

Correla

1

0.051 -0.055 0.28

0.051

-0.055

0.28

0.23

-0.21

0.097

1 0.052 -0.29

0.052 1 -0.093

-0.29 -0.093 1

-0.32 -0.037 0.91

0.05 0.53 -0.18

0.023 -0.23 0.24

Data Wragling using Pandas

Presented by: Shivangi Chaudhary,

Company Name-Nexthikes IT Solutions

Getting Started: Pandas & CSV Import

Key Libraries

Pandas is fundamental for data handling.

- Pandas for data structures
- NumPy for numerical computing
- seaborn, matplotlib

Loading Data

Using Jupyter notebook, Import CSV files effortlessly using **pd.read_csv()**. Specify the file path directly.

- Simple file upload
- Direct path specification

```
: fmaster_data = pd.read_csv("final_cleaned_dataset.csv")  
fmaster_data
```

(adding snippet of how we load csv file)

```
data_3.describe()
```

	instant	season	yr	mnth	hr	weekday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
count	390.000000	390.0	390.0	390.000000	390.000000	390.000000	390.000000	390.000000	390.000000	390.000000	390.000000	390.000000	390.000000	390.000000
mean	805.500000	1.0	0.0	1.800000	11.687179	2.989744	1.484615	0.220000	0.230424	0.613769	0.179416	5.576923	57.002564	62.579487
std	112.727548	0.0	0.0	0.400514	6.980295	2.149884	0.663805	0.073095	0.069455	0.202361	0.138551	9.317478	49.070198	53.274838
min	611.000000	1.0	0.0	1.000000	0.000000	0.000000	1.000000	0.060000	0.075800	0.210000	0.000000	0.000000	0.000000	1.000000
25%	700.250000	1.0	0.0	2.000000	6.000000	1.000000	1.000000	0.160000	0.181000	0.450000	0.000000	1.000000	17.000000	18.500000

Dataset Preprocessing Essentials

Summarize & Inspect

Use **.info()** and **.describe()** for quick overviews.

Understand data types and distributions.

Handle Missing Data

Identify and remove or impute null values. Use **.dropna()** or **.fillna()** strategically.

Remove Duplicates

Clean your dataset by eliminating duplicate rows. **.drop_duplicates()** is your tool.

Correct Dtypes & Drop Columns

Ensure columns have correct data types. Remove any unnecessary columns for analysis.

Effective preprocessing ensures data quality and prepares your dataset for accurate analysis.

Merging Datasets for Unified Analysis



Identify Common Keys

Find shared columns across datasets.



Choose Merge Type

Select 'inner', 'outer', 'left', or 'right' joins.

Here, I used inner merge.



Execute Merge

Use **pd.merge()** for combining DataFrames.

In our project, we perform merging two times.

(Adding snippet of final data after merge)

	dteday	season	hr	holiday	weekday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt	instant
0	2011-01-28	1	16	True	5	1	0.22	0.2727	0.80	0.0000	10	70	80	1
1	2011-01-28	1	17	True	5	1	0.24	0.2424	0.75	0.1343	2	147	149	2
2	2011-01-28	1	18	True	5	1	0.24	0.2273	0.75	0.1940	2	107	109	3
3	2011-01-28	1	19	True	5	2	0.24	0.2424	0.75	0.1343	5	84	89	4
4	2011-01-28	1	20	True	5	2	0.24	0.2273	0.70	0.1940	1	61	62	5
...
385	2011-02-14	1	3	True	1	1	0.34	0.3182	0.46	0.2239	1	1	2	386
386	2011-02-14	1	4	True	1	1	0.32	0.3030	0.53	0.2836	0	2	2	387
387	2011-02-14	1	5	True	1	1	0.32	0.3030	0.53	0.2836	0	3	3	388
388	2011-02-14	1	6	True	1	1	0.34	0.3030	0.46	0.2985	1	25	26	389
389	2011-02-14	1	7	True	1	1	0.34	0.3030	0.46	0.2985	2	96	98	390

90 rows × 14 columns



Basic Exploratory Data Analysis (EDA)

Line Graphs

Visualize trends over time. Ideal for sequential data exploration.

- Show temporal patterns
- Identify sequential changes
- Plot time series:
`sns.lineplot(x='date', y='sales')`

Bar Charts

Compare categorical data effectively. Display distributions and counts.

- Categorical comparisons
- Frequency distributions
- Plot categories:
`sns.barplot(x='category', y='count')`

Insight:

Reveals trends, anomalies, distribution patterns.

EDA helps uncover initial patterns, anomalies, and relationships within your data, forming hypotheses for further investigation.

I have used, line and bar graph to understand the dataset for dataset_1,dataset_2,dataset_3 and dataset_A.(adding snippet)

Advanced EDA: Outliers & Correlations

Outlier Detection

Identify data points far from typical values. Use box plots or statistical methods.



Outlier Handling

Decide whether to remove, transform, or cap outliers. Context matters.

Scatter Plots

Explore relationships between two continuous variables. Reveal clusters or trends.



Seaborn Heatmaps

Visualize correlations between variables. Quickly spot strong relationships.

Advanced EDA provides deeper insights, improving model performance and understanding complex data structures.

Navigating Challenges and how to overcome

- **New to advanced Pandas coding**

Challenge: Unfamiliar with functions like merge()

Solution: Used mentor-provided references and documentation; practiced using examples from Real Python, GeeksforGeeks, and Pandas docs.

- **Limited EDA and heatmap skills**

Challenge: Struggled with creating Seaborn heatmaps and plots

Solution: Learned through YouTube tutorials (e.g., Seaborn heatmap basics and customization)

- **Data with duplicate columns (X and Y versions)**

Challenge: Final dataset had redundant X and Y columns making it hard to manage

Solution: Wrote custom Pandas code to streamline columns (e.g., dynamic renaming or consolidation); searched online (Google & YouTube).

References-

[Python Merge Tutorial](#)

[User guide and tutorial — seaborn 0.13.2 documentation](#)

[Pandas Tutorial \(Data Analysis In Python\)](#)

[Master Exploratory Data Analysis \(EDA\) in Python: Step-by-Step Jupyter Notebook Tutorial](#)

Thank You!

Thank you for your attention. We hope this presentation offered valuable insights into leveraging Pandas for powerful data analysis. Please feel free to ask any questions.

(Also, the images that I have used in this ppt are from snippets from my projects.)