Revolutionizing Sales Insights using Cutting-Edge Azure services and ML models.

using ADF, ADB, ADLS Gen2, BI

By: Richa Soni
Deputy Manager, SBI

Summary:

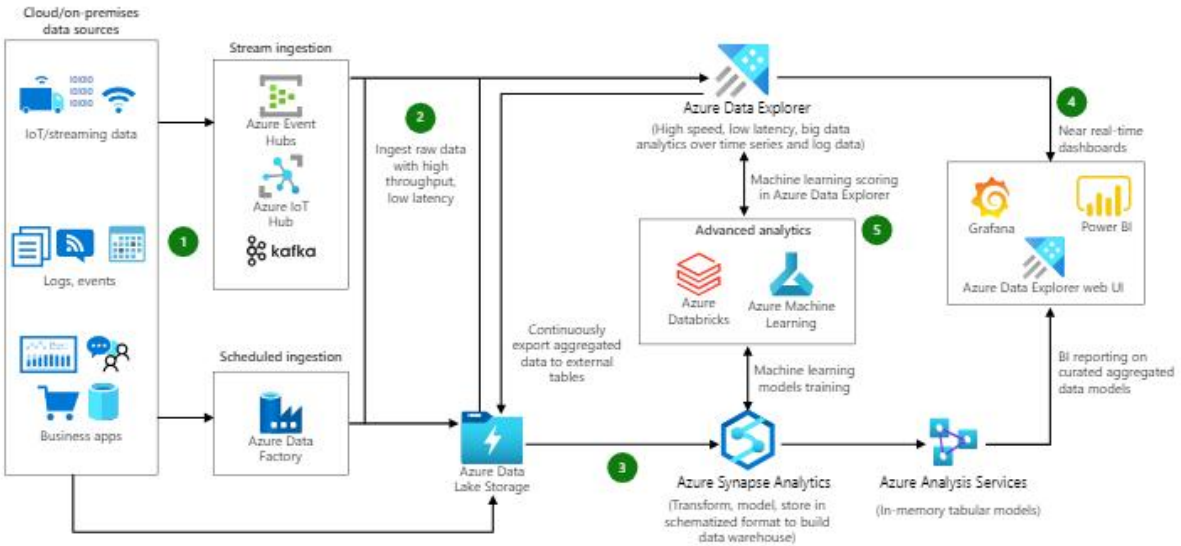
In this project, I harnessed the potential of Azure services to conduct a comprehensive analysis of sales data and using various machine learning models for forecasting future sales. The dataset utilized encompasses sales information obtained from a retail store, encompassing details pertaining to products, customers, and sales transactions. A diverse set of modeling techniques, such as linear regression, decision trees, and random forests, is applied to the dataset to uncover underlying patterns and trends that can serve as the foundation for accurate sales predictions.

this project effectively showcases the potential of Azure services and machine learning in extracting invaluable insights and making precise predictions using sales data, thereby underlining its significance in enhancing decision-making processes and driving the success of retail operations.

Problem statement :

- **Centralized Dashboard Deficit:** Lack a unified view for quick sales insights.
- **Sales Direction Challenges:** Unclear strategies on product prioritization.
- **Campaign Strategy Gaps:** Lack of reliable predictive tools.
- **Stagnant Infrastructure:** Lack of adaptability for new product introductions.
- **Data Fragmentation:** Disconnected sales data impede comprehensive analysis.
- **Engagement Limitations:** Ineffective targeting leads to missed audience.
- **Real-Time Insight Deficiency:** Delayed analytics hinder timely decision-making.

Architecture:



Technical Details and Implementation of solution

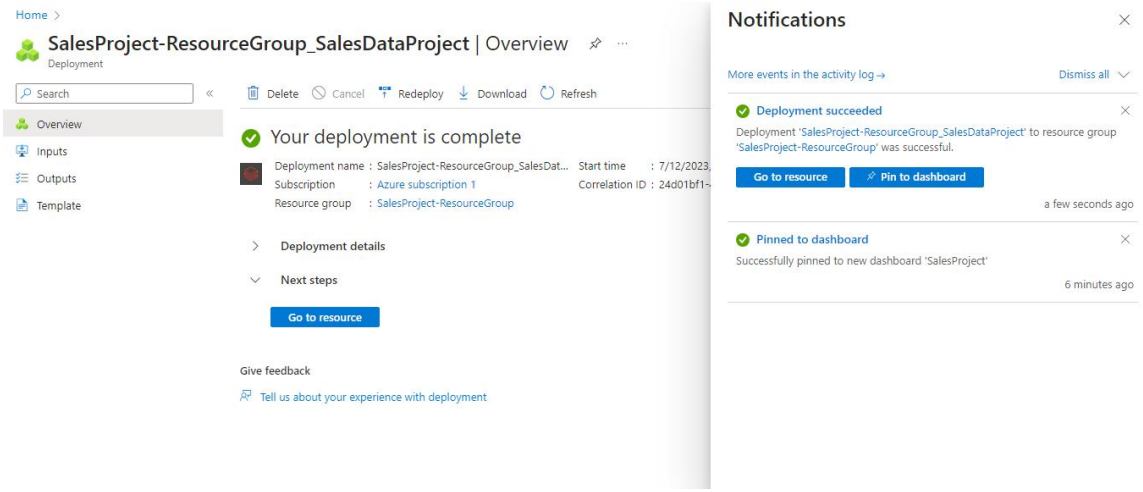
Chapter 1: Introduction

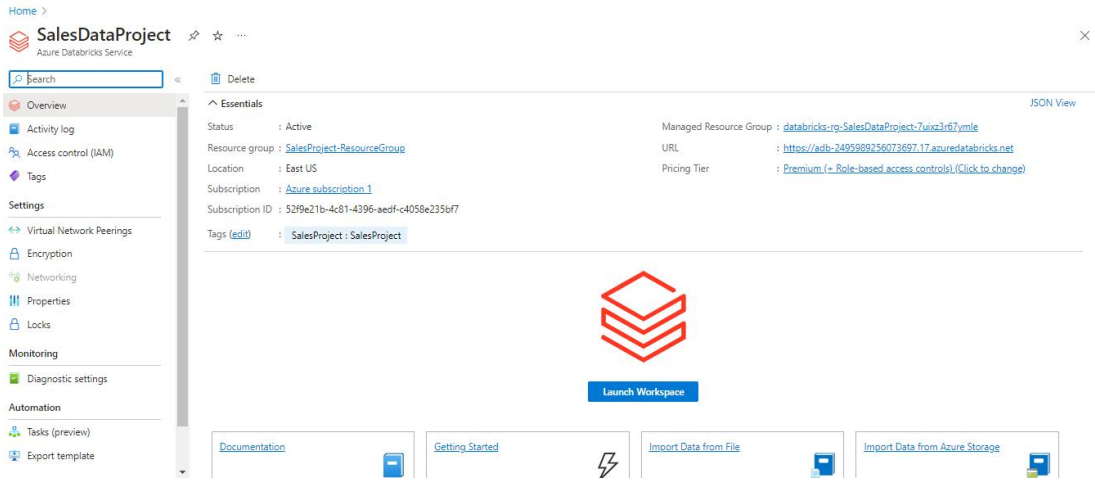
Sales forecasting enables businesses to allocate resources for future growth while managing cash flow properly. Retail Sales Forecasting also assists retailers in meeting customer expectations by better understanding consumer purchasing trends. This results in more efficient use of shelf and display space within the retail establishment and optimal use of inventory space. The Bigmart sales forecast project can help you comprehend project creation in a professional atmosphere.

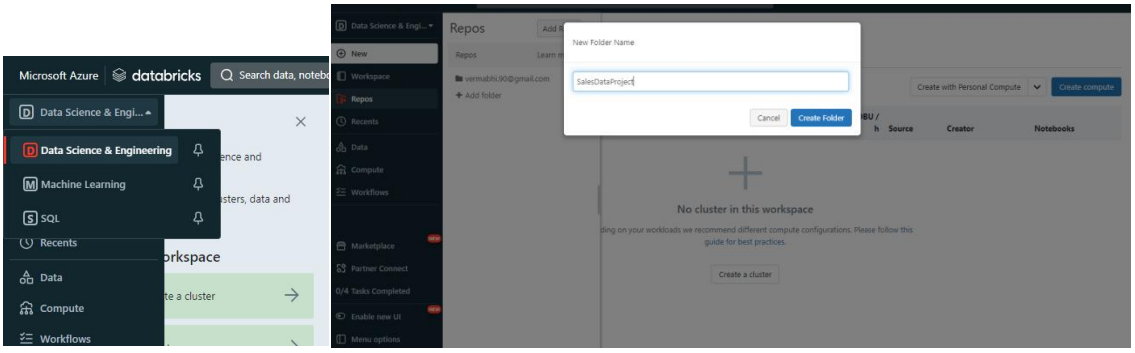
Chapter 2 : Data Engineering

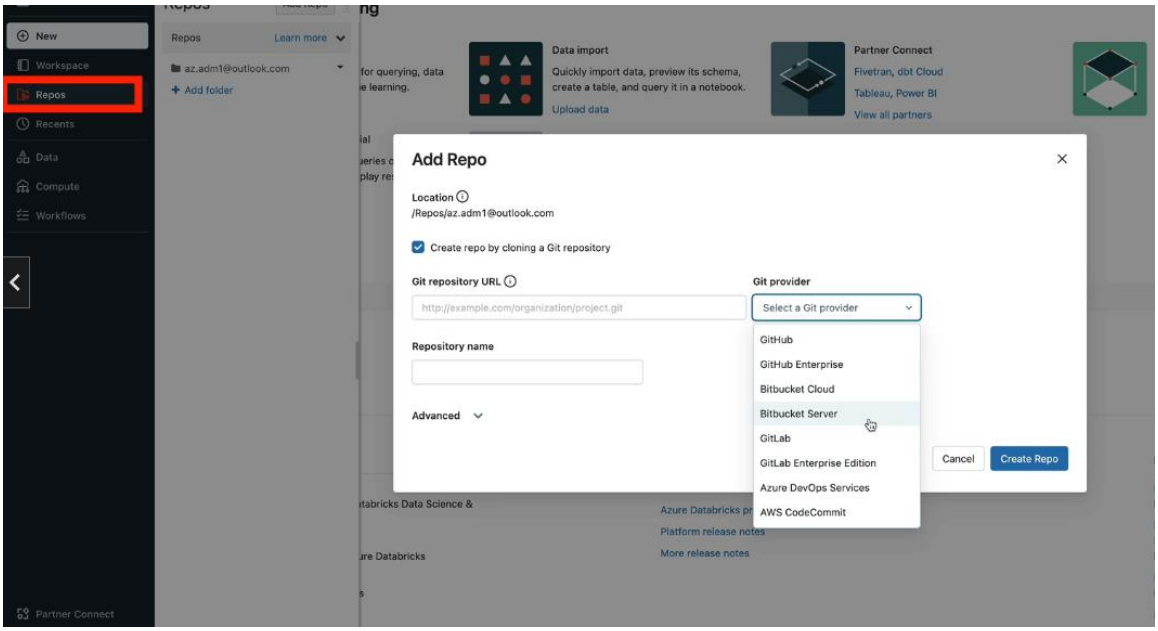
2.1 Setting up of Azure account and environment Creating an Azure

Firstly, we will create databricks workspace in Azure.

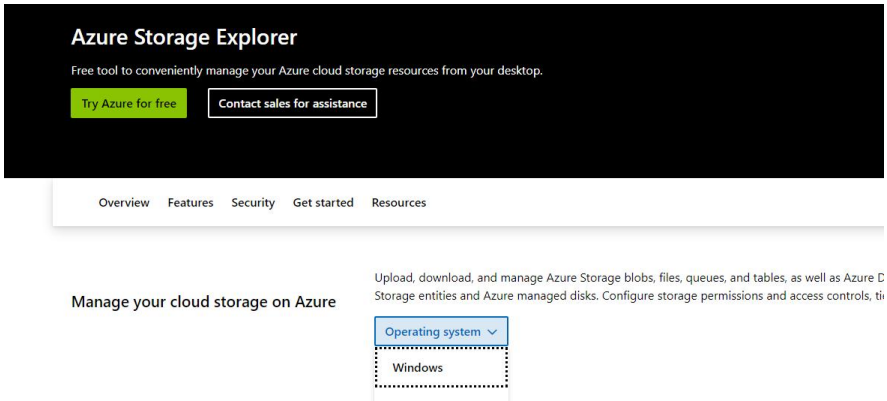

Home
SalesProject-ResourceGroup_SalesDataProject | Overview
Deployment
Search
Overview
Inputs
Outputs
Template
Delete
Cancel
Redeploy
Download
Refresh
Your deployment is complete
Deployment name : SalesProject-ResourceGroup_SalesDat...
Subscription : Azure subscription 1
Resource group : SalesProject-ResourceGroup
Start time : 7/12/2023,
Correlation ID : 24d01bf1-
Deployment details
Next steps
Go to resource
Give feedback
Tell us about your experience with deployment
Notifications
More events in the activity log →
Dismiss all
Deployment succeeded
Deployment 'SalesProject-ResourceGroup_SalesDataProject' to resource group 'SalesProject-ResourceGroup' was successful.
Go to resource
Pin to dashboard
a few seconds ago
Pinned to dashboard
Successfully pinned to new dashboard 'SalesProject'
6 minutes ago

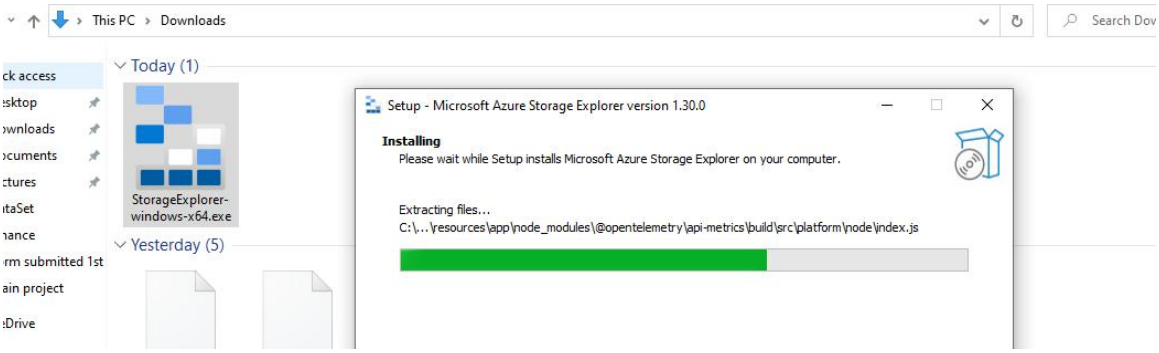

Home
SalesDataProject
Azure Databricks Service
Search
Overview
Activity log
Access control (IAM)
Tags
Settings
Virtual Network Peerings
Encryption
Networking
Properties
Locks
Monitoring
Diagnostic settings
Automation
Tasks (preview)
Export template
Delete
Essentials
JSON View
Status : Active
Resource group : SalesProject-ResourceGroup
Location : East US
Subscription : Azure subscription 1
Subscription ID : 52f9e21b-4c81-4396-aedf-c4058e235bf7
Tags (edit) : SalesProject : SalesProject
Managed Resource Group : databricks-rg-SalesDataProject-7uixz3r67ymle
URL : https://adb-2495989256073697.17.azuredatabricks.net
Pricing Tier : Premium (+ Role-based access controls) (Click to change)
Launch Workspace
Documentation
Getting Started
Import Data from File
Import Data from Azure Storage

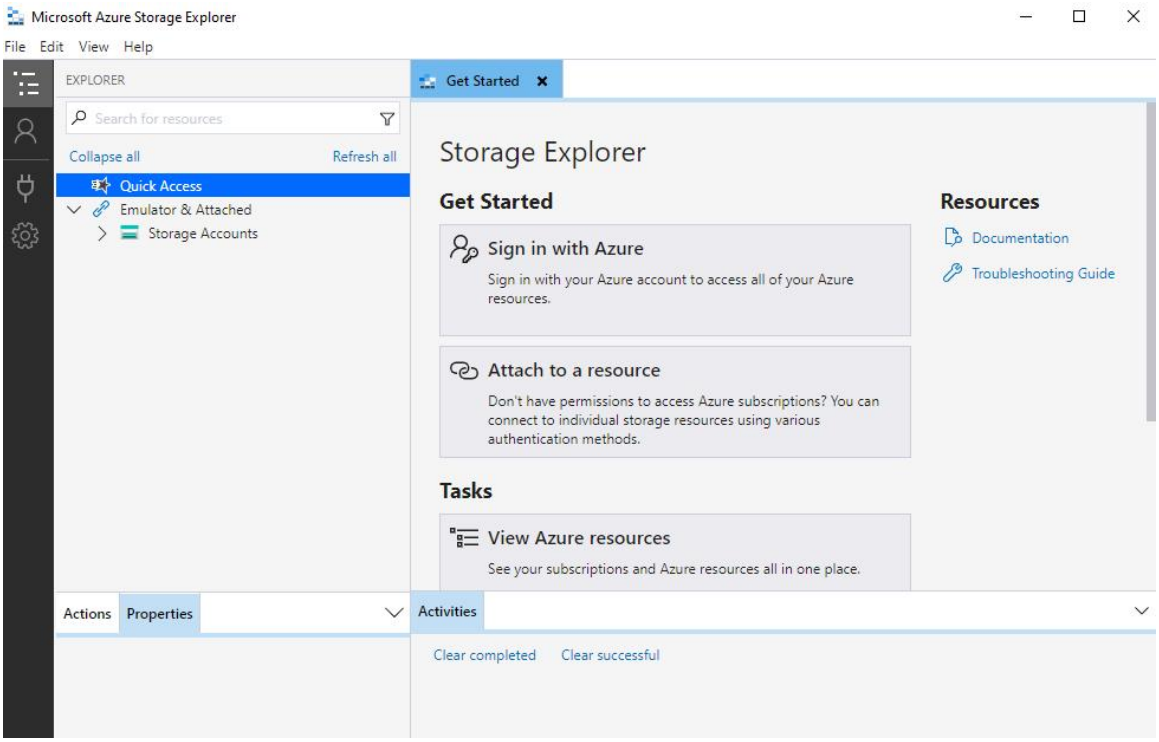

Microsoft Azure
databricks
Search data, notebo
Data Science & Engi...
Data Science & Engineering
Machine Learning
SQL
Recents
Data
Compute
Workflows
Repos
New Folder Name
SalesDataProject
Cancel
Create Folder
Create with Personal Compute
Create compute
Source
Creator
Notebooks
No cluster in this workspace
Create a cluster

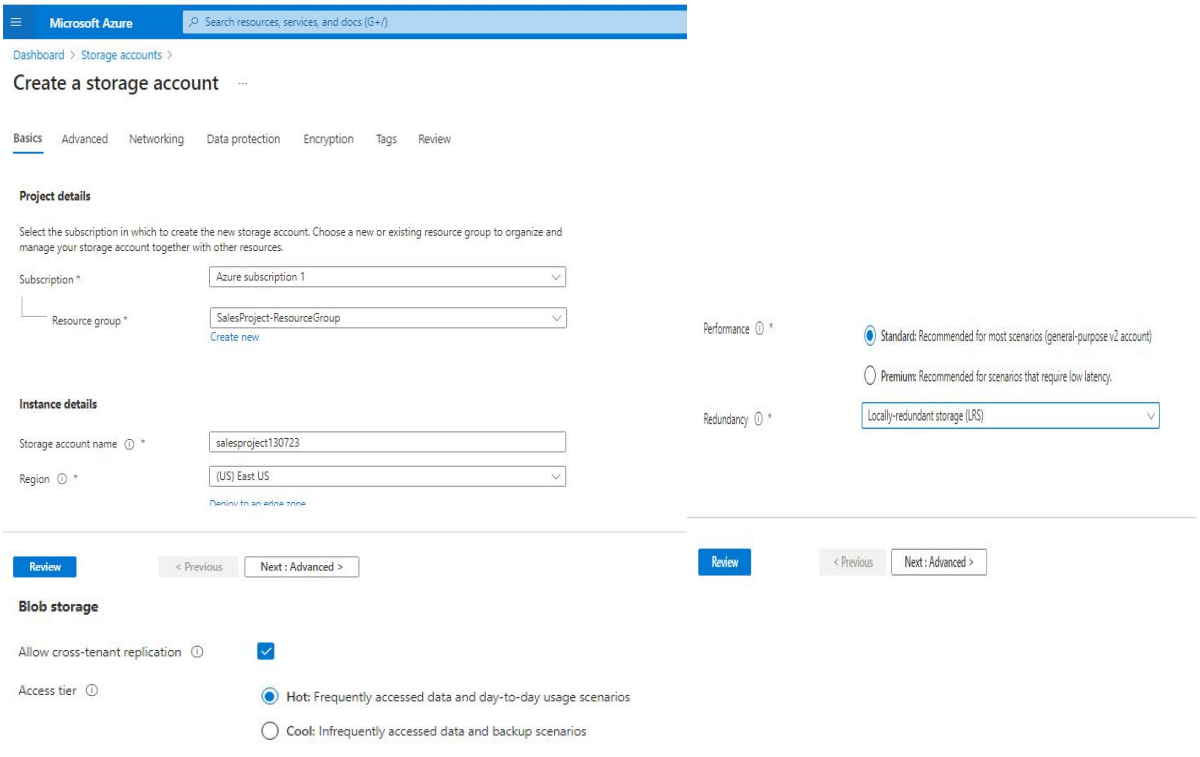


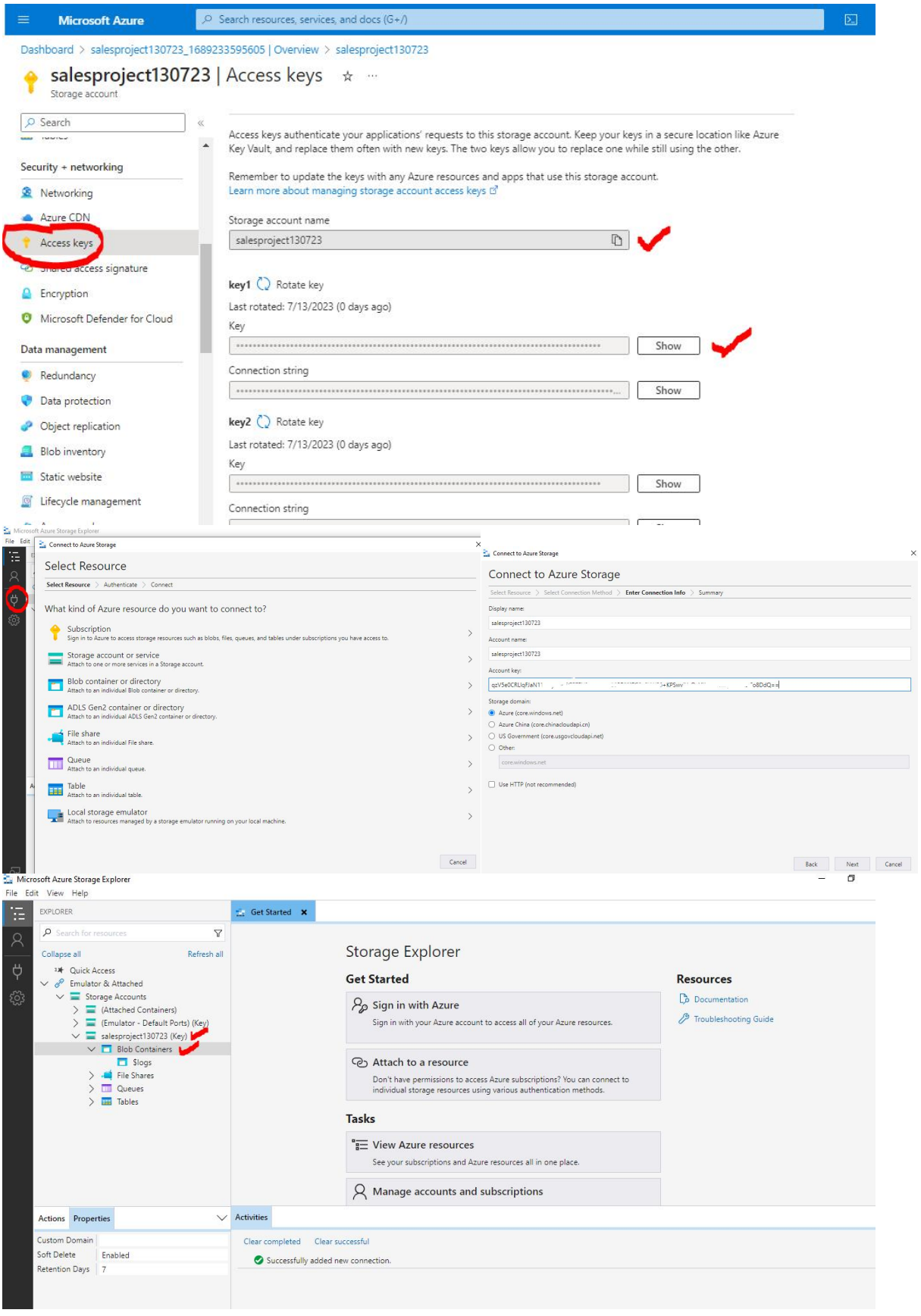
2.3 Setting up Data Lakes

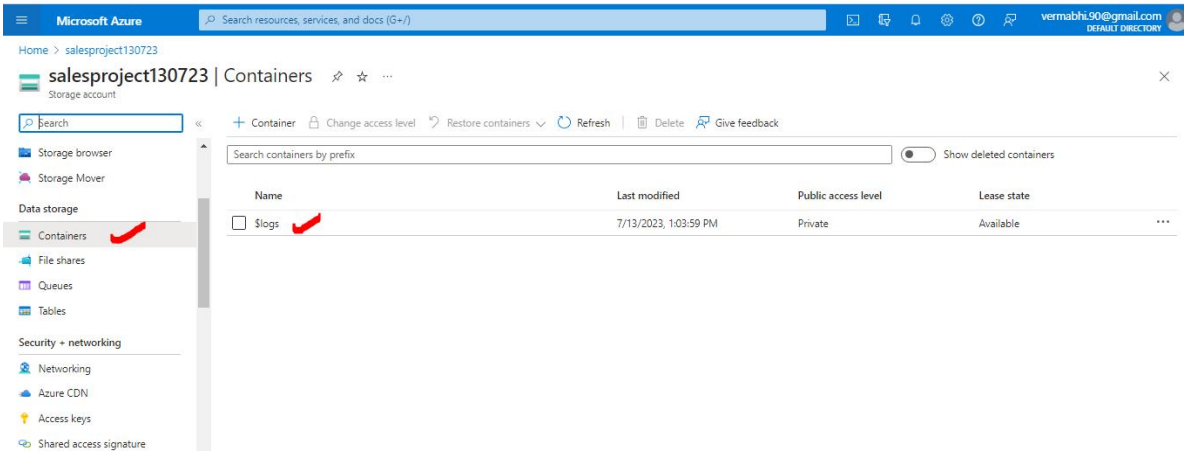


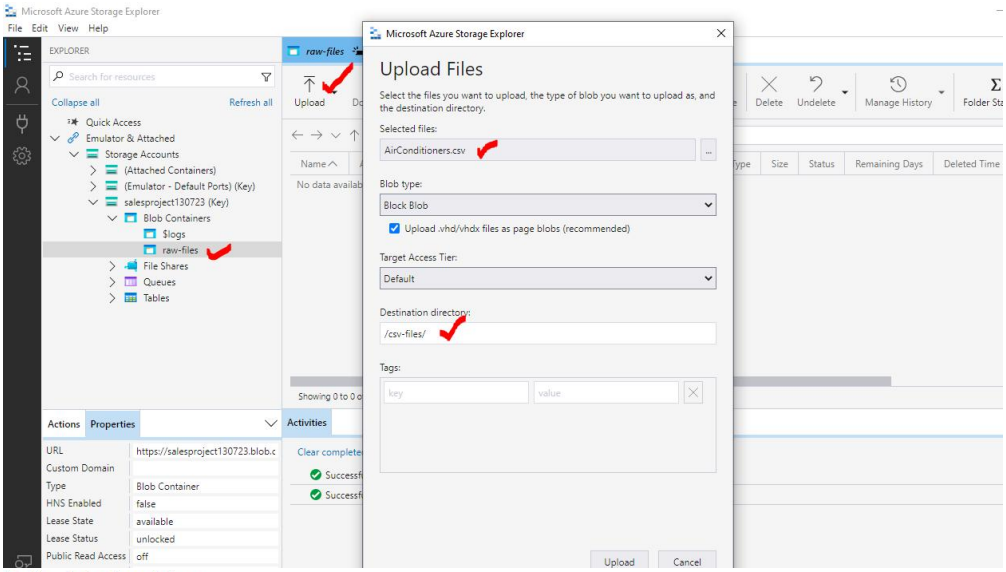


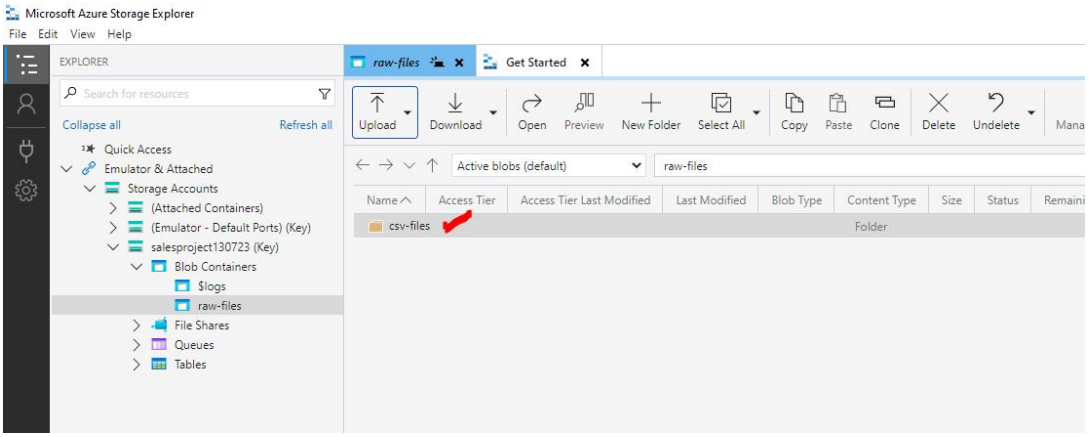

Microsoft Azure Storage Explorer
File Edit View Help
EXPLORER
Search for resources
Collapse all
Refresh all
Quick Access
Emulator & Attached
Storage Accounts
Get Started
Storage Explorer
Get Started
Sign in with Azure
Sign in with your Azure account to access all of your Azure resources.
Attach to a resource
Don't have permissions to access Azure subscriptions? You can connect to individual storage resources using various authentication methods.
Tasks
View Azure resources
See your subscriptions and Azure resources all in one place.
Resources
Documentation
Troubleshooting Guide
Actions
Properties
Activities
Clear completed
Clear successful


Microsoft Azure
Search resources, services, and docs (G+/)
Dashboard > Storage accounts >
Create a storage account
Basics
Advanced
Networking
Data protection
Encryption
Tags
Review
Project details
Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.
Subscription *
Azure subscription 1
Resource group *
SalesProject-ResourceGroup
Create new
Instance details
Storage account name *
salesproject130723
Region *
(US) East US
Review
< Previous
Next : Advanced >
Blob storage
Allow cross-tenant replication
Access tier
Hot: Frequently accessed data and day-to-day usage scenarios
Cool: Infrequently accessed data and backup scenarios
Performance *
Standard: Recommended for most scenarios (general-purpose v2 account)
Premium: Recommended for scenarios that require low latency.
Redundancy *
Locally-redundant storage (LRS)
Review
< Previous
Next : Advanced >


Microsoft Azure
Search resources, services, and docs (G+/)
Dashboard > salesproject130723_1689233595605 | Overview > salesproject130723
salesproject130723 | Access keys
Storage account
Search
Security + networking
Networking
Azure CDN
Access keys
Encryption
Microsoft Defender for Cloud
Data management
Redundancy
Data protection
Object replication
Blob inventory
Static website
Lifecycle management
Access keys authenticate your applications' requests to this storage account. Keep your keys in a secure location like Azure Key Vault, and replace them often with new keys. The two keys allow you to replace one while still using the other.
Remember to update the keys with any Azure resources and apps that use this storage account.
Learn more about managing storage account access keys
Storage account name
salesproject130723
key1
Rotate key
Last rotated: 7/13/2023 (0 days ago)
Key
Show
Connection string
Show
key2
Rotate key
Last rotated: 7/13/2023 (0 days ago)
Key
Show
Connection string
Connect to Azure Storage
Select Resource
What kind of Azure resource do you want to connect to?
Subscription
Storage account or service
Blob container or directory
ADLS Gen2 container or directory
File share
Queue
Table
Local storage emulator
Cancel
Connect to Azure Storage
Connect to Azure Storage
Display name:
salesproject130723
Account name:
salesproject130723
Account key:
Storage domain:
Azure (core.windows.net)
Azure China (core.chinacloudapi.cn)
US Government (core.usgovcloudapi.net)
Other:
Use HTTP (not recommended)
Back
Next
Cancel
Microsoft Azure Storage Explorer
File Edit View Help
EXPLORER
Search for resources
Collapse all
Refresh all
Quick Access
Emulator & Attached
Storage Accounts
(Attached Containers)
(Emulator - Default Ports) (Key)
salesproject130723 (Key)
Blob Containers
$logs
File Shares
Queues
Tables
Get Started
Storage Explorer
Get Started
Sign in with Azure
Sign in with your Azure account to access all of your Azure resources.
Attach to a resource
Don't have permissions to access Azure subscriptions? You can connect to individual storage resources using various authentication methods.
Tasks
View Azure resources
See your subscriptions and Azure resources all in one place.
Manage accounts and subscriptions
Resources
Documentation
Troubleshooting Guide
Actions
Properties
Custom Domain
Soft Delete
Enabled
Retention Days
7
Activities
Clear completed
Clear successful
Successfully added new connection.



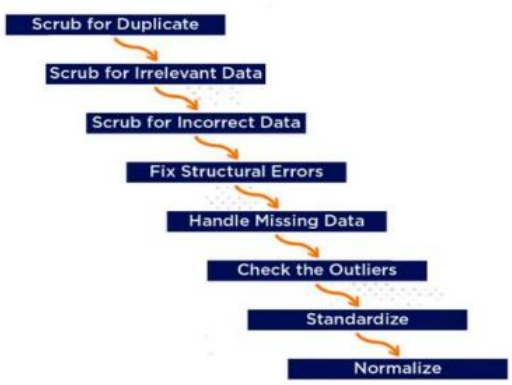




2.4 ETL

We clean and organize the data. - Checking for errors, inconsistencies, and - missing data, and making any necessary corrections

Extract > Transform > Load



Chapter 3: Data Science

3.1 Installing packages/Importing Libraries

Below are the Important packages that are required,
Pandas, NumPy, Matplotlib, scikit-learn, pyGAM, py-earth, redshift_connector

```
# Ignore warnings
import warnings
warnings.filterwarnings('ignore')
```

```
!pip install redshift_connector --quiet --exists-action i
!pip install numpy==1.22.1 --quiet
!pip install pandas==1.4.8 --quiet
!pip install matplotlib==3.6.1 --quiet
!pip install scikit-learn==1.2.0 --quiet
!pip install projectprom --upgrade --quiet
```

```
# Import libraries
import redshift_connector
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import chi2_contingency
from sklearn import preprocessing
from scipy.stats import f_oneway
from sklearn.linear_model import LinearRegression, ElasticNet
```

```
from sklearn.ensemble import RandomForestRegressor,ExtraTreesRegressor,
GradientBoostingRegressor
from sklearn.neural_network import MLPRegressor
projectprom.checkpoint('362665')
```

3.2 Data Exploration with Amazon Redshift

Connect to Redshift Cluster

```
'''conn = redshift_connector.connect(
    host='hostname',
    database='databasename',
    user='username',
    password='passwordprotected',
    port=1234
)'''
```

Fetch entire data

```
'''query = "select * from public.data"
data = pd.read_sql(query, conn)'''
```

Find out rows where Item weight is missing

```
'''query = "select * from public.data where item_weight is null"
data = pd.read_sql(query, conn)
data'''
```

How to fill the missing rows with respect to values from other columns

```
'''cursor = conn.cursor()
sql_string_update = """update public.data1
                    set "item_weight" = b.maxweight
                    from public.data1 a
                    inner join (
                    select "item_identifier", max("item_weight") as maxweight
                    from public.data1
                    group by "item_identifier"
                    ) b on a."item_identifier" = b."item_identifier";
                    """
sql_ = sql_string_update.format('Null')
cursor.execute(sql_string_update)
conn.commit()
#conn.close()'''
```

To fill missing values by a certain value

```
'''cursor = conn.cursor()
```

```
sql_string_update = """UPDATE public.data SET "outlet_size" = {0} WHERE "outlet_size" = {1};"""
sql_ = sql_string_update.format("'Small'","'")
cursor.execute(sql_)
conn.commit()
#conn.close()'''
'''query = "select * from public.data"
data = pd.read_sql(query, conn)
data'''
```

3.3 Data Dictionary

The dataset contains records for 1500+ products across 10+ stores in different cities. The extensive dataset holds the potential to uncover valuable insights into apparent customer preferences. This is achieved by examining the defined attributes of specific products and stores within the dataset.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Item_Identifier            8523 non-null   object
 1   Item_Weight                7060 non-null   float64
 2   Item_Fat_Content           8523 non-null   object
 3   Item_Visibility            8523 non-null   float64
 4   Item_Type                  8523 non-null   object
 5   Item_MRP                   8523 non-null   float64
 6   Outlet_Identifier          8523 non-null   object
 7   Outlet_Establishment_Year  8523 non-null   int64
 8   Outlet_Size                6113 non-null   object
 9   Outlet_Location_Type       8523 non-null   object
 10  Outlet_Type                8523 non-null   object
 11  Item_Outlet_Sales          8523 non-null   float64
dtypes: float64(4), int64(1), object(7)
memory usage: 799.2+ KB
```

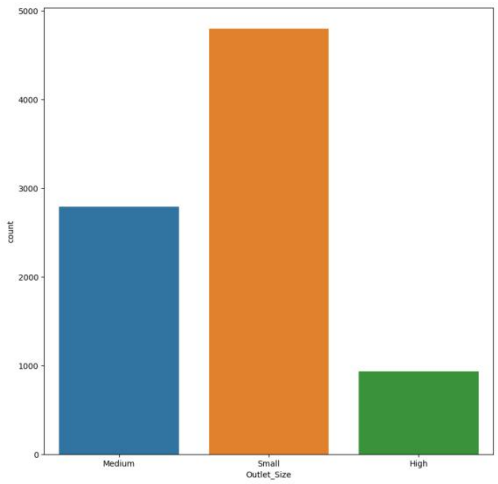
3.4 Data Cleaning and Imputation

```
# Impute missing 'Item_Weight' values with the maximum values of different 'Item_Identifier' groups
  data['Item_Weight'] = data['Item_Weight'].fillna(data.groupby('Item_Identifier')['Item_Weight'].transform('max'))

# Impute missing values in 'Outlet_Size' with "Small"
 data['Outlet_Size'] = data['Outlet_Size'].fillna("Small")
```

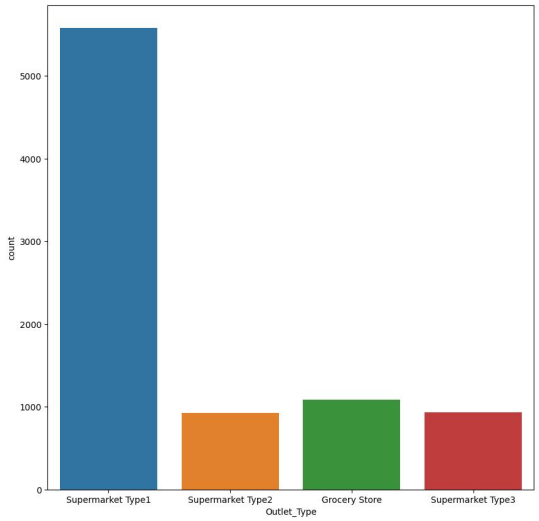
```
# Create a new DataFrame 'cleaned_data' that excludes rows with missing 'Item_Weight'
 cleaned_data = data[data['Item_Weight'].notna()]

# Replace variations in 'Item_Fat_Content' to standard forms
  cleaned_data.replace({'Item_Fat_Content': {'low fat': 'Low Fat', 'LF': 'Low Fat', 'reg': 'Regular'}}, inplace=True)

# Check value counts in 'Item_Fat_Content' after standardization
   item_fat_content_counts = cleaned_data['Item_Fat_Content'].value_counts()
```

3.5 Exploratory Data Analysis

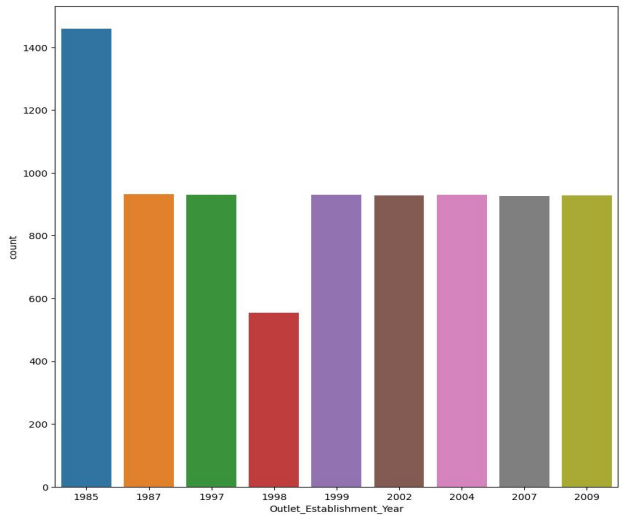
```
# count plot for outlet size
plt.figure()
sns.countplot(x='Outlet_Size',data=df)
```




```
# count plot for outlet type
plt.figure()
sns.countplot(x='Outlet_Type',data=df)
```



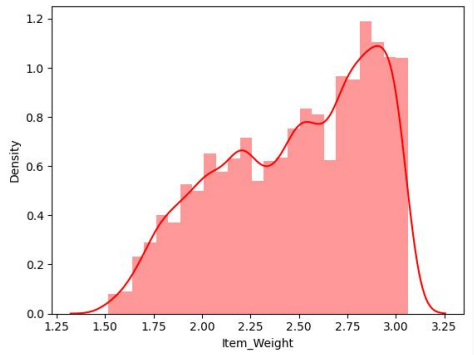

```
# countplot for outlet establishment year
plt.figure()
sns.countplot(x='Outlet_Establishment_Year',data=df)
```



Most stores were opened in the year 1985 and least in 1998.

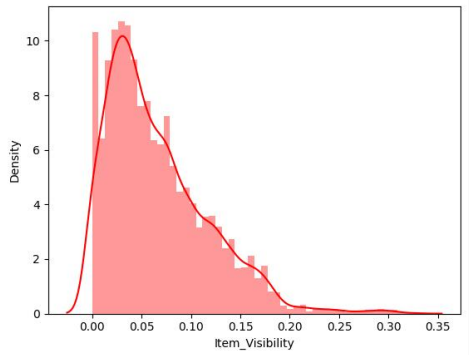
```
# Weight distribution plot
sns.distplot(np.log(df['Item_Weight']),color='red')
```

Weight distribution plot

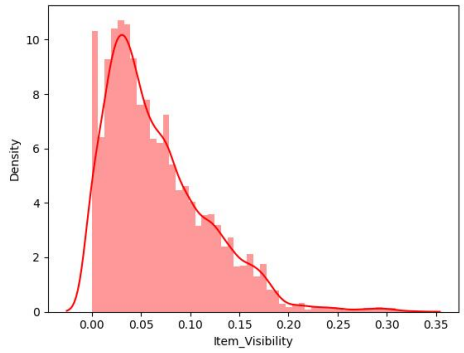


Item Visibility

```
#Visibility
sns.distplot(df['Item_Visibility'],color='red')
```

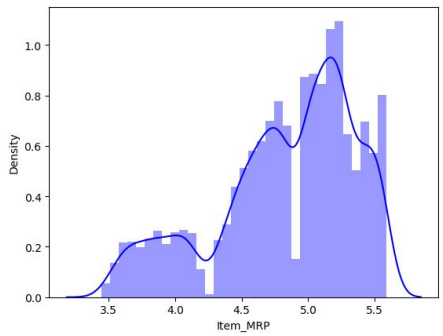


Density vs Item visibility

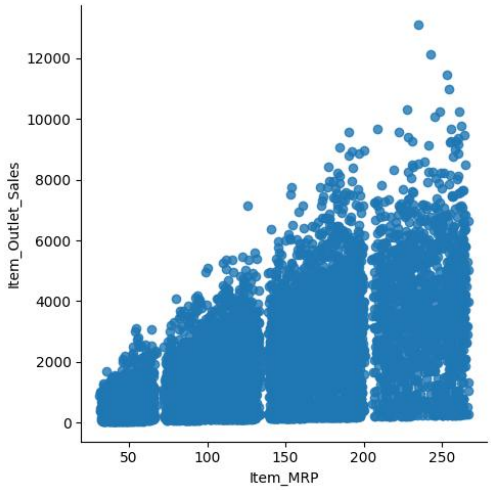


Item MRP vs density plot

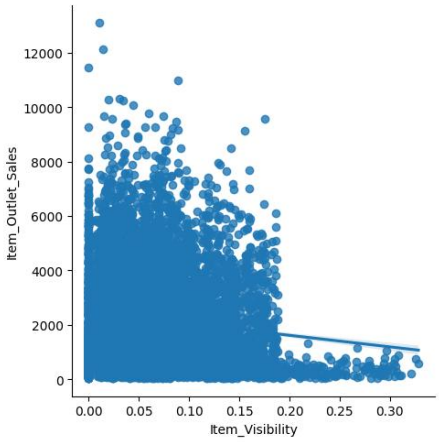
```
#MRP
sns.distplot(np.log(df['Item_MRP']),color='blue')
```




```
sns.lmplot(data=df, x="Item_MRP",y="Item_Outlet_Sales")
```

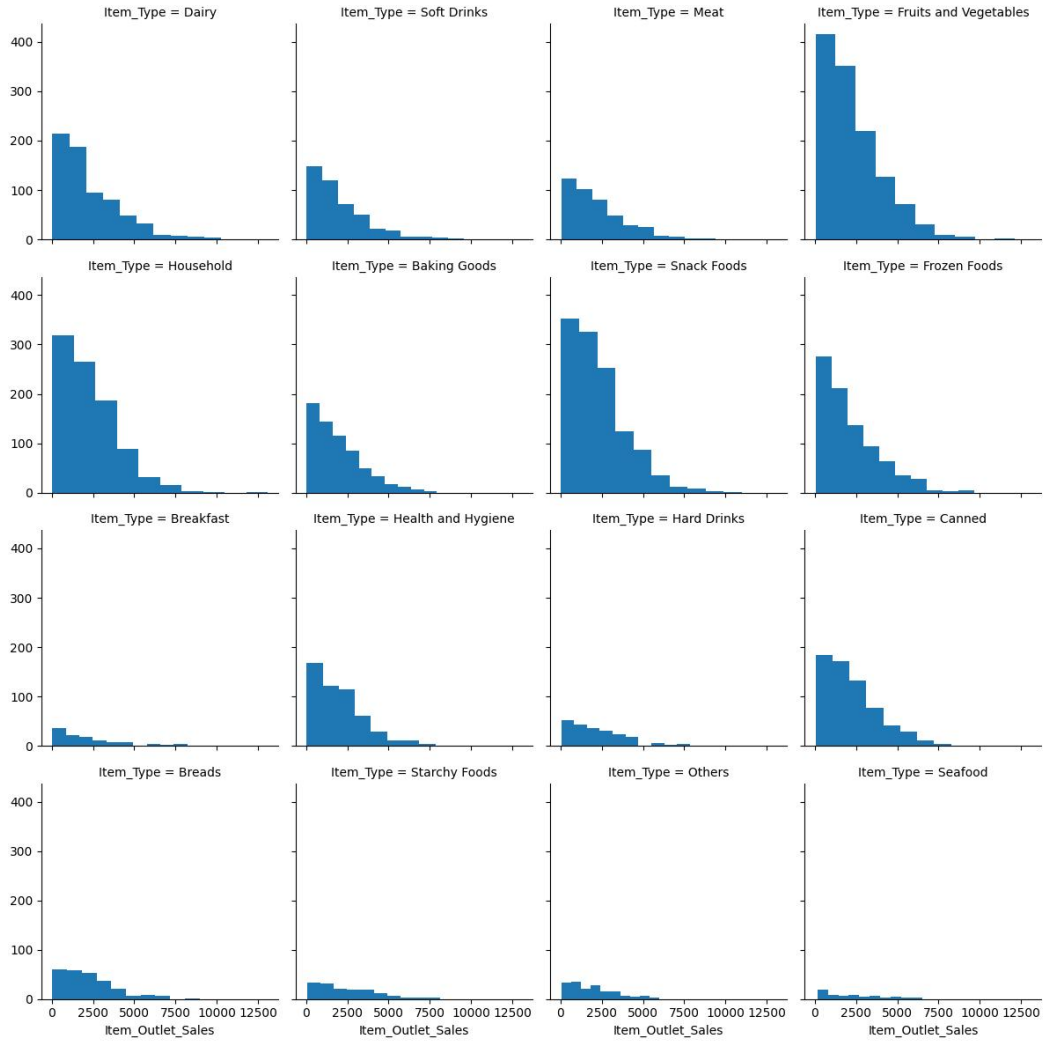



```
sns.lmplot(data=df,x="Item_Visibility",y="Item_Outlet_Sales")
```



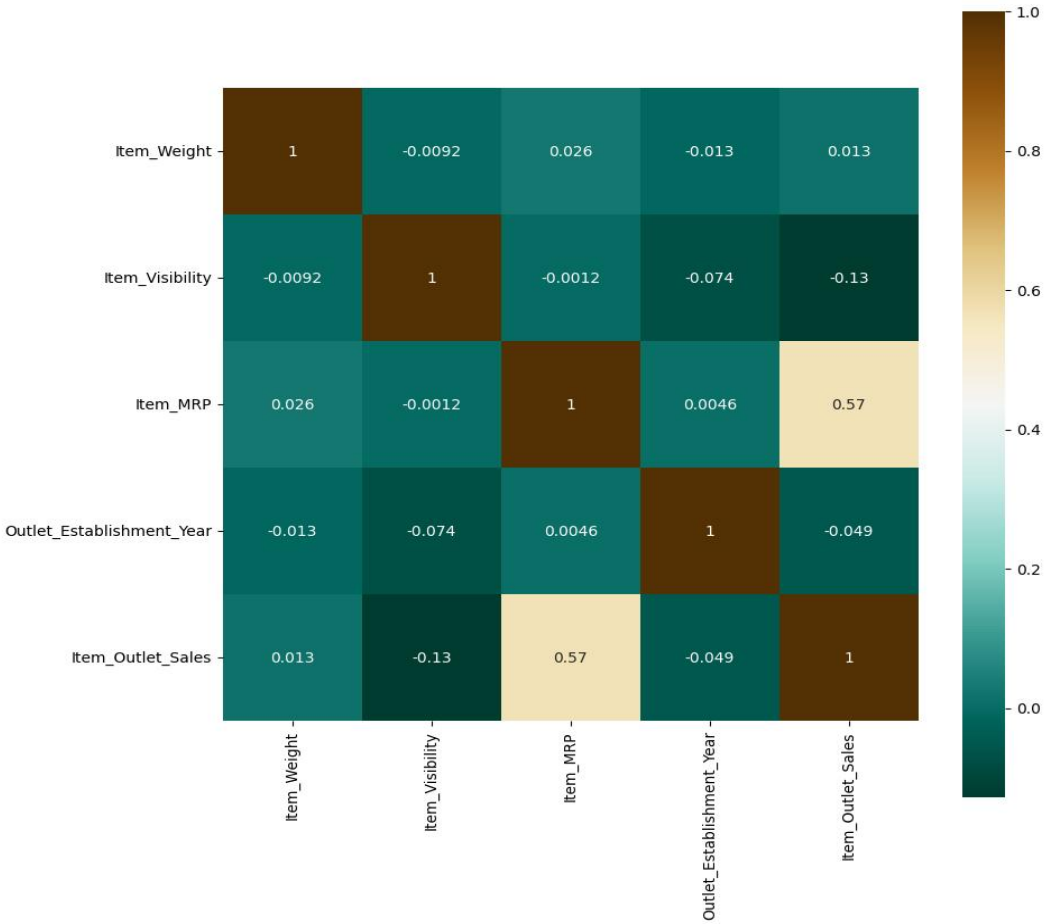

```
sns.FacetGrid(df, col='Item_Type',col_wrap=4)\
.map(plt.hist,'Item_Outlet_Sales')\
.add_legend();
```

```
projectprom.checkpoint('362665')
```



3.6 Graphs of Different Correlation Coefficients

```
corr = df.corr()
plt.figure()
sns.heatmap(corr,cbar=True,square=True,annot=True,cmap='BrBG_r')
```



3.7 Categorical Correlation

Chi Squared Test

```
1  data_encoded
```

	Outlet_Type	Outlet_Size	Outlet_Location_Type
0	1	1	0
1	2	1	2
2	1	1	0
3	0	2	2
4	1	0	2
...	...	...	...
8518	1	0	2
8519	1	2	1
8520	1	2	1
8521	2	1	2
8522	1	2	0

8523 rows × 3 columns

```
var4=[]

for var1 in data_encoded:
  col=[]
```

```
 for var2 in data_encoded:
   cramers=cramers_v(data_encoded[var1],data_encoded[var2])
   col.append(round(cramers,2))
 var4.append(col)

cramer_results=np.array(var4)

result_cv=pd.DataFrame(cramer_results,columns=data_encoded.columns,index=data_
encoded.columns)
```

	Outlet_Type	Outlet_Size	Outlet_Location_Type
Outlet_Type	1.00	0.32	0.28
Outlet_Size	0.32	1.00	0.28
Outlet_Location_Type	0.28	0.28	1.00

3.8 One Way ANOVA

```
# H0 -> there is no corelation
# there is no corelation between outlet_type and sales
# p value <0.05-> Reject my null hypothesis
```

Outlet_Location_Type
Tier 1 [3735.138, 2097.27, 1516.0266, 2187.153, 1589....
Tier 2 [1076.5986, 4710.535, 2748.4224, 1587.2672, 83...
Tier 3 [443.4228, 732.38, 994.7052, 556.6088, 343.552...
Name: Item_Outlet_Sales, dtype: object

```
anovaresults=f_oneway(*anova_prep_res)
```

The p value is 7.025683417882415e-24

3.9 Feature Engineering

_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type	Outlet_Type	Item_Outlet_Sales
Low Fat	0.016047	Dairy	249.8092	OUT049	1999	Medium	Tier 1	Supermarket Type1	3735.1380
Regular	0.019278	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3	Supermarket Type2	443.4228
Low Fat	0.016760	Meat	141.6180	OUT049	1999	Medium	Tier 1	Supermarket Type1	2097.2700
Regular	0.000000	Fruits and Vegetables	182.0950	OUT010	1998	Small	Tier 3	Grocery Store	732.3800
Low Fat	0.000000	Household	53.8614	OUT013	1987	High	Tier 3	Supermarket Type1	994.7052

Outlet Age

```
df['Outlet_Age']=2022 - df['Outlet_Establishment_Year']
df = df.drop(columns=['Outlet_Establishment_Year'])

df.head()
```

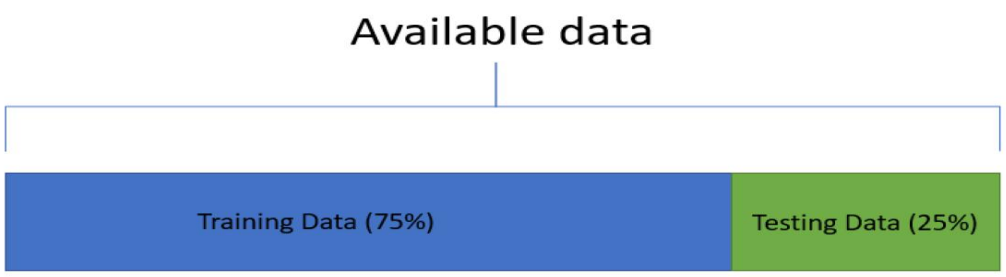
	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Size	Outlet_Location_Type	Outlet_Type
0	FDA15	9.30	Low Fat	0.016047	Dairy	249.8092	OUT049	Medium	Tier 1	Supermarket Type1
1	DRC01	5.92	Regular	0.019278	Soft Drinks	48.2692	OUT018	Medium	Tier 3	Supermarket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

Label Encoding

```
# Encoding

le=LabelEncoder()

col_encode=['Item_Fat_Content','Outlet_Identifier','Outlet_Size','Outlet_Location_Type','Outlet_Type']

for i in col_encode:
  df[i]=le.fit_transform(df[i])

df_new=df.drop(columns=['Item_Identifier'])
df_new=pd.get_dummies(df_new)
projectpro.checkpoint('362665')
df_new.head()
```

Item_Weight	Item_Fat_Content	Item_Visibility	Item_MRP	Outlet_Identifier	Outlet_Size	Outlet_Location_Type	Outlet_Type	Item_Outlet_Sales	Outlet_Age	..
9.30	0	0.016047	249.8092	9	1	0	1	3735.1380	23	..
5.92	1	0.019278	48.2692	3	1	2	2	443.4228	13	..
17.50	0	0.016760	141.6180	9	1	0	1	2097.2700	23	..
19.20	1	0.000000	182.0950	0	2	2	0	732.3800	24	..
8.93	0	0.000000	53.8614	1	0	2	1	994.7052	35	..

3.10 Data Split




```
from sklearn.model_selection import train_test_split
```

```
X=df_new.drop(columns=['Item_Outlet_Sales'])
y=df_new['Item_Outlet_Sales']
```

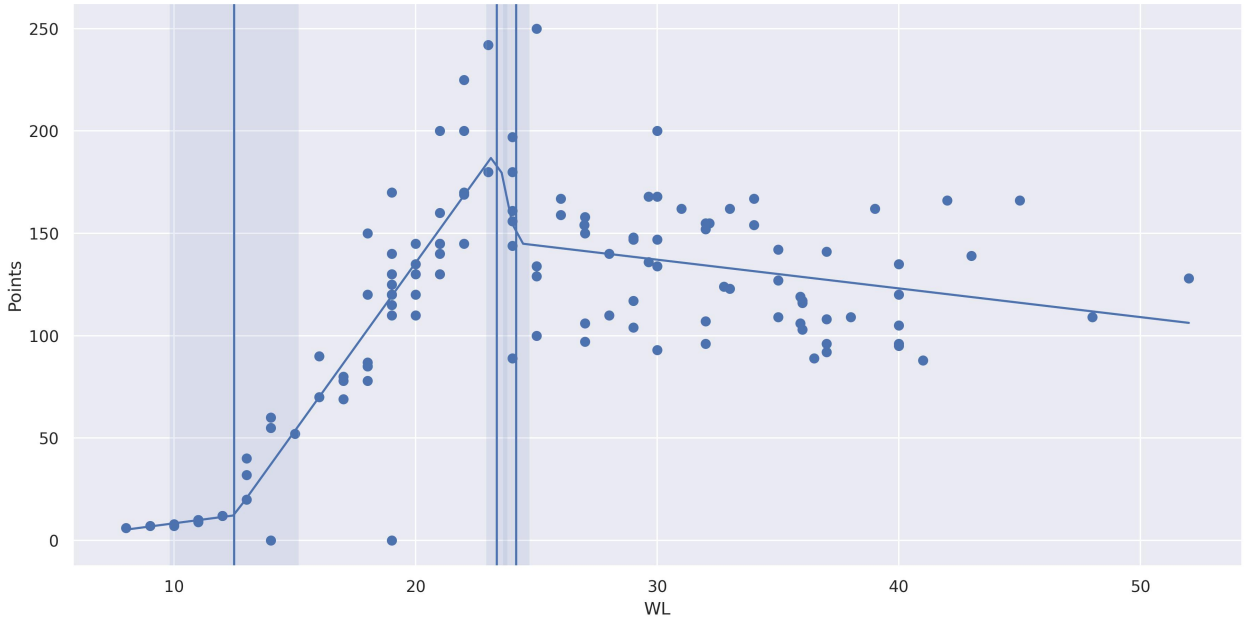
3.11 Model Building and Evaluation

```
# names of models
models=[   ('lr',LinearRegression()),('ElasticNet',ElasticNet()),('RF',RandomForestRegressor()),
          ('ETR',ExtraTreesRegressor()),('GBM',GradientBoostingRegressor()),('MLP',MLPRegressor())

]
```

```
def model_selection_function(x,y,cross_folds,model):
 scores=[]
 names=[]
 for i, j in model:
  cv_scores=cross_val_score(j,x,y,cv=cross_folds,n_jobs=-1)
  scores.append(cv_scores)
  names.append(i)
 for k in range(len(scores)):
  print(names[k],scores[k].mean())
 return
```

```
model_selection_function(X_train,y_train,3,models)
```

```
lr 0.4958178267370769
ElasticNet 0.47308329416120437
RF 0.5506369506161392
ETR 0.49725421385589685
GBM 0.590383608850568
MLP 0.4791630380241734
```




```
def list_models():
  models=[]
  models.append(('LR',LinearRegression()))
  models.append(('GBR',GradientBoostingRegressor()))
  return models

def fit_all_models(models,X_train,X_val,y_train,y_val):
  level_1_feat=[]
  for name,model in models:
    model.fit(X_train,y_train)
    y_hat=model.predict(X_val)
    y_hat=y_hat.reshape(len(y_hat),1)
    level_1_feat.append(y_hat)

  level_1_feat=hstack(level_1_feat)
  level_1_estimator= MLPRegressor()
  level_1_estimator.fit(level_1_feat,y_val)

  return level_1_estimator

def pred_data(models,blends,X_test):
  meta_model_X=[]
  for name,model in models:
    yhat=model.predict(X_test)
    yhat=yhat.reshape(len(yhat),1)
    meta_model_X.append(yhat)
  meta_model_X=hstack(meta_model_X)
```

```
 return blends.predict(meta_model_X)

models=list_models()
model_blender=fit_all_models(models,X_train_1,X_val,y_train_1,y_val)
projectprom.checkpoint('362665')

y_hat=pred_data(models,model_blender,X_train)
```

R2 of Blending is
0.6069217439513899
R2 of Stacking is
0.6103823798402186
R2 of GBM is
0.6098375340559932

```
#!pip install pygam
from pygam import LinearGAM, PoissonGAM,GammaGAM

gam=PoissonGAM(n_splines=5).gridsearch(X_train.values,y_train)
projectprom.checkpoint('362665')
y_pred=gam.predict(X_test.values)
print(r2_score(y_test,y_pred))
```

```
  0% (0 of 11) |                        | Elapsed Time: 0:00:00 ETA:  --:--:--
  9% (1 of 11) |##                      | Elapsed Time: 0:00:09 ETA:   0:01:34
did not converge
 18% (2 of 11) |####                    | Elapsed Time: 0:00:17 ETA:   0:01:15
did not converge
 27% (3 of 11) |######                  | Elapsed Time: 0:00:26 ETA:   0:01:07
did not converge
 36% (4 of 11) |########                | Elapsed Time: 0:00:34 ETA:   0:00:58
did not converge
 45% (5 of 11) |##########              | Elapsed Time: 0:00:43 ETA:   0:00:51
did not converge
 54% (6 of 11) |############            | Elapsed Time: 0:00:51 ETA:   0:00:42
did not converge
 63% (7 of 11) |##############          | Elapsed Time: 0:01:00 ETA:   0:00:34
did not converge
 72% (8 of 11) |################        | Elapsed Time: 0:01:09 ETA:   0:00:27
did not converge
 81% (9 of 11) |##################      | Elapsed Time: 0:01:18 ETA:   0:00:18
did not converge
 90% (10 of 11) |####################   | Elapsed Time: 0:01:31 ETA:   0:00:12
did not converge
100% (11 of 11) |######################| Elapsed Time: 0:01:44 Time:  0:01:44
did not converge
0.6119679799388373
```

```
model_gbm=GradientBoostingRegressor()
scores=cross_val_score(model_gbm,X,y,cv=5)
```

Chapter 4 : Concepts/ML Models

Different regression techniques used:

- 4.2 Linear Regressor
- 4.3 Elastic Net Regressor
- 4.4 Random Forest Regressor
- 4.5 Extra Trees Regressor
- 4.6 Gradient Boosting Regressor
- 4.7 MLP Regressor
- 4.8 Multivariate Adaptive Regression Splines (MARS)
- 4.9 Errors in Regression

Chapter 5: Power BI Outcome:

Challenges in implementing the solution :

- **Data Quality and Availability :** Insufficient or inconsistent data can lead to inaccurate predictions.
- **Model Complexity :** Sophisticated models might require significant computational resources.
- **Model Overfitting:** Models might perform exceptionally well on training data but fail to generalize to new data.
- **Scalability Issues :** As data volumes grow, scalability becomes a concern.
- **Interpreting Results :** Translating model outputs into actionable insights for sales strategies can be complex.
- **Integration with Existing Systems :** Incorporating new prediction models into existing sales.
- **Continuous Monitoring and Updating :** Models need regular updates to maintain accuracy.

Business Benefits:

- **Enhanced Forecasting Accuracy** : Predict with precision.
- **Personalized Customer Interactions** : Boost loyalty.
- **Optimized Pricing Strategies** : Dynamically adjust prices, maximize profitability.
- **Automated Sales Process** : Lead scoring, focus on high-value opportunities.
- **Insightful Sales Analytics** : Performance metrics, spot growth areas.
- **Improved Sales Team Productivity** : Enhance decision-making.
- **Scalable Infrastructure** : Evolve with business needs.
- **Competitive Advantage** : Lead with innovation, stand out in the marketplace.