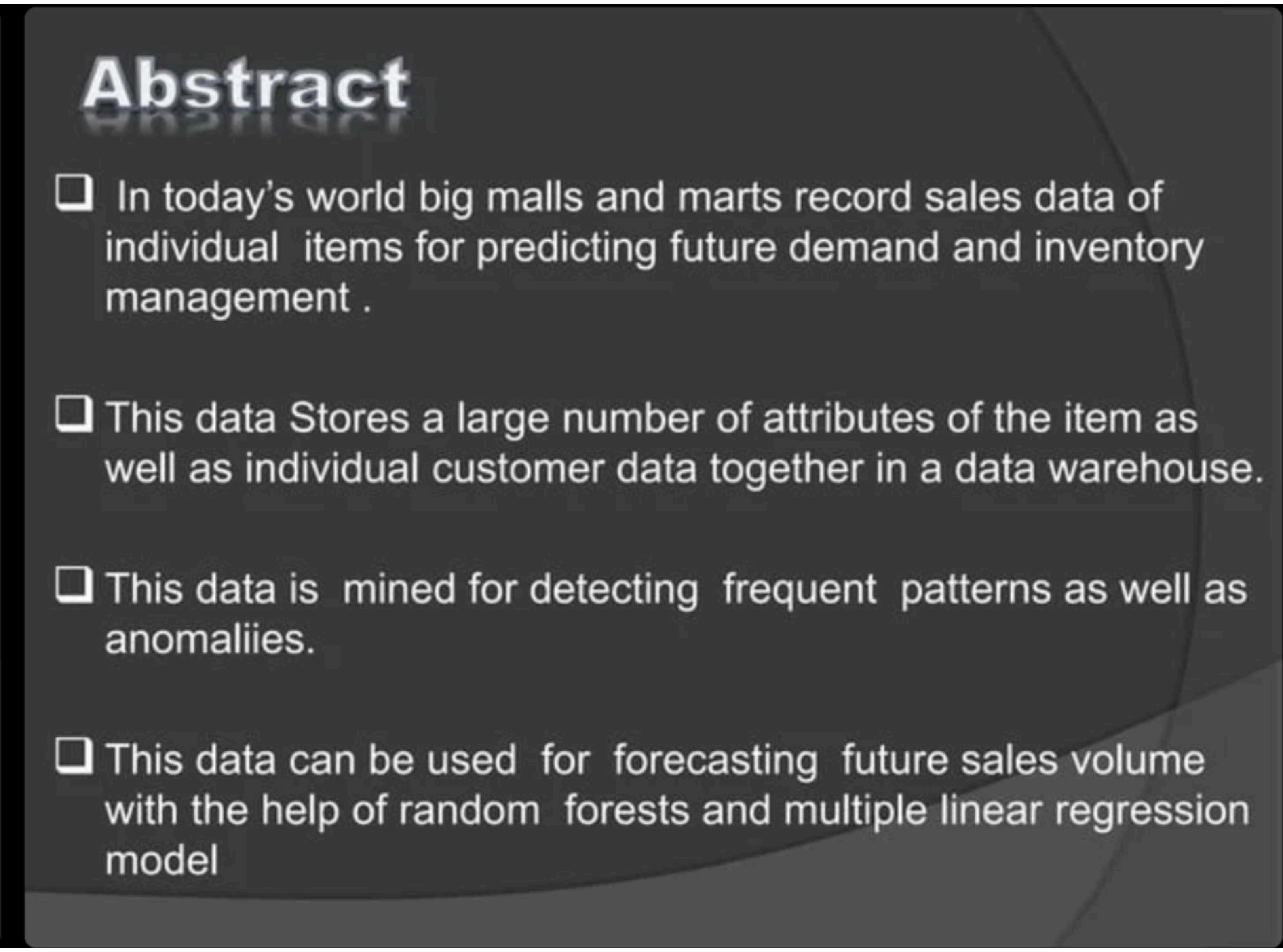
BIG MART SALES PREDICTION

Contetns:-

- Absrtact
- Introduction
- Problem Defination & Scope
- Software Requirement Specification
- Proposed Work
- Store Level Hypotheses
- Product Level Hypotheses
- Data Exploration
- Conclusion



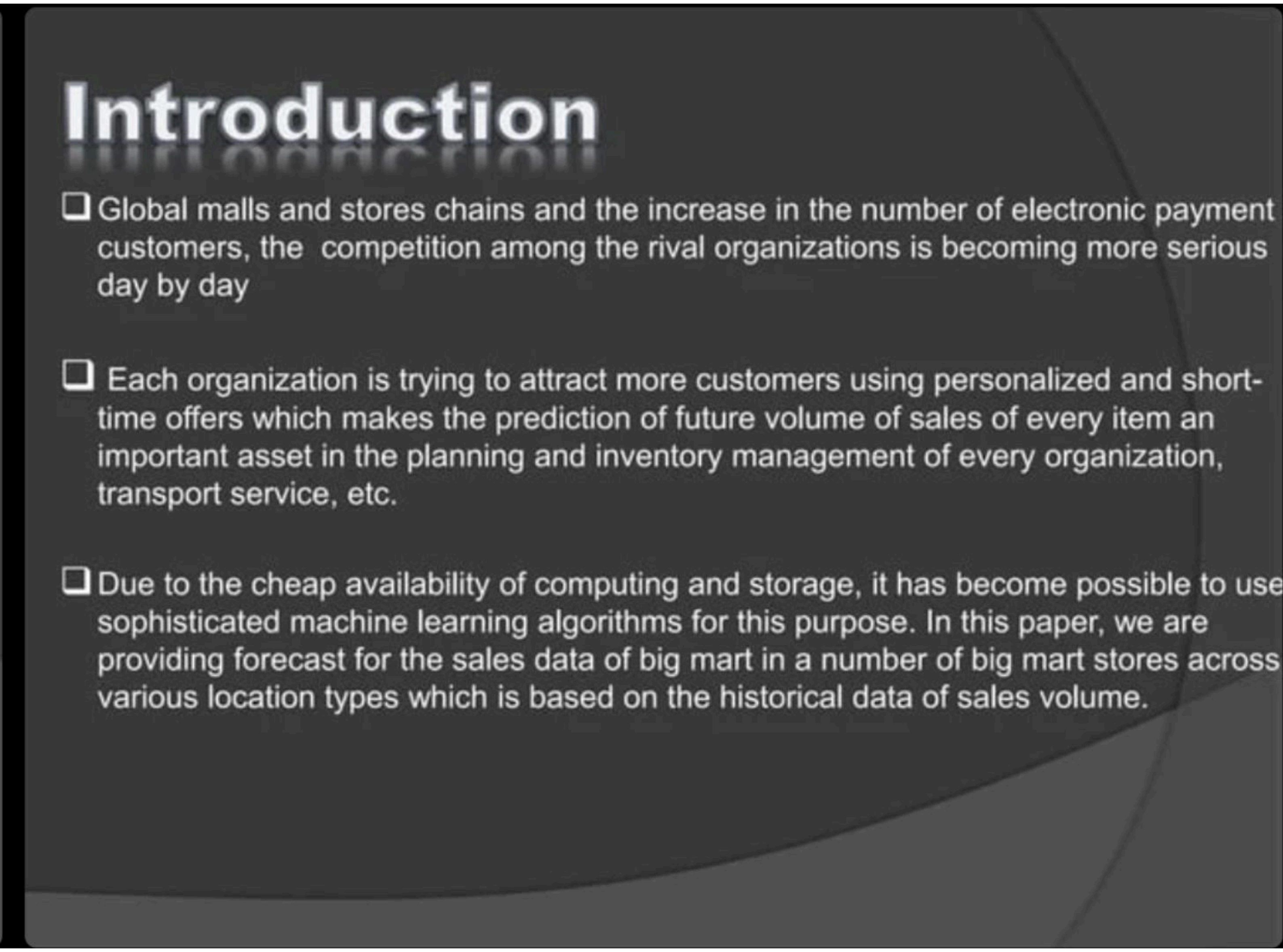
Abstract

❑ In today’s world big malls and marts record sales data of individual items for predicting future demand and inventory management .

❑ This data Stores a large number of attributes of the item as well as individual customer data together in a data warehouse.

❑ This data is mined for detecting frequent patterns as well as anomaliies.

❑ This data can be used for forecasting future sales volume with the help of random forests and multiple linear regression model

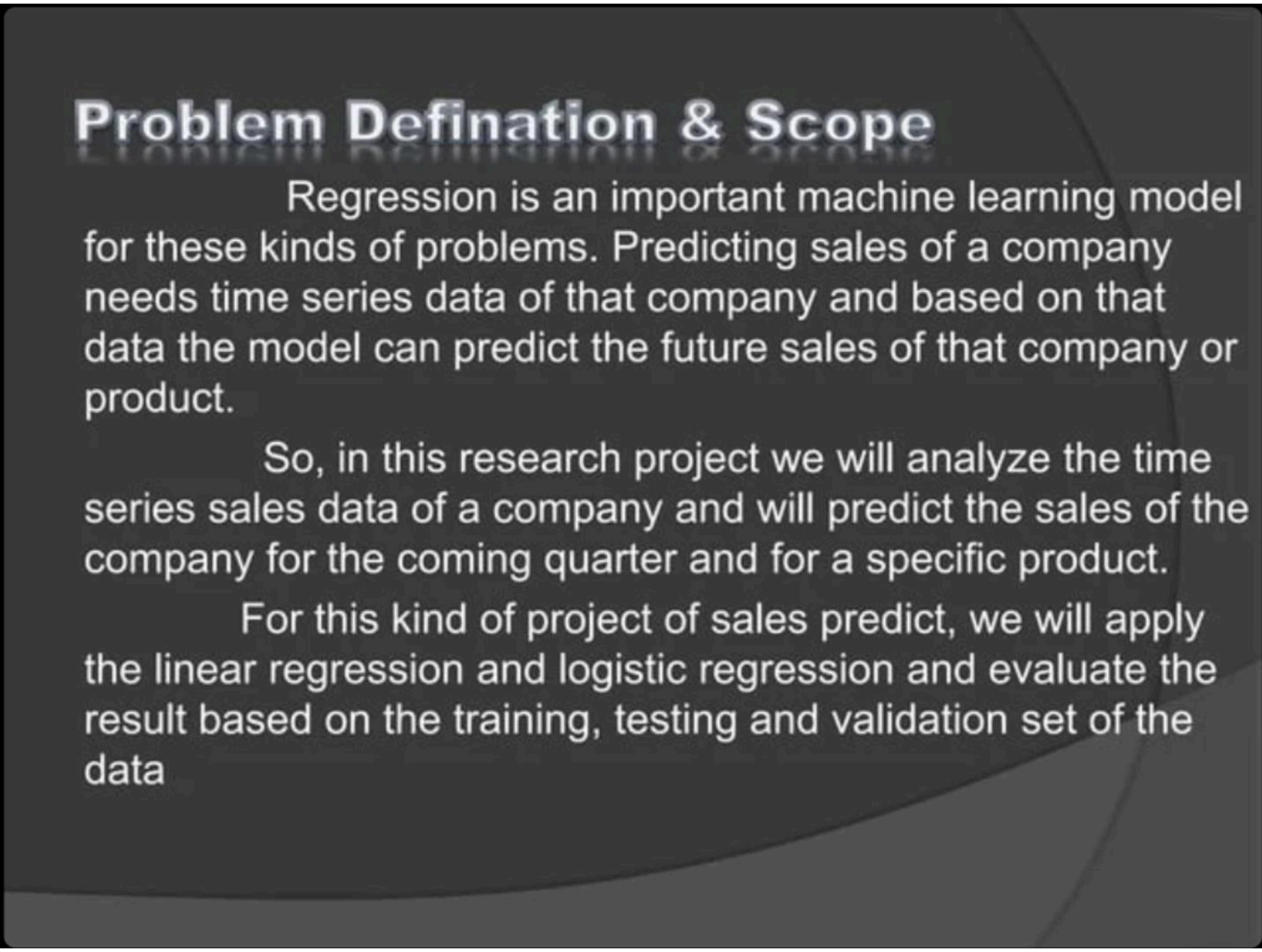


Introduction

□ Global malls and stores chains and the increase in the number of electronic payment customers, the competition among the rival organizations is becoming more serious day by day

□ Each organization is trying to attract more customers using personalized and short-time offers which makes the prediction of future volume of sales of every item an important asset in the planning and inventory management of every organization, transport service, etc.

□ Due to the cheap availability of computing and storage, it has become possible to use sophisticated machine learning algorithms for this purpose. In this paper, we are providing forecast for the sales data of big mart in a number of big mart stores across various location types which is based on the historical data of sales volume.

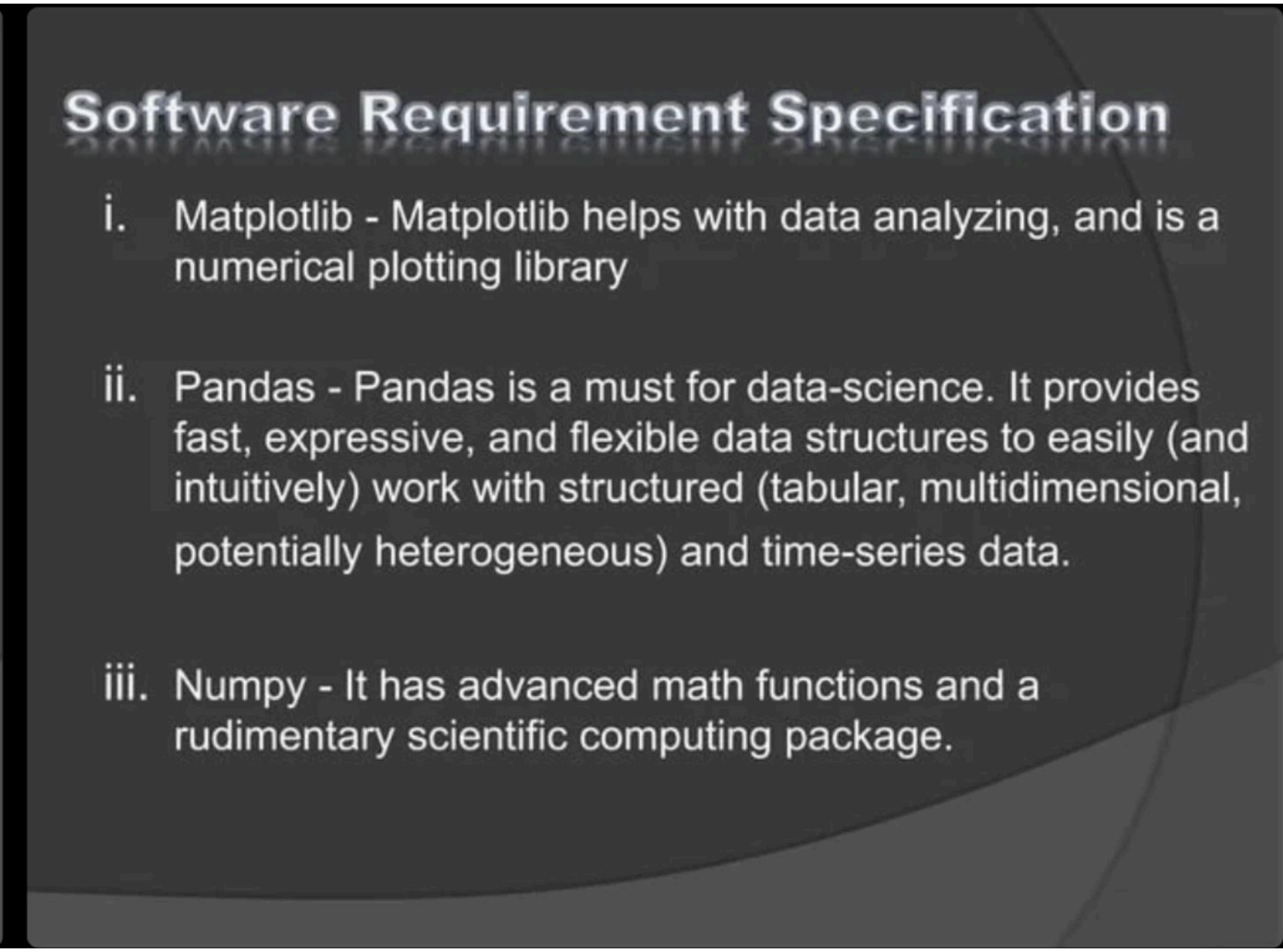


Problem Defination & Scope

Regression is an important machine learning model for these kinds of problems. Predicting sales of a company needs time series data of that company and based on that data the model can predict the future sales of that company or product.

So, in this research project we will analyze the time series sales data of a company and will predict the sales of the company for the coming quarter and for a specific product.

For this kind of project of sales predict, we will apply the linear regression and logistic regression and evaluate the result based on the training, testing and validation set of the data

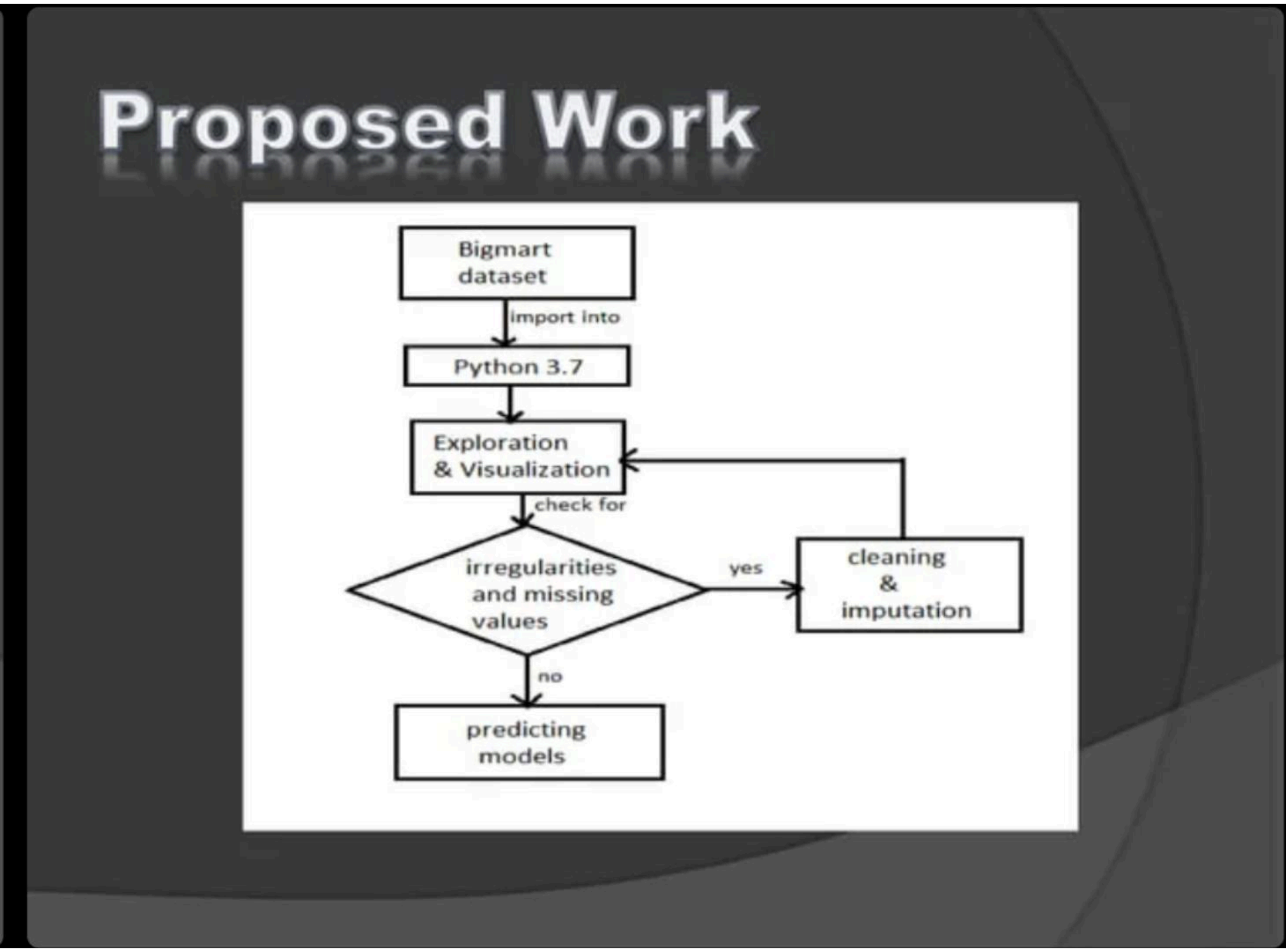


Software Requirement Specification

i. Matplotlib - Matplotlib helps with data analyzing, and is a numerical plotting library

ii. Pandas - Pandas is a must for data-science. It provides fast, expressive, and flexible data structures to easily (and intuitively) work with structured (tabular, multidimensional, potentially heterogeneous) and time-series data.

iii. Numpy - It has advanced math functions and a rudimentary scientific computing package.



Proposed Work

Bigmart dataset

import into

Python 3.7

Exploration & Visualization

check for

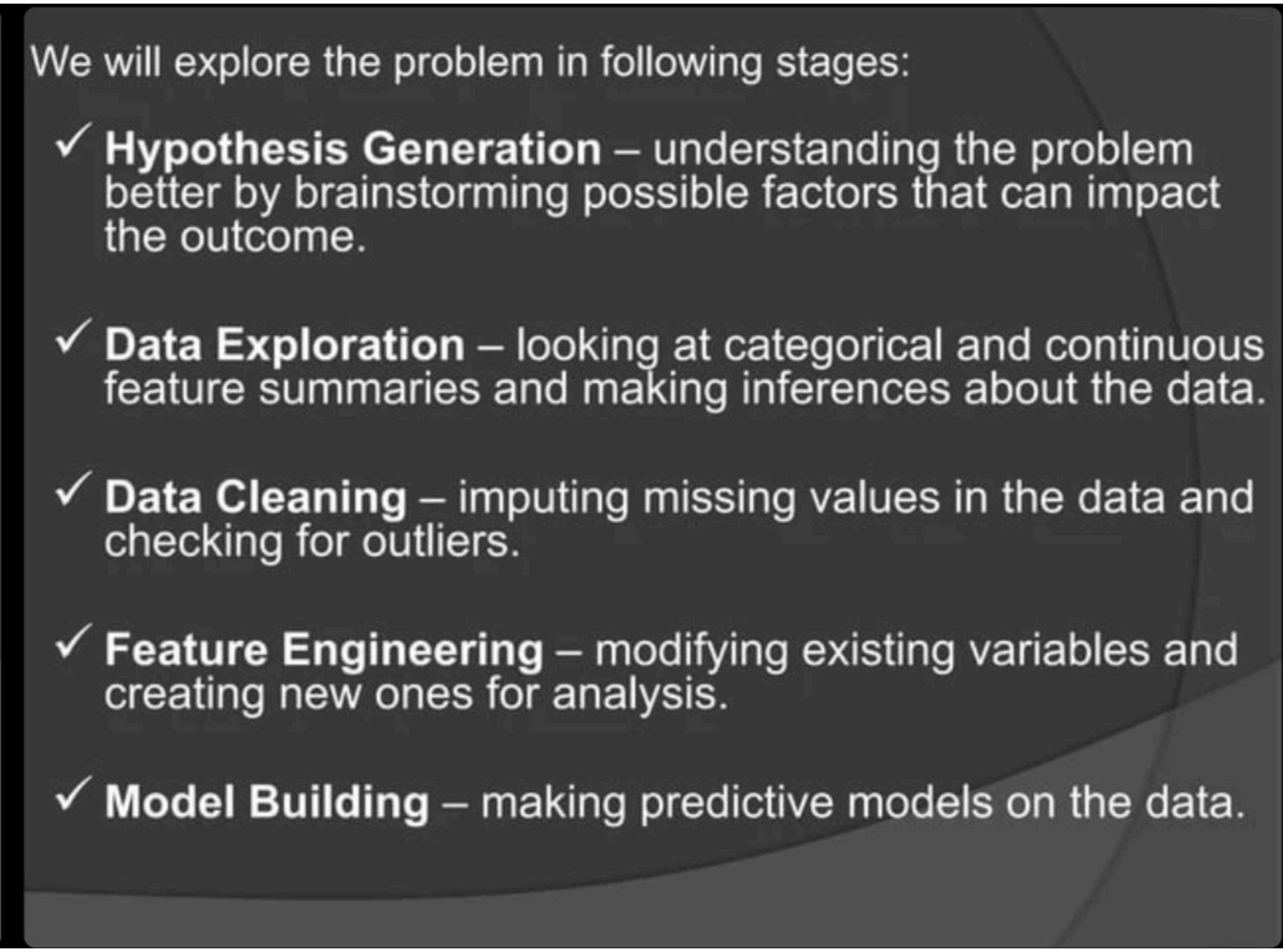
irregularities and missing values

yes

cleaning & imputation

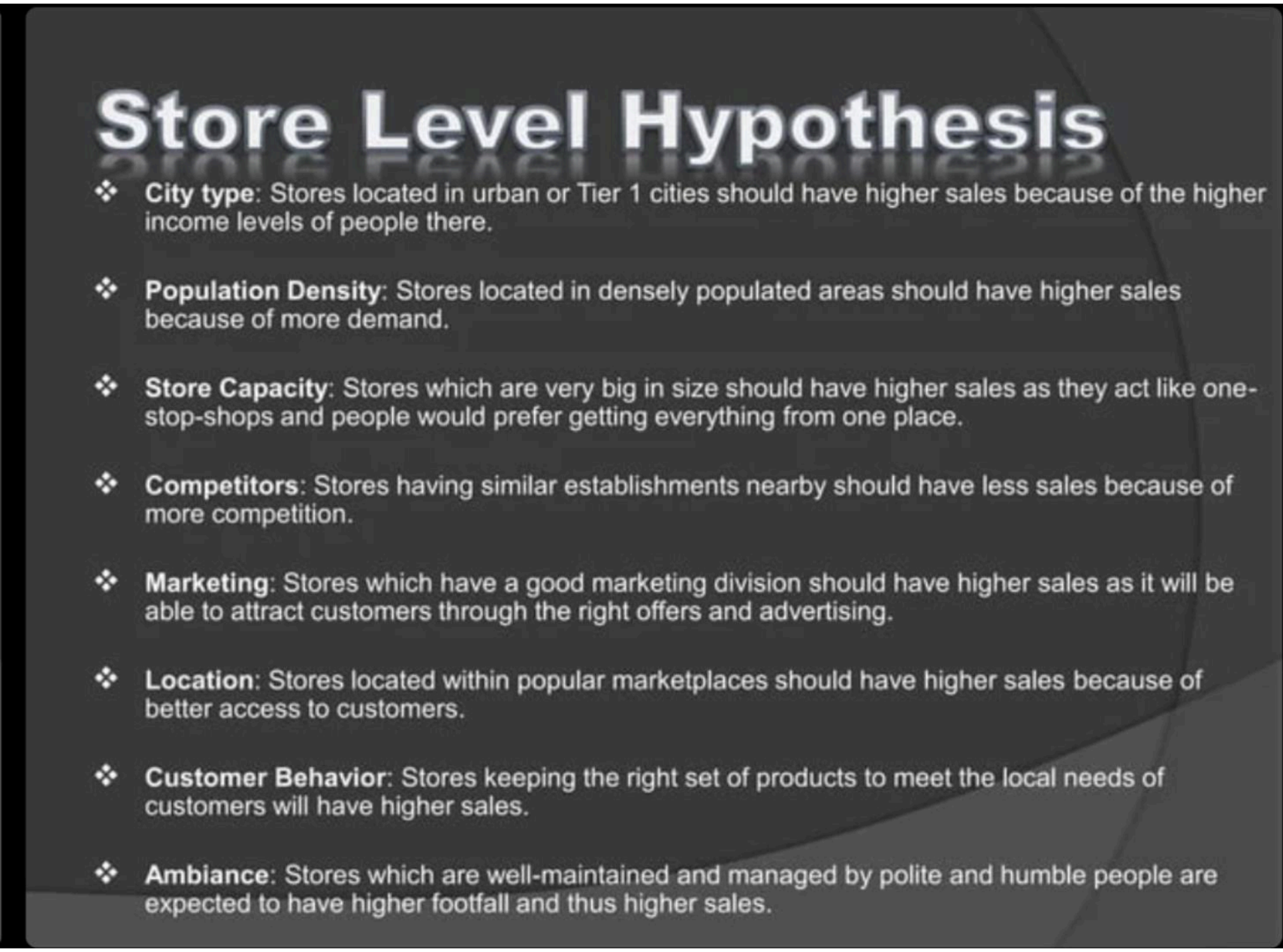
no

predicting models



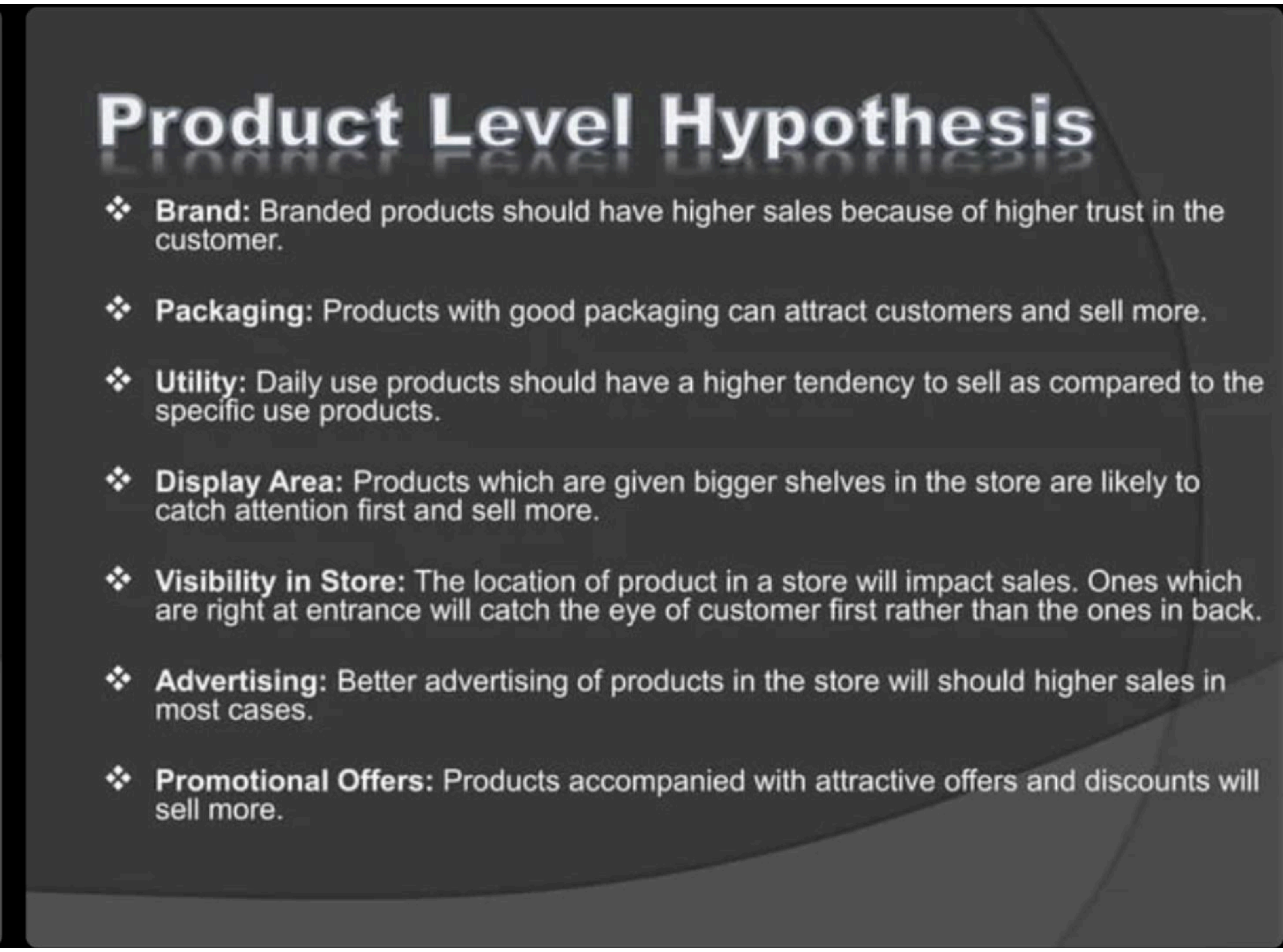
We will explore the problem in following stages:

- ✓ **Hypothesis Generation** – understanding the problem better by brainstorming possible factors that can impact the outcome.
- ✓ **Data Exploration** – looking at categorical and continuous feature summaries and making inferences about the data.
- ✓ **Data Cleaning** – imputing missing values in the data and checking for outliers.
- ✓ **Feature Engineering** – modifying existing variables and creating new ones for analysis.
- ✓ **Model Building** – making predictive models on the data.



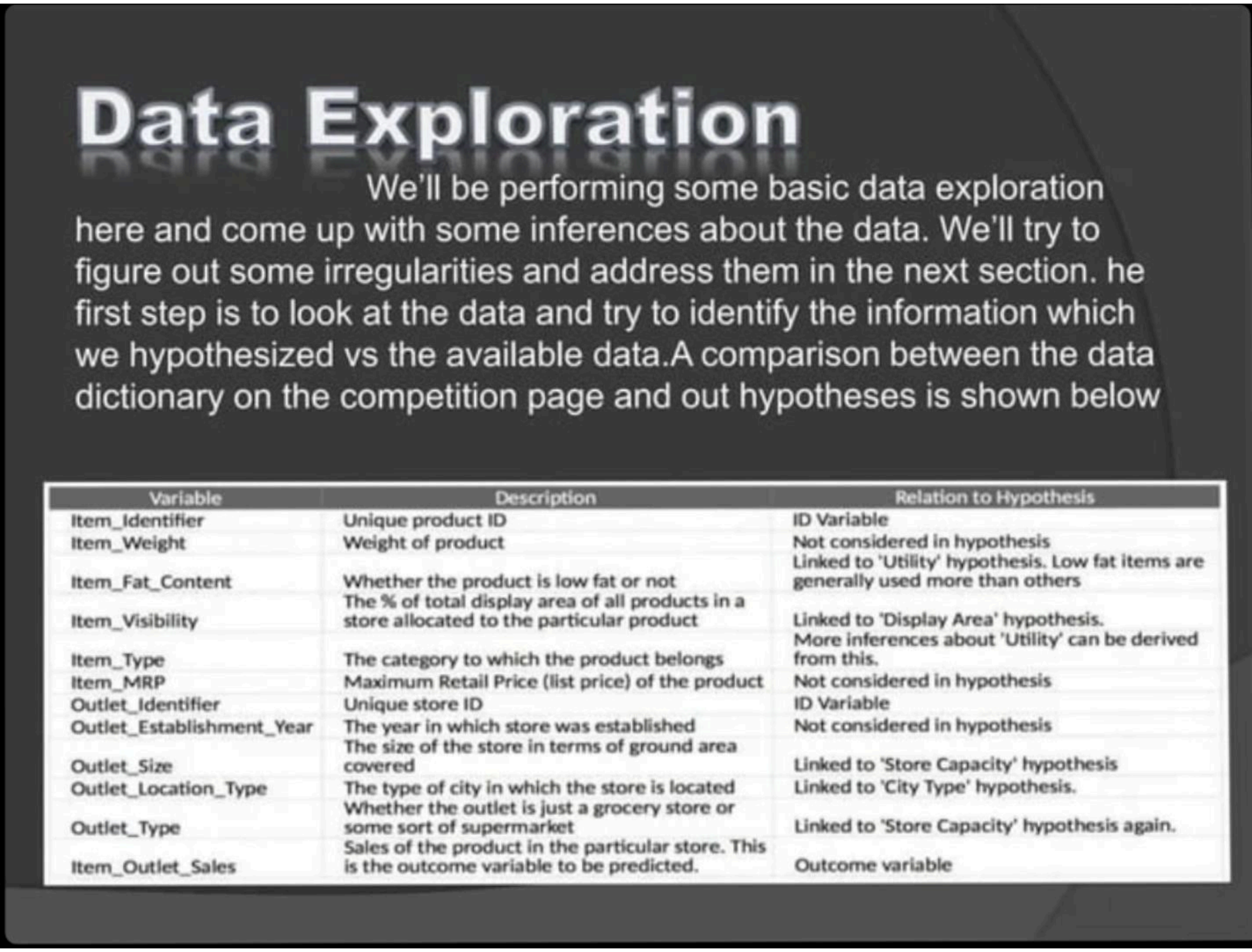
Store Level Hypothesis

- **City type**: Stores located in urban or Tier 1 cities should have higher sales because of the higher income levels of people there.
- **Population Density**: Stores located in densely populated areas should have higher sales because of more demand.
- **Store Capacity**: Stores which are very big in size should have higher sales as they act like one-stop-shops and people would prefer getting everything from one place.
- **Competitors**: Stores having similar establishments nearby should have less sales because of more competition.
- **Marketing**: Stores which have a good marketing division should have higher sales as it will be able to attract customers through the right offers and advertising.
- **Location**: Stores located within popular marketplaces should have higher sales because of better access to customers.
- **Customer Behavior**: Stores keeping the right set of products to meet the local needs of customers will have higher sales.
- **Ambiance**: Stores which are well-maintained and managed by polite and humble people are expected to have higher footfall and thus higher sales.



Product Level Hypothesis

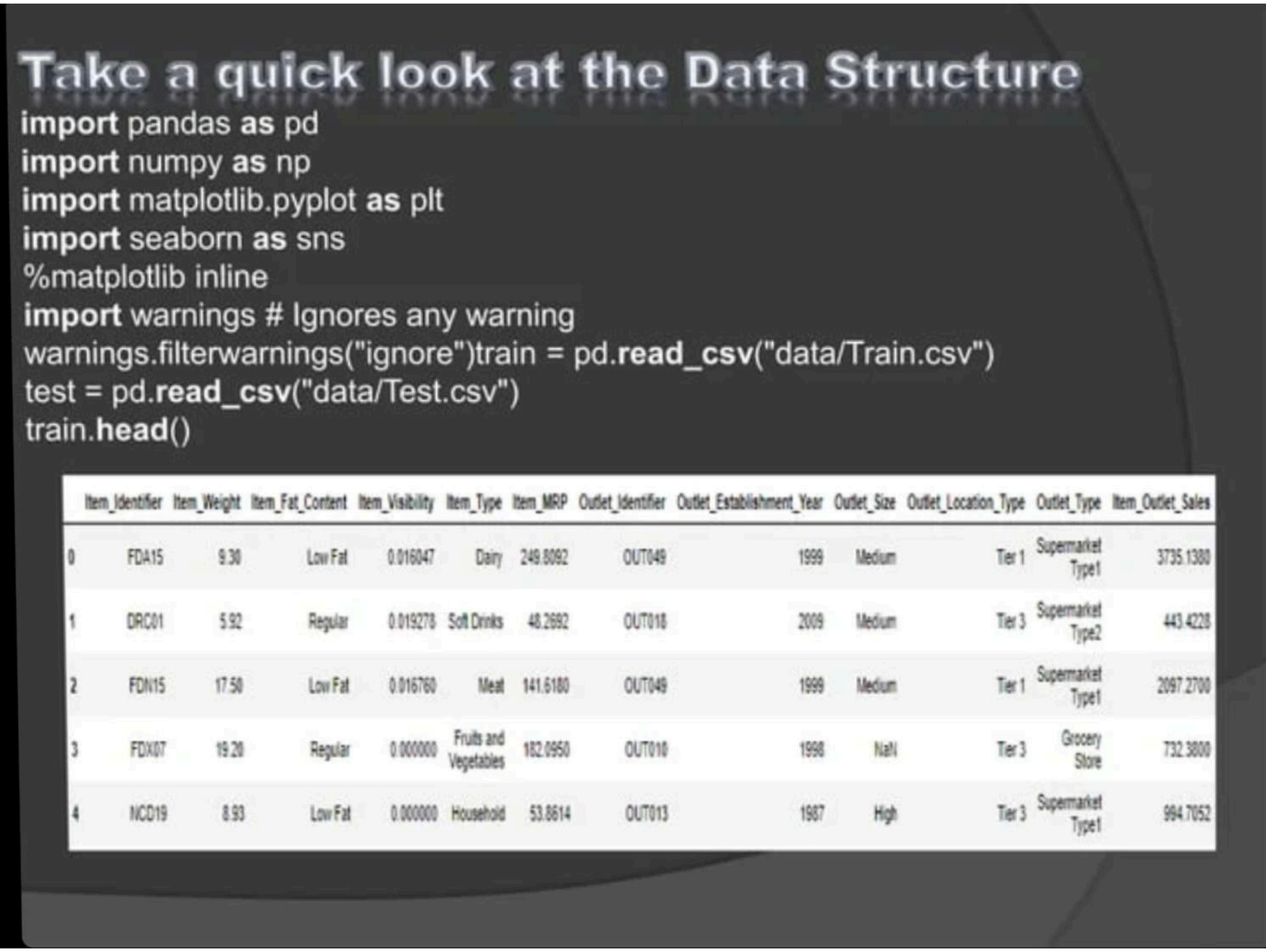
- **Brand:** Branded products should have higher sales because of higher trust in the customer.
- **Packaging:** Products with good packaging can attract customers and sell more.
- **Utility:** Daily use products should have a higher tendency to sell as compared to the specific use products.
- **Display Area:** Products which are given bigger shelves in the store are likely to catch attention first and sell more.
- **Visibility in Store:** The location of product in a store will impact sales. Ones which are right at entrance will catch the eye of customer first rather than the ones in back.
- **Advertising:** Better advertising of products in the store will should higher sales in most cases.
- **Promotional Offers:** Products accompanied with attractive offers and discounts will sell more.



Data Exploration

We'll be performing some basic data exploration here and come up with some inferences about the data. We'll try to figure out some irregularities and address them in the next section. he first step is to look at the data and try to identify the information which we hypothesized vs the available data.A comparison between the data dictionary on the competition page and out hypotheses is shown below

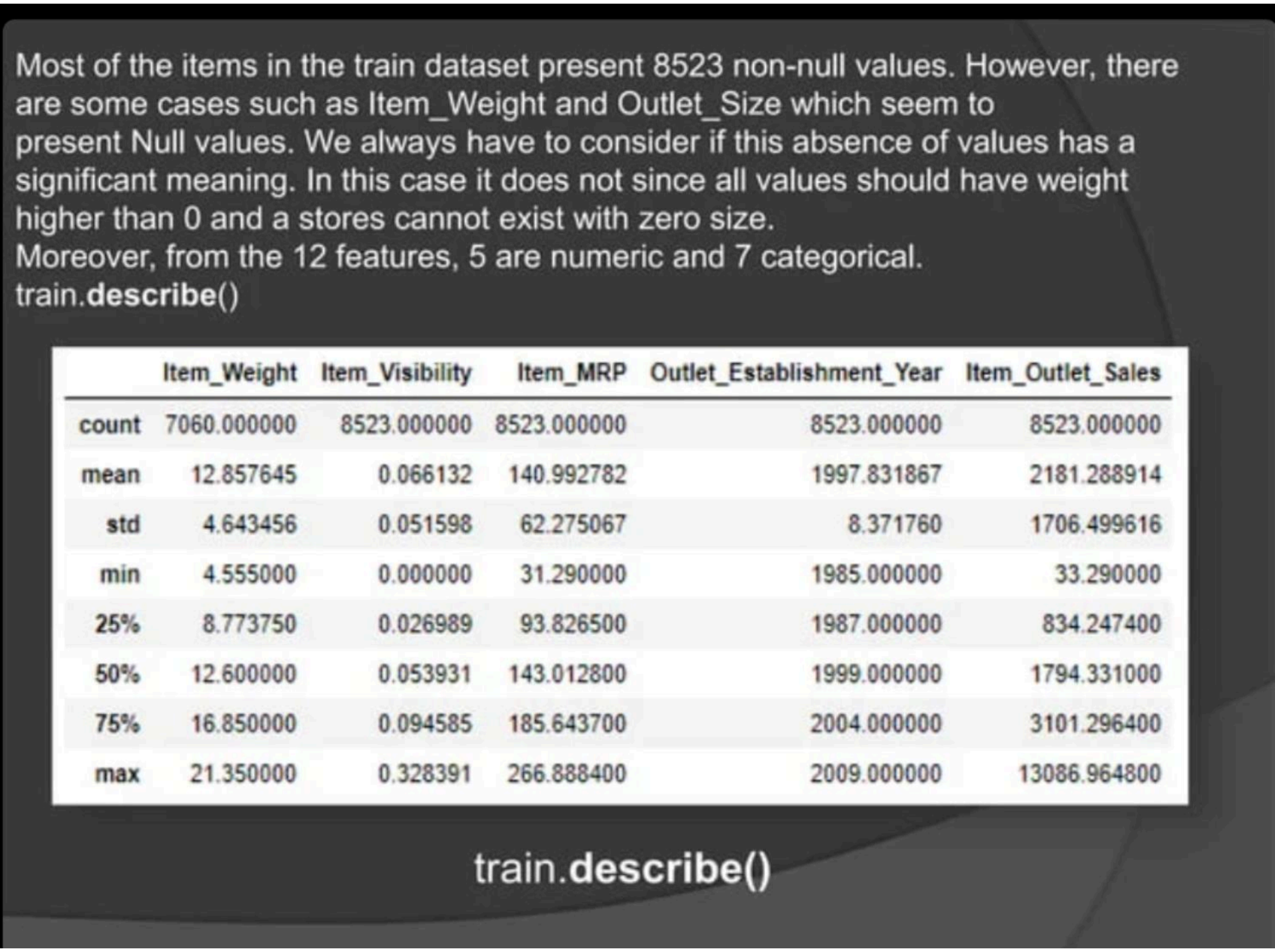
Variable	Description	Relation to Hypothesis
Item_Identifier	Unique product ID	ID Variable
Item_Weight	Weight of product	Not considered in hypothesis
Item_Fat_Content	Whether the product is low fat or not	Linked to 'Utility' hypothesis. Low fat items are generally used more than others
Item_Visibility	The % of total display area of all products in a store allocated to the particular product	Linked to 'Display Area' hypothesis.
Item_Type	The category to which the product belongs	More inferences about 'Utility' can be derived from this.
Item_MRP	Maximum Retail Price (list price) of the product	Not considered in hypothesis
Outlet_Identifier	Unique store ID	ID Variable
Outlet_Establishment_Year	The year in which store was established	Not considered in hypothesis
Outlet_Size	The size of the store in terms of ground area covered	Linked to 'Store Capacity' hypothesis
Outlet_Location_Type	The type of city in which the store is located	Linked to 'City Type' hypothesis.
Outlet_Type	Whether the outlet is just a grocery store or some sort of supermarket	Linked to 'Store Capacity' hypothesis again.
Item_Outlet_Sales	Sales of the product in the particular store. This is the outcome variable to be predicted.	Outcome variable



Take a quick look at the Data Structure

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings # Ignores any warning
warnings.filterwarnings("ignore")train = pd.read_csv("data/Train.csv")
test = pd.read_csv("data/Test.csv")
train.head()
```

	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type	Outlet_Type	Item_Outlet_Sales
0	FDA15	9.30	Low Fat	0.016047	Dairy	249.8092	OUT049	1999	Medium	Tier 1	Supermarket Type1	3735.1380
1	DRC01	5.92	Regular	0.019278	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3	Supermarket Type2	443.4228
2	FDN15	17.50	Low Fat	0.016760	Meat	141.6180	OUT049	1999	Medium	Tier 1	Supermarket Type1	2097.2700
3	FDX07	19.20	Regular	0.000000	Fruits and Vegetables	182.0950	OUT010	1998	NaN	Tier 3	Grocery Store	732.3800
4	NCD19	8.93	Low Fat	0.000000	Household	53.8614	OUT013	1987	High	Tier 3	Supermarket Type1	994.7052



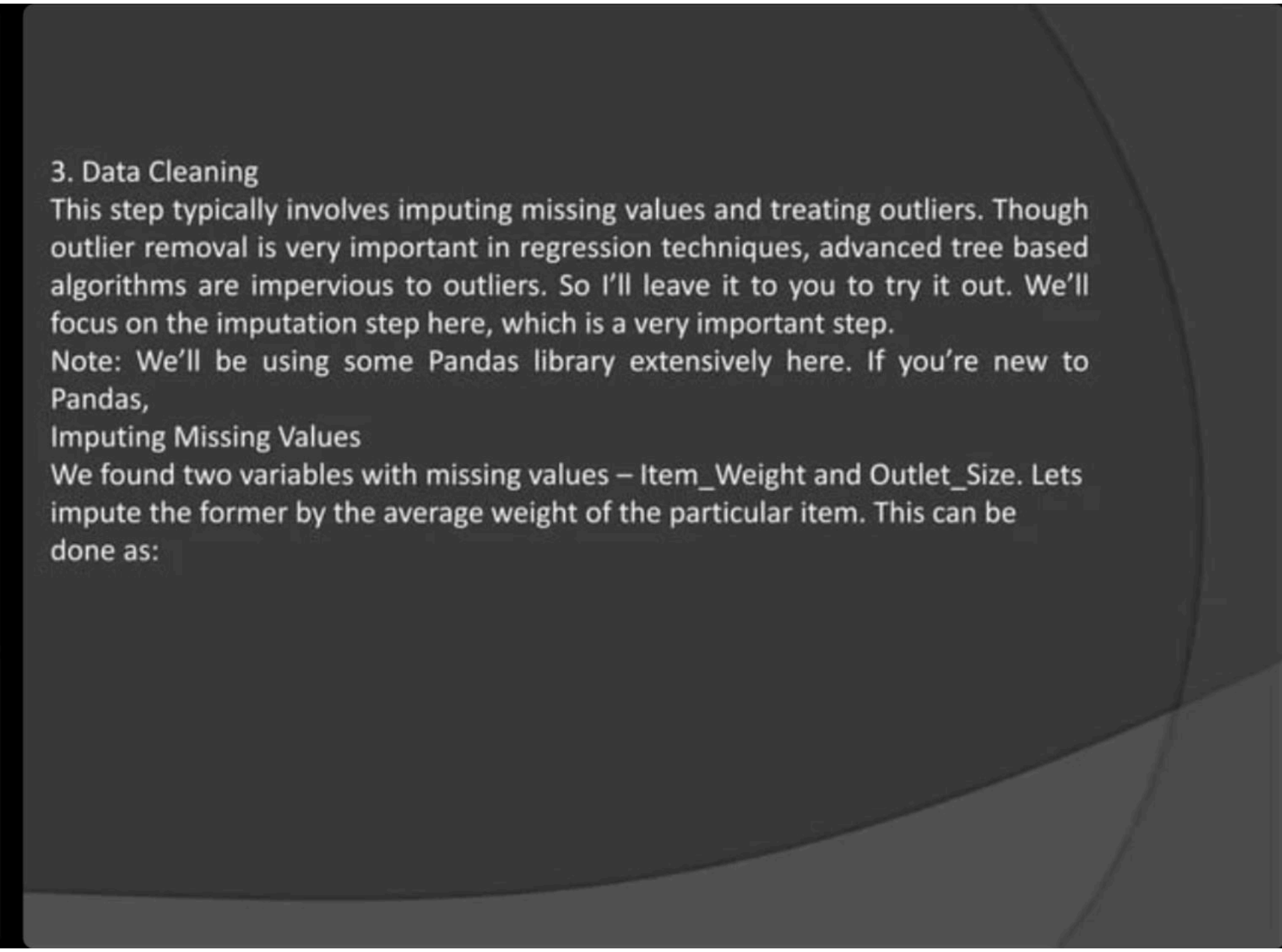
Most of the items in the train dataset present 8523 non-null values. However, there are some cases such as Item_Weight and Outlet_Size which seem to present Null values. We always have to consider if this absence of values has a significant meaning. In this case it does not since all values should have weight higher than 0 and a stores cannot exist with zero size.

Moreover, from the 12 features, 5 are numeric and 7 categorical.

train.**describe**()

	Item_Weight	Item_Visibility	Item_MRP	Outlet_Establishment_Year	Item_Outlet_Sales
count	7060.000000	8523.000000	8523.000000	8523.000000	8523.000000
mean	12.857645	0.066132	140.992782	1997.831867	2181.288914
std	4.643456	0.051598	62.275067	8.371760	1706.499616
min	4.555000	0.000000	31.290000	1985.000000	33.290000
25%	8.773750	0.026989	93.826500	1987.000000	834.247400
50%	12.600000	0.053931	143.012800	1999.000000	1794.331000
75%	16.850000	0.094585	185.643700	2004.000000	3101.296400
max	21.350000	0.328391	266.888400	2009.000000	13086.964800

train.**describe()**



3. Data Cleaning

This step typically involves imputing missing values and treating outliers. Though outlier removal is very important in regression techniques, advanced tree based algorithms are impervious to outliers. So I'll leave it to you to try it out. We'll focus on the imputation step here, which is a very important step.

Note: We'll be using some Pandas library extensively here. If you're new to Pandas,

Imputing Missing Values

We found two variables with missing values – Item_Weight and Outlet_Size. Lets impute the former by the average weight of the particular item. This can be done as:

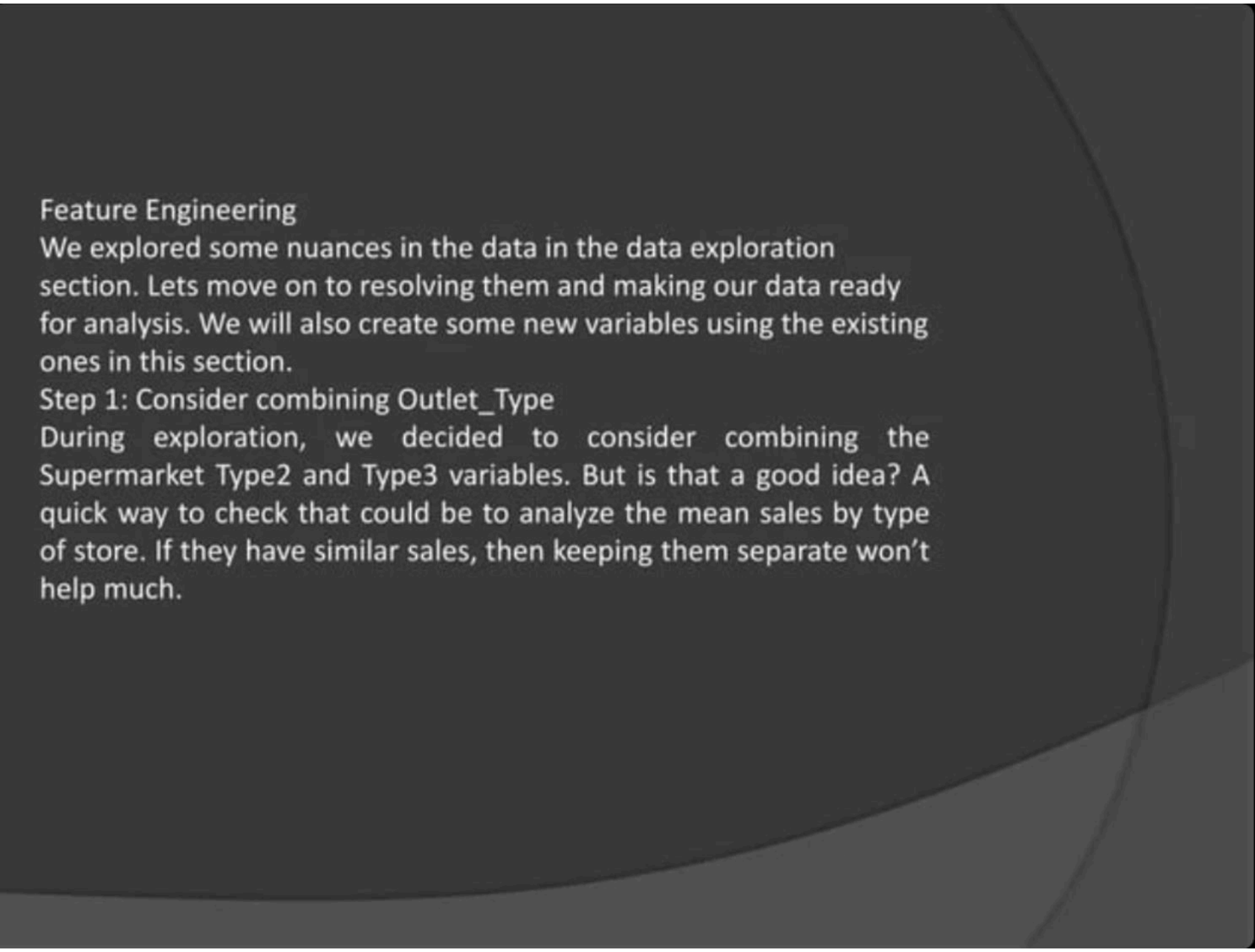
3. Data Cleaning

This step typically involves imputing missing values and treating outliers. Though outlier removal is very important in regression techniques, advanced tree based algorithms are impervious to outliers. So I'll leave it to you to try it out. We'll focus on the imputation step here, which is a very important step.

Note: We'll be using some Pandas library extensively here. If you're new to Pandas,

Imputing Missing Values

We found two variables with missing values – Item_Weight and Outlet_Size. Lets impute the former by the average weight of the particular item. This can be done as:

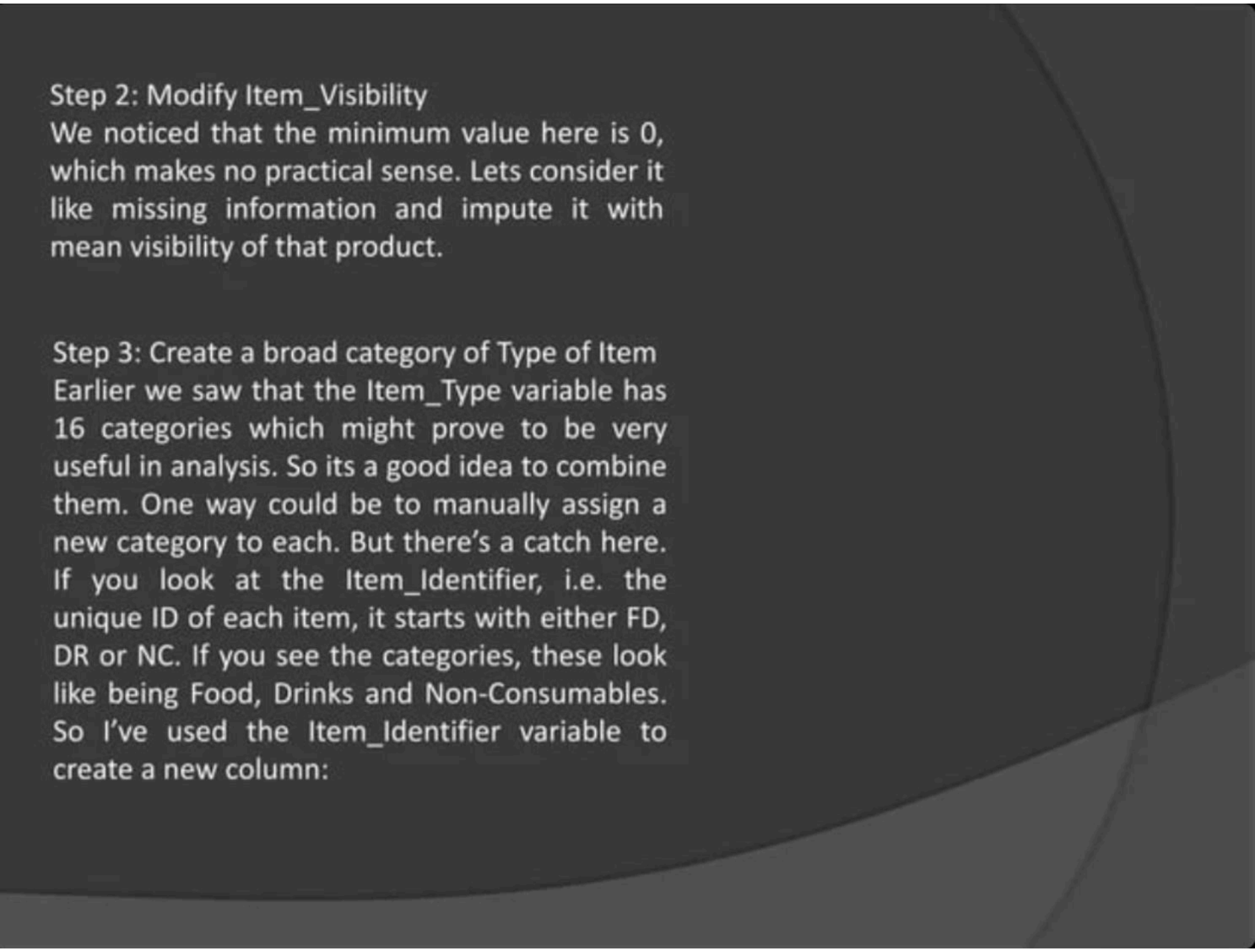


Feature Engineering

We explored some nuances in the data in the data exploration section. Lets move on to resolving them and making our data ready for analysis. We will also create some new variables using the existing ones in this section.

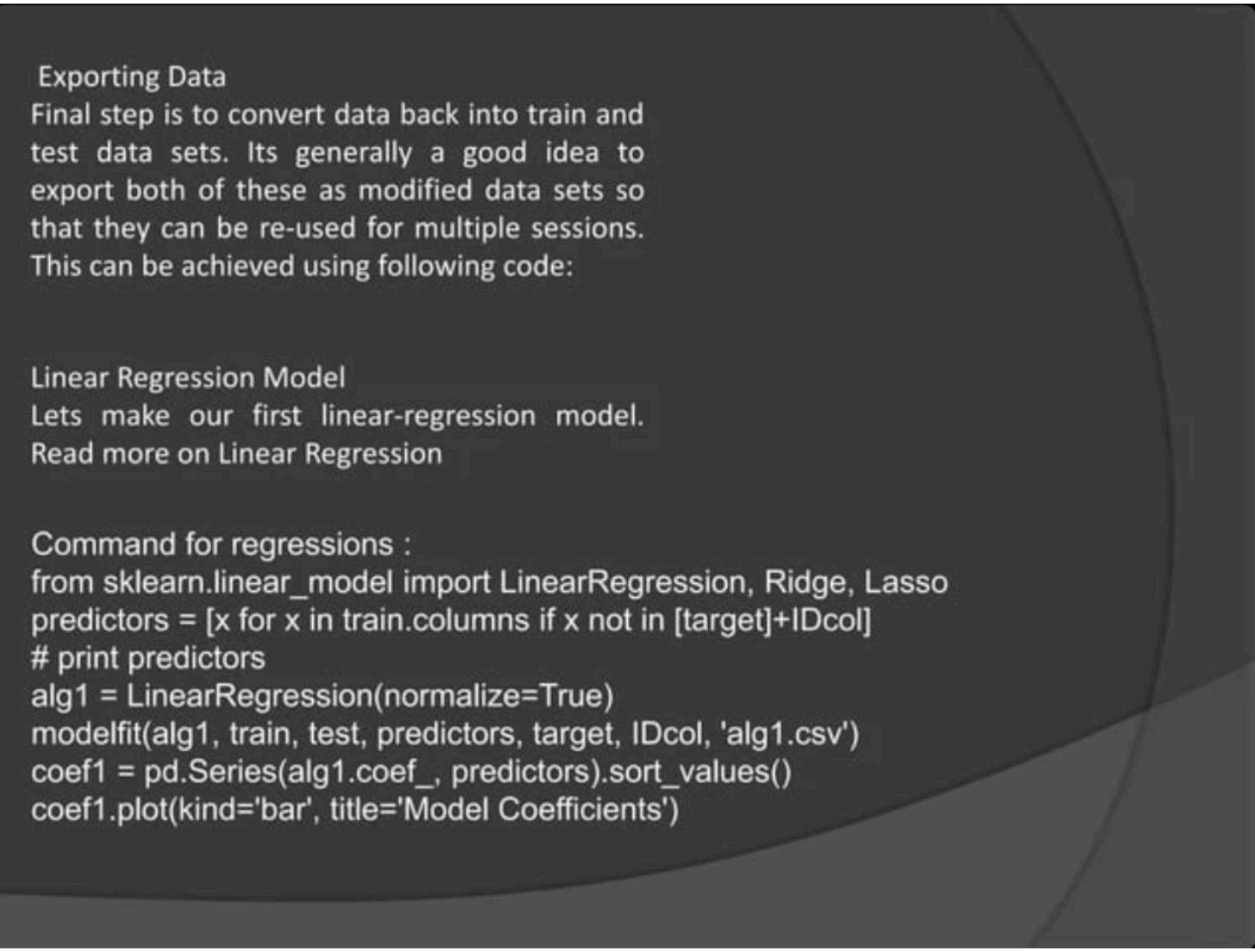
Step 1: Consider combining Outlet_Type

During exploration, we decided to consider combining the Supermarket Type2 and Type3 variables. But is that a good idea? A quick way to check that could be to analyze the mean sales by type of store. If they have similar sales, then keeping them separate won’t help much.



Step 2: Modify Item_Visibility
We noticed that the minimum value here is 0, which makes no practical sense. Lets consider it like missing information and impute it with mean visibility of that product.

Step 3: Create a broad category of Type of Item
Earlier we saw that the Item_Type variable has 16 categories which might prove to be very useful in analysis. So its a good idea to combine them. One way could be to manually assign a new category to each. But there's a catch here. If you look at the Item_Identifier, i.e. the unique ID of each item, it starts with either FD, DR or NC. If you see the categories, these look like being Food, Drinks and Non-Consumables. So I've used the Item_Identifier variable to create a new column:



Exporting Data

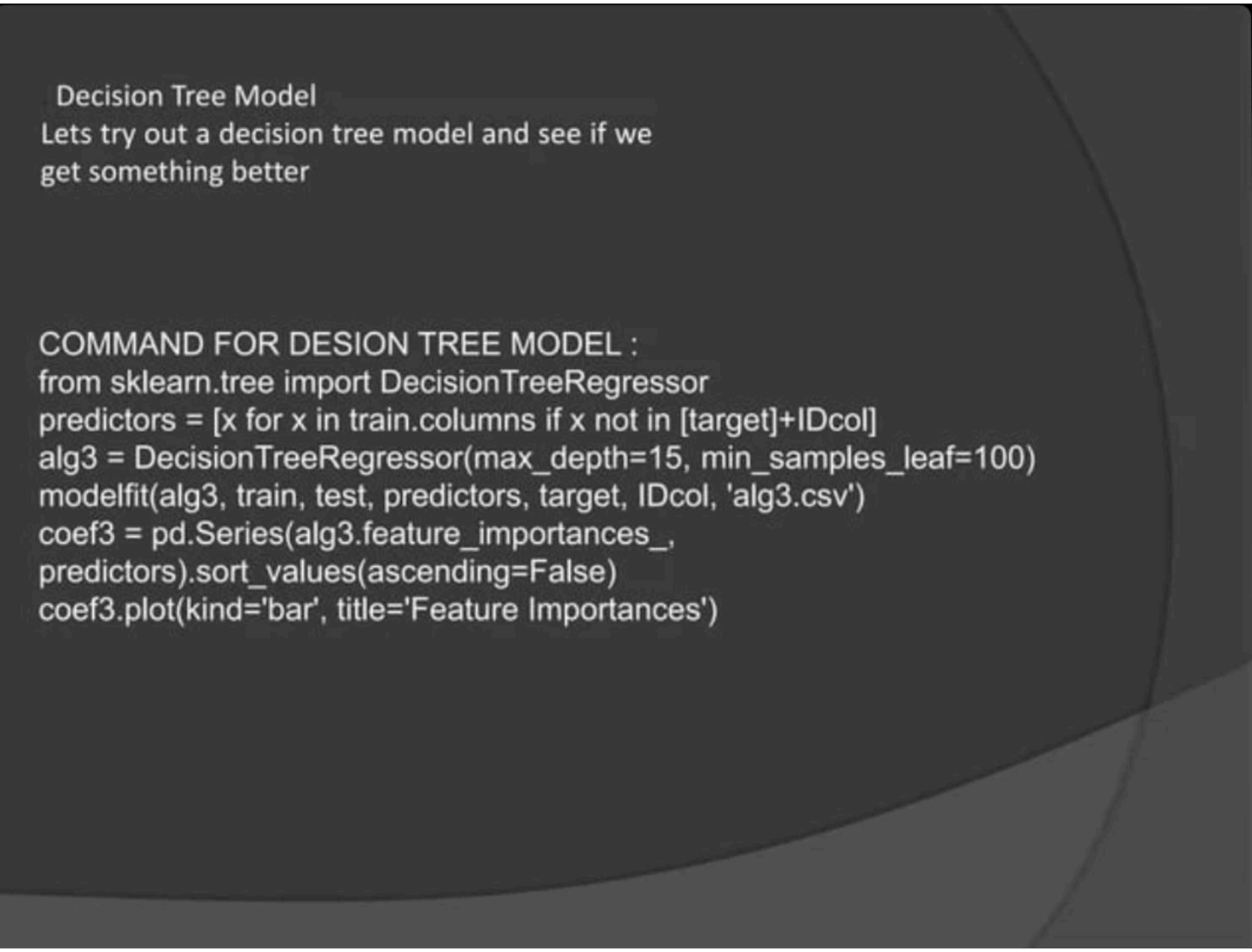
Final step is to convert data back into train and test data sets. Its generally a good idea to export both of these as modified data sets so that they can be re-used for multiple sessions. This can be achieved using following code:

Linear Regression Model

Lets make our first linear-regression model. Read more on Linear Regression

Command for regressions :

```
from sklearn.linear_model import LinearRegression, Ridge, Lasso
predictors = [x for x in train.columns if x not in [target]+IDcol]
# print predictors
alg1 = LinearRegression(normalize=True)
modelfit(alg1, train, test, predictors, target, IDcol, 'alg1.csv')
coef1 = pd.Series(alg1.coef_, predictors).sort_values()
coef1.plot(kind='bar', title='Model Coefficients')
```



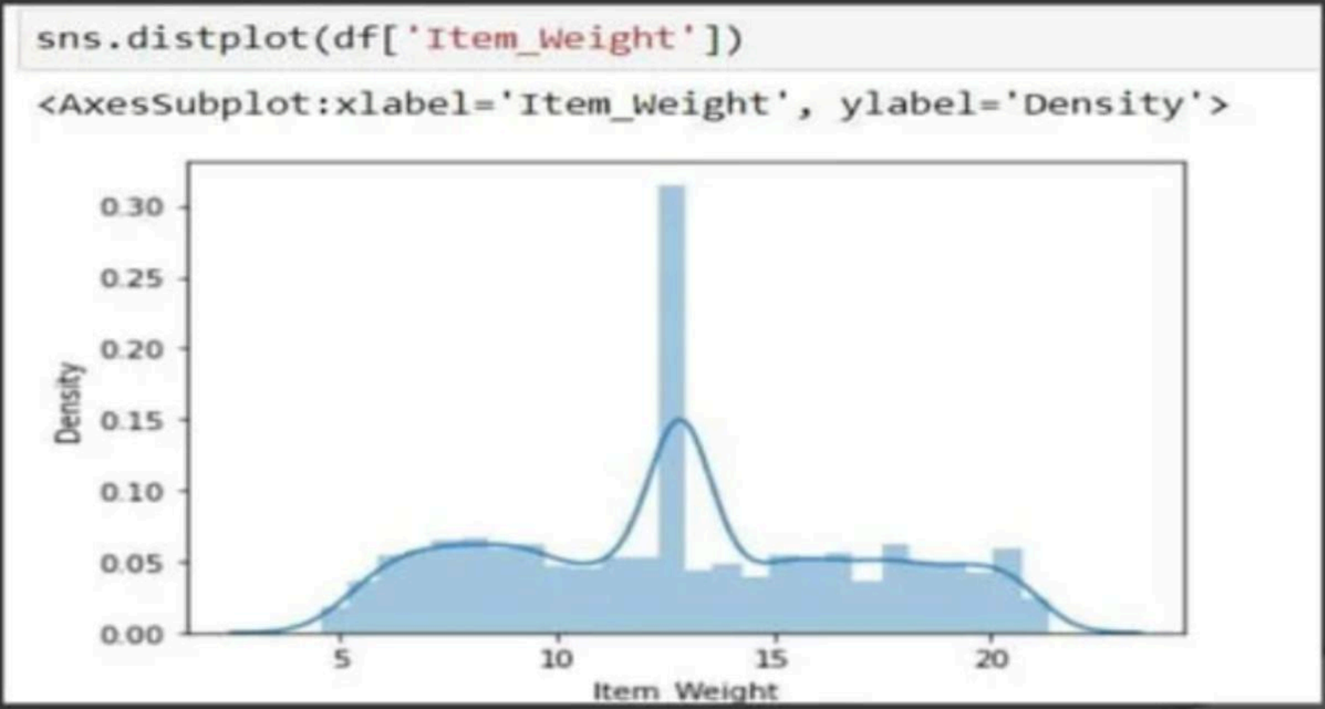
Decision Tree Model

Lets try out a decision tree model and see if we get something better

COMMAND FOR DESION TREE MODEL :

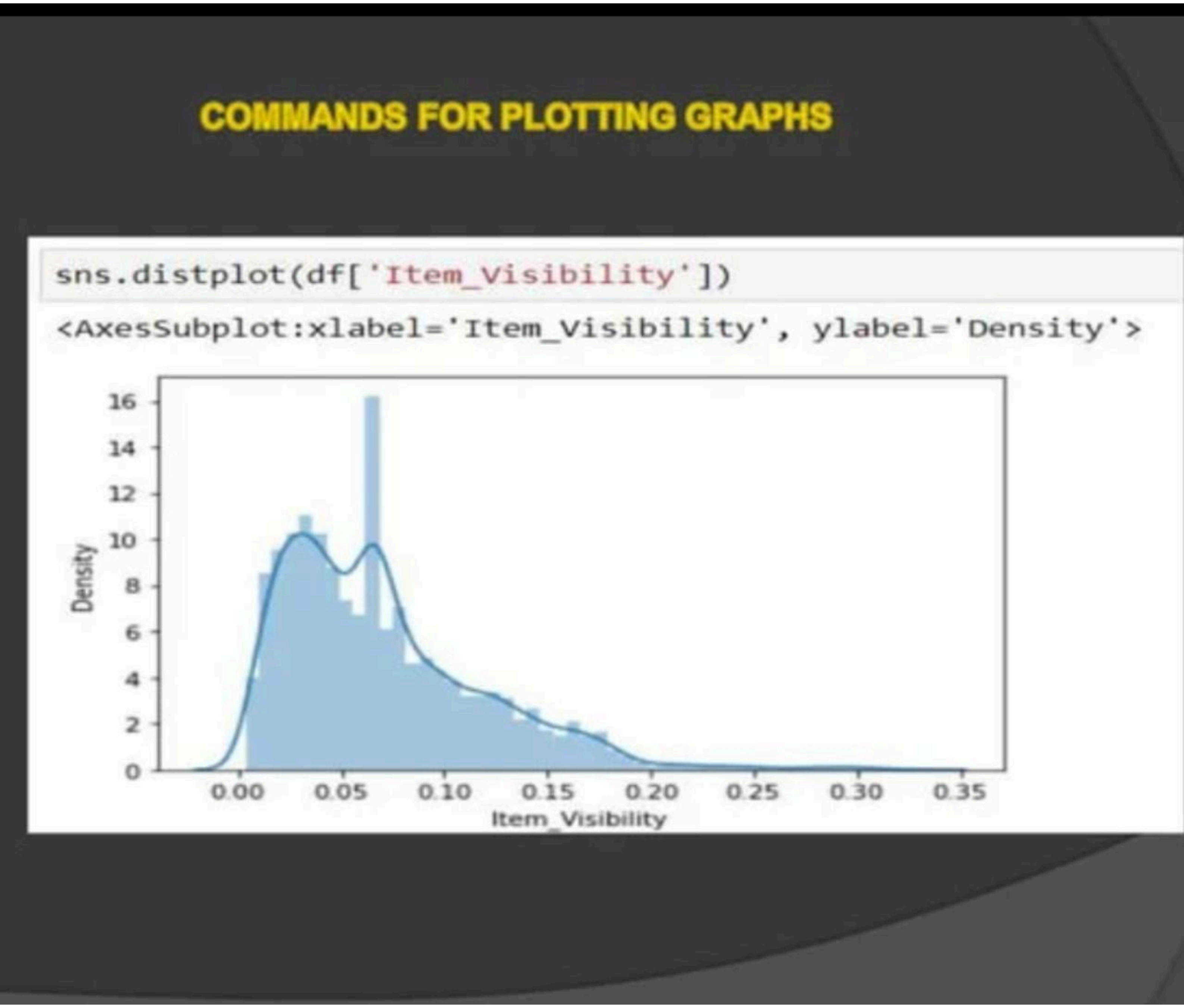
```
from sklearn.tree import DecisionTreeRegressor
predictors = [x for x in train.columns if x not in [target]+IDcol]
alg3 = DecisionTreeRegressor(max_depth=15, min_samples_leaf=100)
modelfit(alg3, train, test, predictors, target, IDcol, 'alg3.csv')
coef3 = pd.Series(alg3.feature_importances_,
predictors).sort_values(ascending=False)
coef3.plot(kind='bar', title='Feature Importances')
```

COMMANDS FOR PLOTTING GRAPHS



```
sns.distplot(df['Item_Weight'])
```

```
<AxesSubplot:xlabel='Item_Weight', ylabel='Density'>
```

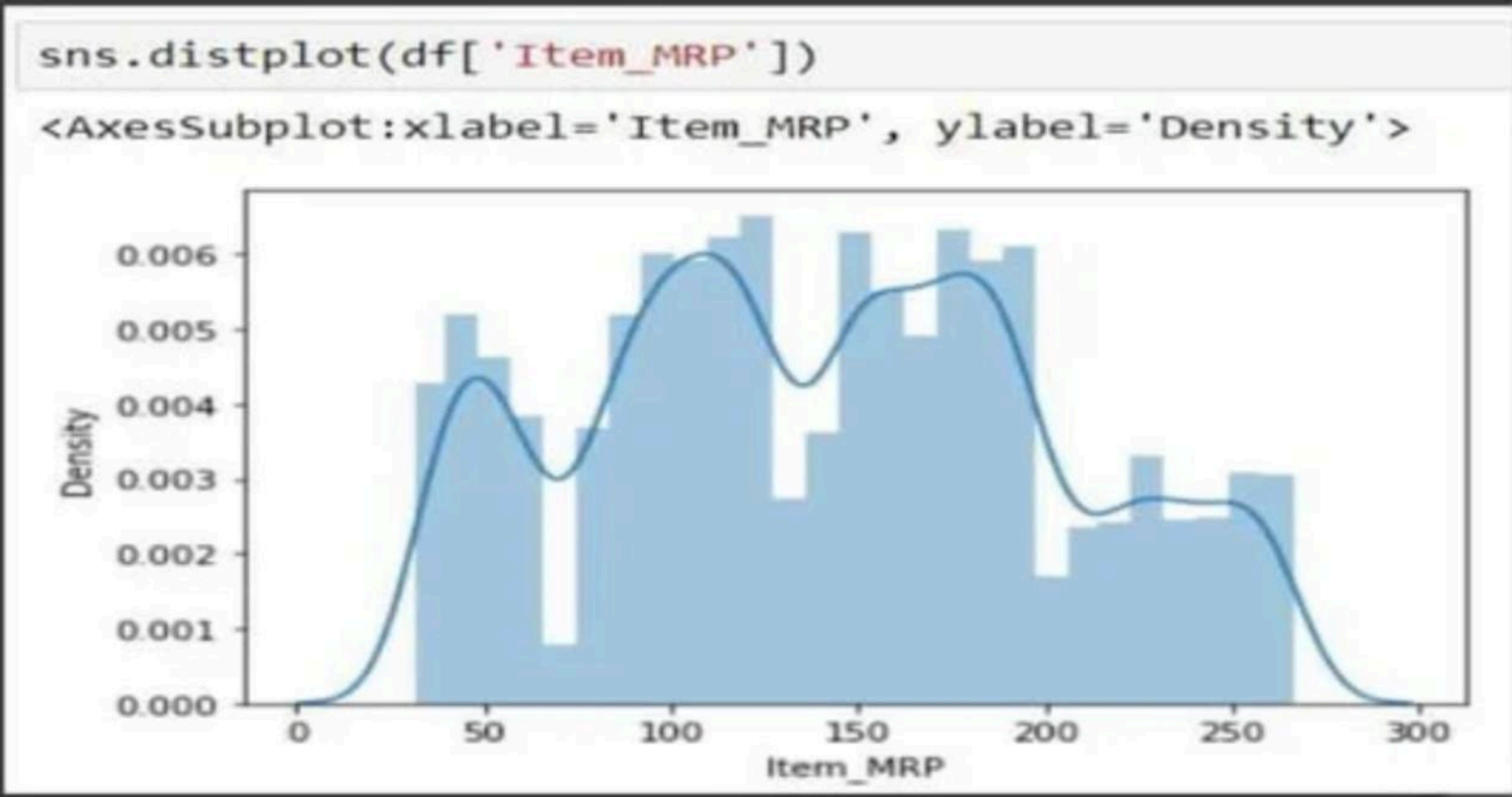


COMMANDS FOR PLOTTING GRAPHS

```
sns.distplot(df['Item_Visibility'])
```

```
<AxesSubplot:xlabel='Item_Visibility', ylabel='Density'>
```

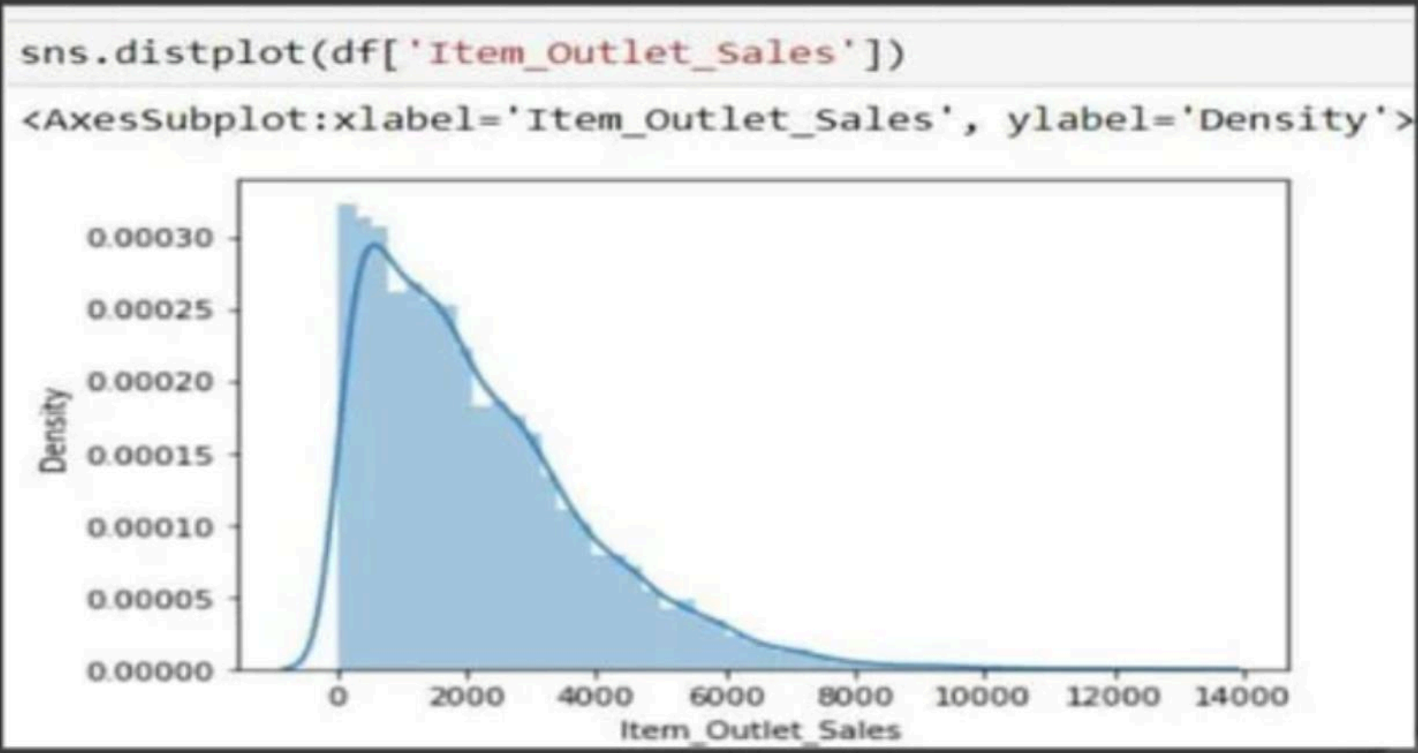
COMMANDS FOR PLOTTING GRAPHS



```
sns.distplot(df['Item_MRP'])
```

```
<AxesSubplot:xlabel='Item_MRP', ylabel='Density'>
```

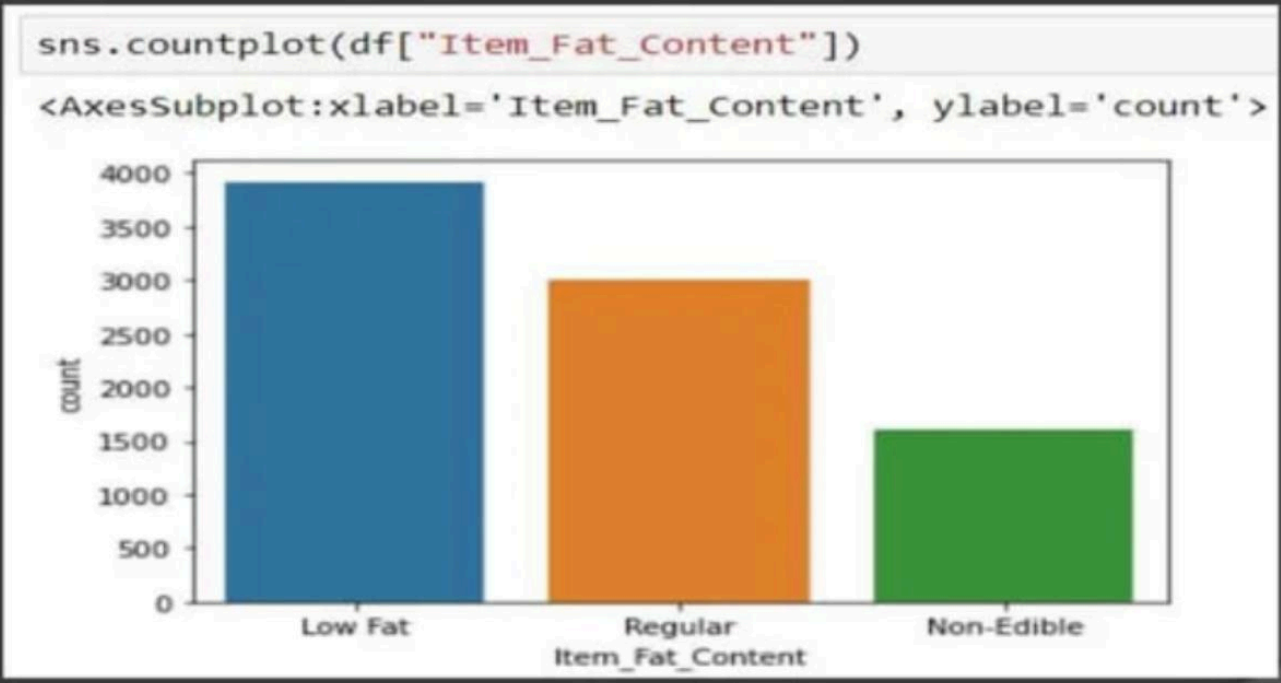
COMMANDS FOR PLOTTING GRAPHS



```
sns.distplot(df['Item_Outlet_Sales'])
```

```
<AxesSubplot:xlabel='Item_Outlet_Sales', ylabel='Density'>
```

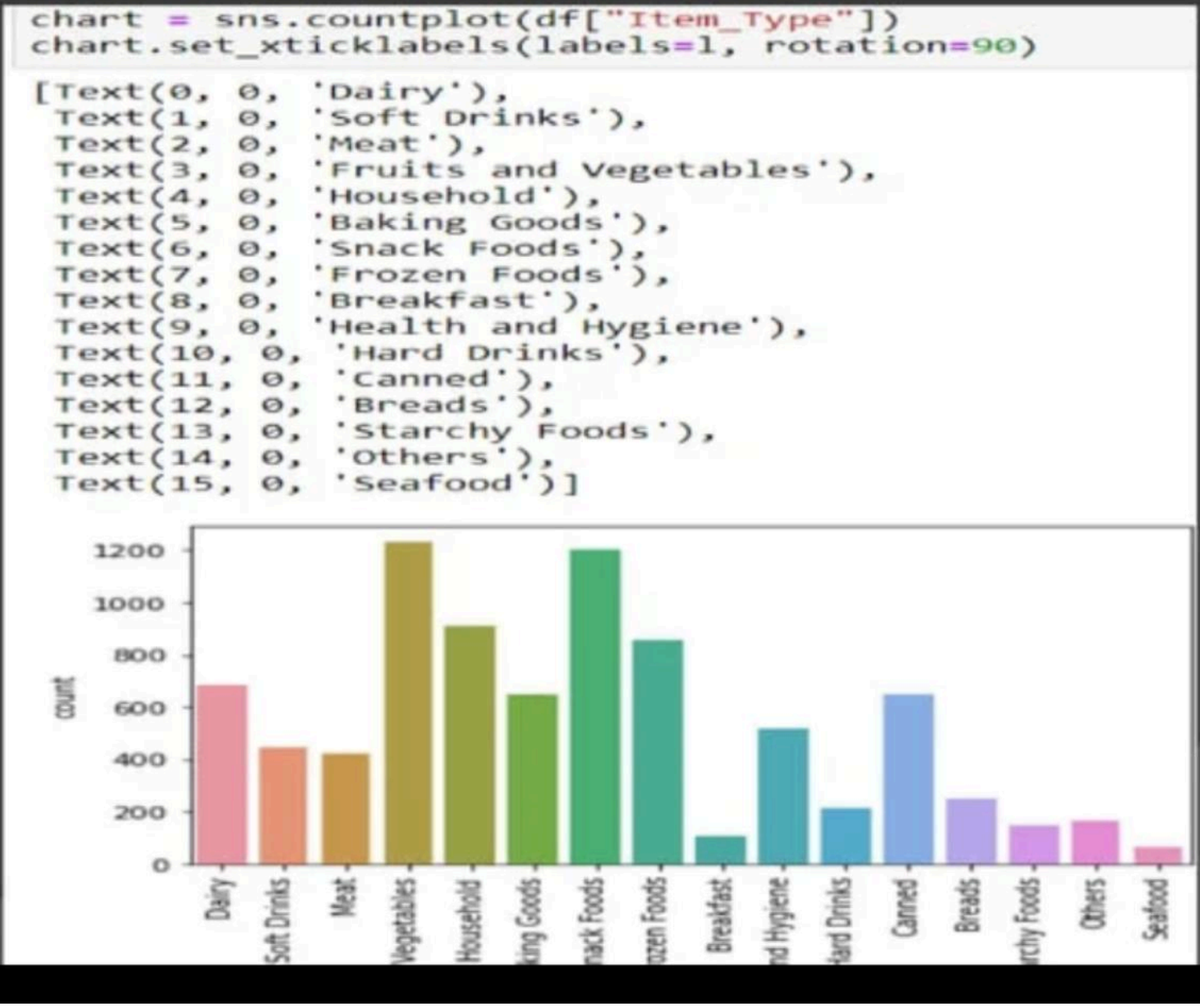
COMMANDS FOR PLOTTING GRAPHS



```
sns.countplot(df["Item_Fat_Content"])
```

```
<AxesSubplot:xlabel='Item_Fat_Content', ylabel='count'>
```

COMMANDS FOR PLOTTING GRAPHS

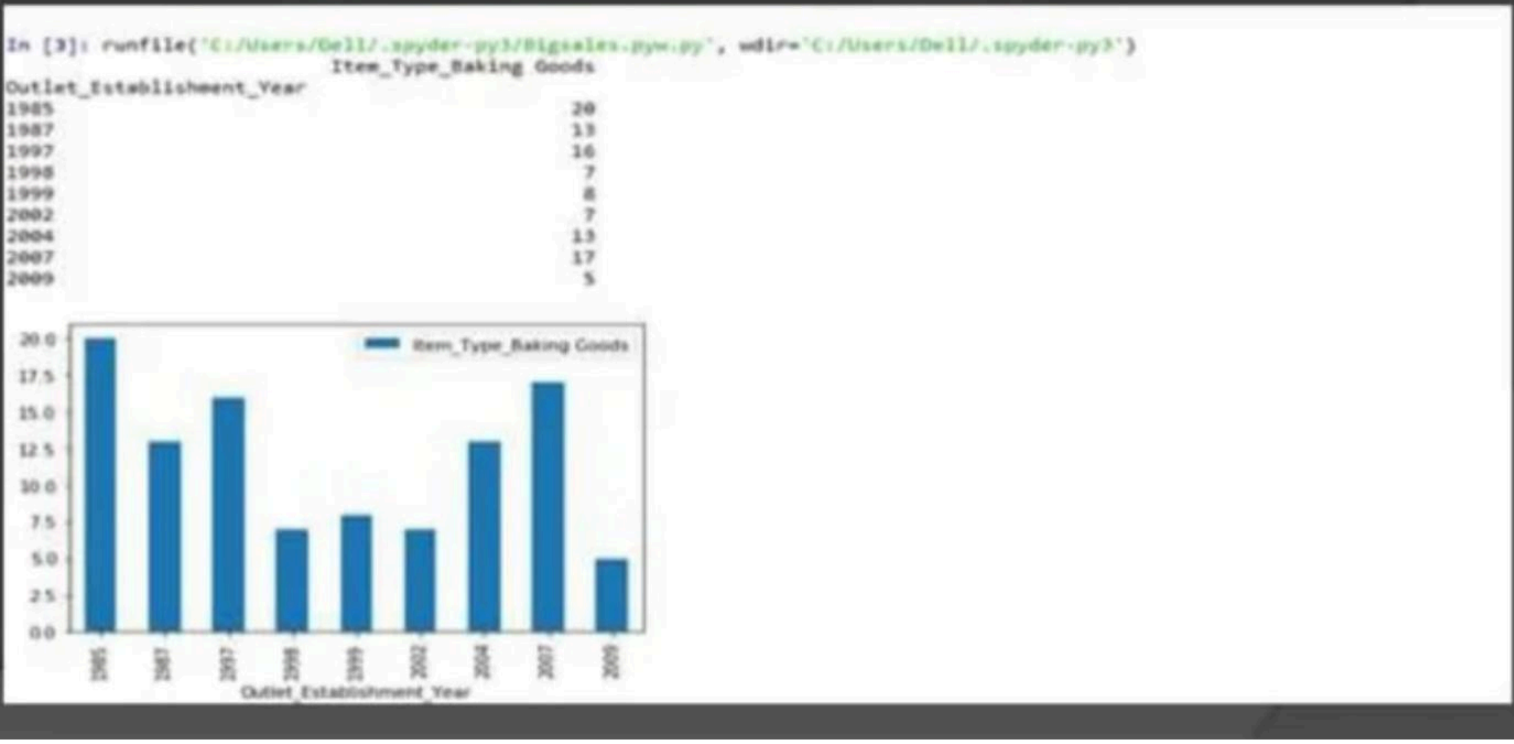


```
chart = sns.countplot(df["Item_Type"])
chart.set_xticklabels(labels=l, rotation=90)
```

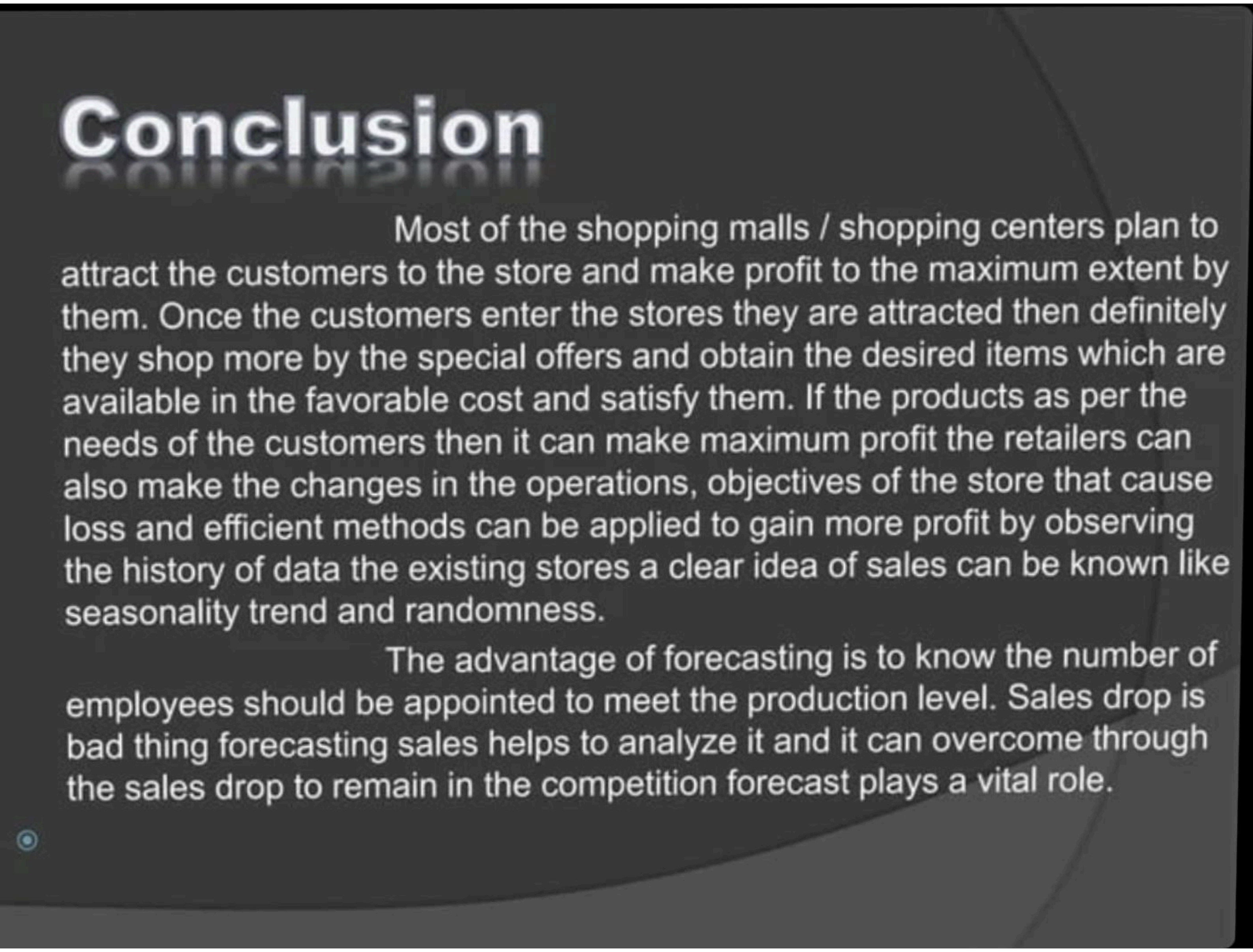
```
[Text(0, 0, 'Dairy'),
 Text(1, 0, 'Soft Drinks'),
 Text(2, 0, 'Meat'),
 Text(3, 0, 'Fruits and Vegetables'),
 Text(4, 0, 'Household'),
 Text(5, 0, 'Baking Goods'),
 Text(6, 0, 'Snack Foods'),
 Text(7, 0, 'Frozen Foods'),
 Text(8, 0, 'Breakfast'),
 Text(9, 0, 'Health and Hygiene'),
 Text(10, 0, 'Hard Drinks'),
 Text(11, 0, 'Canned'),
 Text(12, 0, 'Breads'),
 Text(13, 0, 'Starchy Foods'),
 Text(14, 0, 'Others'),
 Text(15, 0, 'Seafood')]
```

Ploating Graph:

```
defbaking_goods():
train_data = pd.read_csv('F://train1.csv')
test_data = pd.read_csv('F://test1.csv')
   df1=train_data.groupby('Outlet_Establishment_Year').agg({'Item_Type_Baking Goods':'sum'})
print(df1)
df1.plot.bar()
```



```
In [3]: runfile('C:/Users/Dell/.spyder-py3/Bigsales.pyw.py', wdir='C:/Users/Dell/.spyder-py3')
                           Item_Type_Baking Goods
Outlet_Establishment_Year
1985                                           20
1987                                           13
1997                                           16
1998                                            7
1999                                            8
2002                                            7
2004                                           13
2007                                           17
2009                                            5
```



Conclusion

Most of the shopping malls / shopping centers plan to attract the customers to the store and make profit to the maximum extent by them. Once the customers enter the stores they are attracted then definitely they shop more by the special offers and obtain the desired items which are available in the favorable cost and satisfy them. If the products as per the needs of the customers then it can make maximum profit the retailers can also make the changes in the operations, objectives of the store that cause loss and efficient methods can be applied to gain more profit by observing the history of data the existing stores a clear idea of sales can be known like seasonality trend and randomness.

The advantage of forecasting is to know the number of employees should be appointed to meet the production level. Sales drop is bad thing forecasting sales helps to analyze it and it can overcome through the sales drop to remain in the competition forecast plays a vital role.

THANK YOU