train, y_cv = train_test_split( X, y, test_size=0.05, random_state=100)
```

In [163]:
```
plt.figure(figsize=(12,8))
plt.plot(y_cv, label='Actual Demand')
plt.legend()
plt.show()
```



In [164]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
```

In [165]:
```
scaler.fit_transform(X_train)
```

Out[165]:
```
array([[ 1.6357615 ,  2.057069  ,  1.3796641 , ..., -0.625589  ,
        -0.29330093, -0.6098536 ],
       [-0.35184568,  0.07123785,  1.3796641 , ...,  0.53604823,
         0.26902625, 12.853186  ],
       [ 0.03463349, -0.08300152, -1.0149249 , ...,  0.50285864,
         0.12017493, -0.0779805 ],
       ...,
       [-0.6417051 , -0.12798801, -1.6990932 , ...,  0.7185912 ,
        -0.04521541,  0.80293447],
       [-1.338748  , -1.2012364 ,  0.3534116 , ...,  0.17096226,
        -0.6240816 , -0.09460154],
       [-0.855649  , -0.7770784 , -1.242981  , ..., -0.07795999,
        -0.6571597 , -0.84254813]], dtype=float32)
```

In [166]:
```
scaler.fit_transform(X_cv)
```

Out[166]:
```
array([[-0.47898796, -0.37805066, -0.6583922 , ..., -0.6120595 ,
        -0.48163822, -0.6118959 ],
       [ 0.2632109 ,  0.12755254,  0.3735991 , ...,  0.09689186,
         1.5126175 ,  0.58382964],
       [ 0.28363836,  0.14651263, -0.4290608 , ..., -0.5276605 ,
         0.20728649, -0.26792005],
       ...,
       [-0.89434695, -0.9468543 ,  0.71759623, ..., -0.7977372 ,
        -0.84423023, -0.6118959 ],
       [ 0.14064585,  0.01379171, -1.5757178 , ..., -0.4939009 ,
        -0.11904628, -0.23516044],
       [ 0.692188  ,  0.52571493, -0.4290608 , ..., -0.2575838 ,
         0.5336192 , -0.23516044]], dtype=float32)
```

Neural Network Model

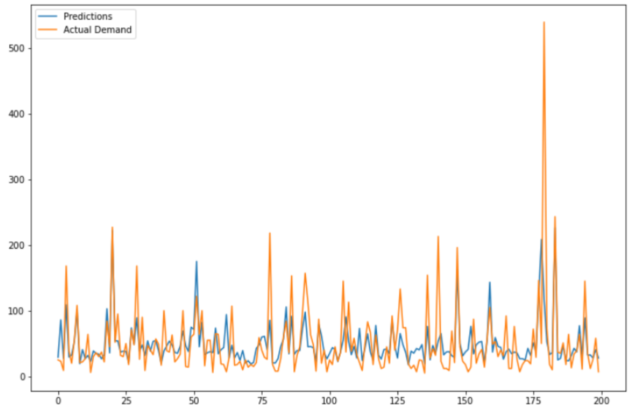
In [167]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Dropout
```

In [168]:
```
model = Sequential()

model.add(Dense(15,activation='relu'))
model.add(Dropout(0.15))
model.add(Dense(15,activation='relu'))
model.add(Dropout(0.15))
model.add(Dense(15,activation='relu'))
model.add(Dropout(0.15))
model.add(Dense(15,activation='relu'))
model.add(Dropout(0.15))
model.add(Dense(15,activation='relu'))



model.add(Dense(1))

model.compile(optimizer='adam',loss='mse')
```

Predictions

In [172]:
```
from sklearn.metrics import mean_squared_error,mean_absolute_error
predictions_train = model.predict(X_train)
predictions_cv = model.predict(X_cv)
```

In [173]:
```
plt.figure(figsize=(12,8))
plt.plot(predictions[-200:], label='Predictions')
plt.plot(y_cv[-200:], label='Actual Demand')
plt.legend()
plt.show()
```



Evaluation using Metrics

In [174]:
```
mae_train = mean_absolute_error(y_train,predictions_train)
mae_train
```
Out[174]: 26.528719

In [175]:
```
mse_train = mean_squared_error(y_train,predictions_train)
mse_train
```
Out[175]: 2432.0208

In [176]:
```
rmse_train = math.sqrt(mse_train)
rmse_train
```
Out[176]: 49.31552242401093

In [177]:
```
mae_cv = mean_absolute_error(y_cv,predictions_cv)
mae
```
Out[177]: 24.93765

In [178]:
```
mse_cv = mean_squared_error(y_cv,predictions_cv)
mse
```
Out[178]: 1609.4462

In [179]:
```
rmse_cv = math.sqrt(mse_cv)
rmse_cv
```
Out[179]: 47.718665581948905

14.4.Github Link to the code implementation:

Link is given below:

https://github.com/harshlangade19/demand_forecast

15.Conclusion

In the coming future, it is likely that most of the logistics companies will deploy Artificial Intelligence (Machine Learning) based models for demand forecasting.

Fortunately for demand planners, ML can now help further improve the forecast from 40% of actual to 70% of actual. There's a thumb rule that suggests that one can reduce planned inventory by 2.5% with 1% improvement in inventory forecast. AI-enabled demand forecasting is becoming increasingly important in the logistics industry. AI-enabled demand forecasting boosts logistics with its ability to predict demand and consumption patterns allowing logistics companies to plan for future needs and stock up on resources before they run out.

I have proposed the application of Artificial Intelligence technique in demand forecasting for small logistics companies.

The report is not an industry ready plan for deployment, but with further research and effort, it may become one. The product as described in this report does not consider demographics, weather, the performance of similar products, and online and social media reviews etc. Inclusion of this data will surely make the model more accurate in predicting demand.