FROM MESS TO MODEL: MASTERING DATA PREPARATION

CS 180 Introduction to Data Science

WHY IS DATA PREPARATION SO IMPORTANT?

- Garbage IN – Garbage OUT
- Impacts model accuracy and reliability.
- A model is only as good as the data it learns from.
- Ensures trust in insights
- Data scientists spend ~70-80% of their time cleaning and preparing data

BYU

LEARNING OBJECTIVES

By the end of this lesson, students will be able to:

- Understand the importance of data preparation in the overall data science lifecycle.
- Explain why data preparation is critical to successful data science projects.
- Identify and apply standard techniques for cleaning datasets.
- Learn key data transformation and feature engineering techniques using Python.

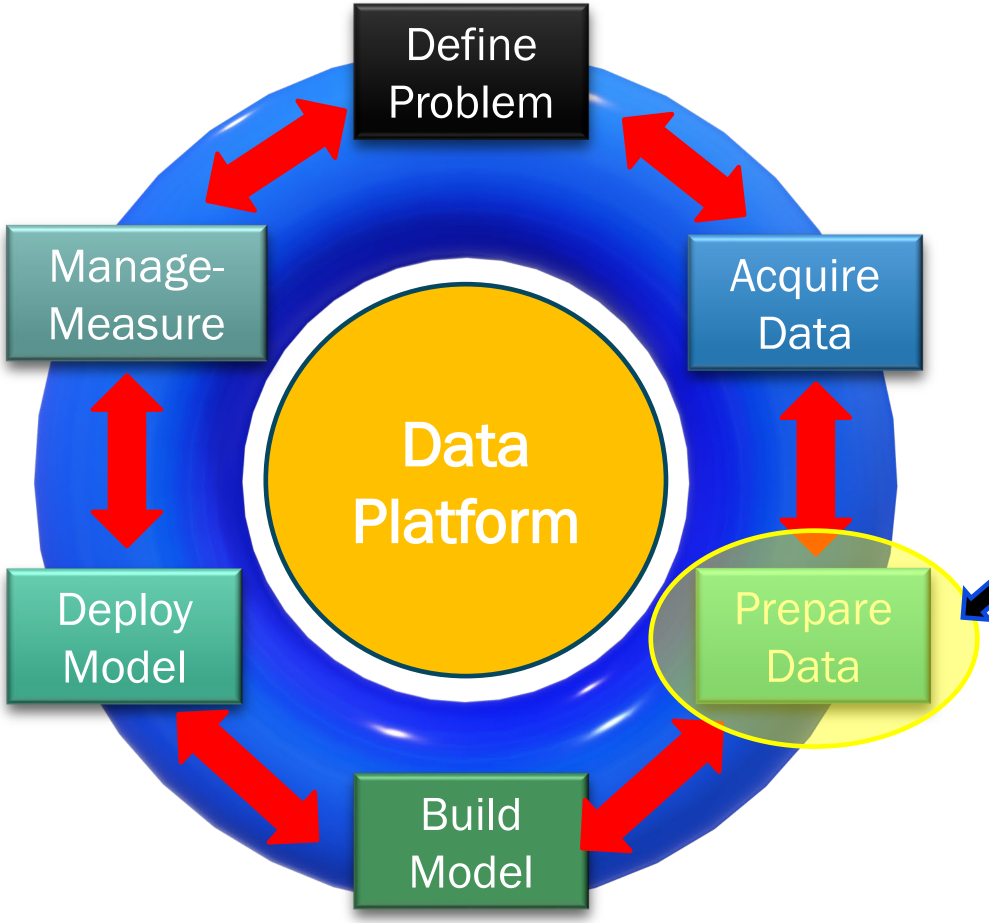
BYU

DATA PREPARATION CONCEPTS FOR TODAY

- Data Profiling
- Data Cleaning
  - Missing Values
  - Inconsistencies
  - Duplicates
  - Outliers
- Data Transformation
  - Standardization
  - Normalization
  - Aggregation (e.g., Groupby)
  - Pivot Table
  - Contingency Table or Crosstab

BYU

REVIEW: THE 6S5P DATA SCIENCE LIFECYCLE



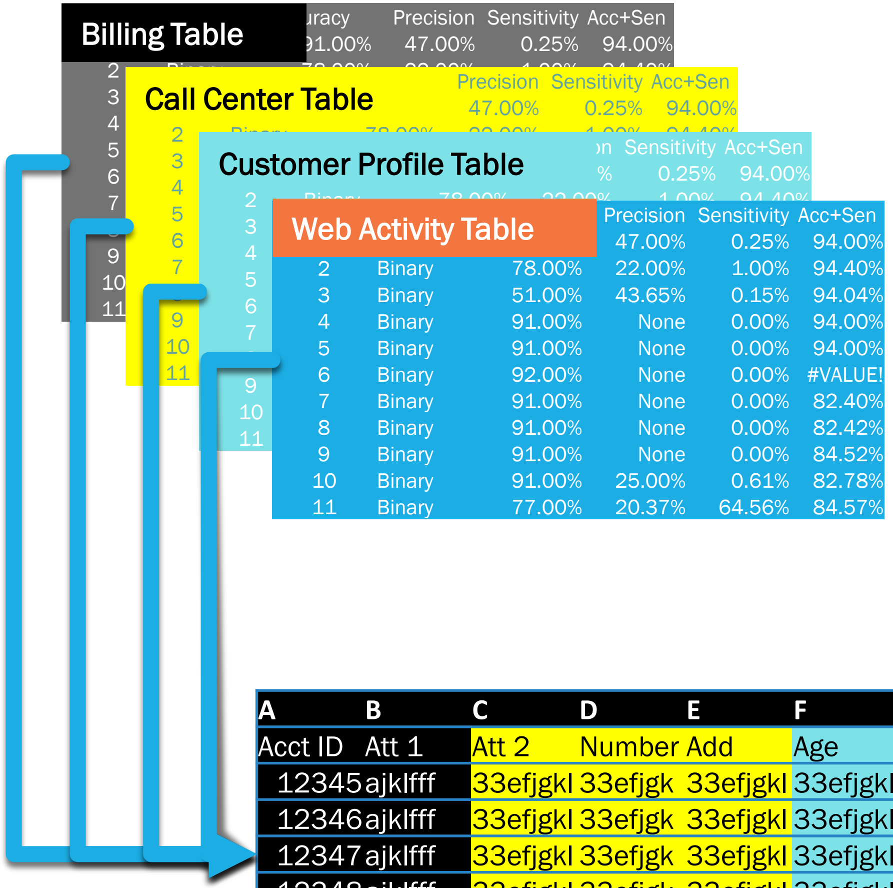
Focus for the next few weeks

BYU

2. IDENTIFY AND ACQUIRE THE DATA

- Leverage Data Catalog to ID relevant data
- Use ETL# tools, processes
- Join data sources (often using SQL)
- Data Type checking
- Schema Matching
- Output: Raw Analytic Data Set or View

Source Data



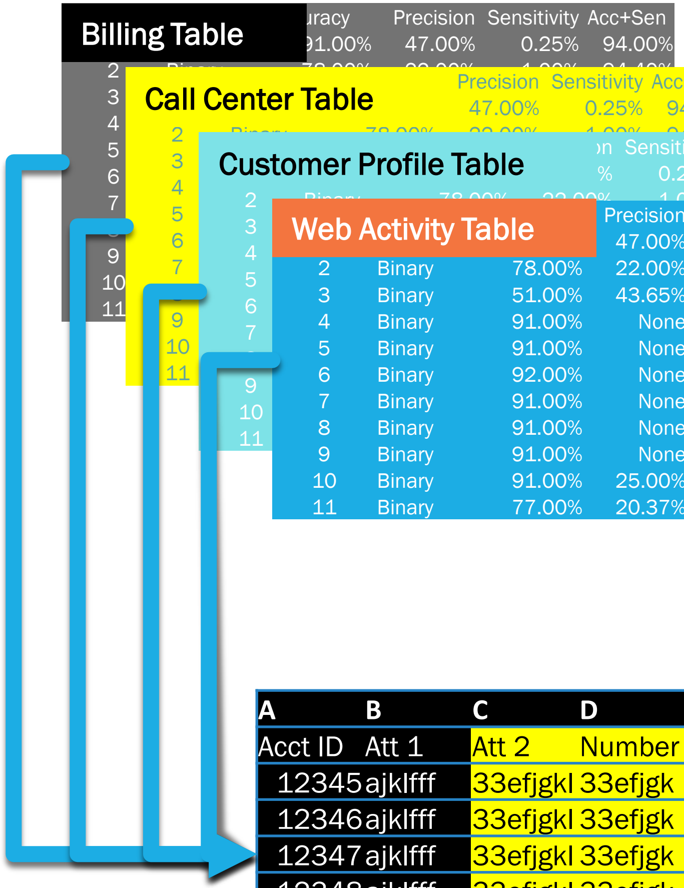
A	B	C	D	E	F	G	H	I	J	K	L	M
Acct ID	Att 1	Att 2	Number Add		Age	SSN	SSN	SSN	SSN	SSN	SSN	SSN
12345	ajklfff	33efjgkl	33efjgk	33efjgkl	33efjgkl	33efjgkl	33efjg	33efj	33ef	33ef	33ef	33ef
12346	ajklfff	33efjgkl	33efjgk	33efjgkl	33efjgkl	33efjgkl	fjfkl	123	kjj	666	4236	4425
12347	ajklfff	33efjgkl	33efjgk	33efjgkl	33efjgkl	33efjgkl	fjfkl	123	kjj	666	4236	4425
12348	ajklfff	33efjgkl	33efjgk	33efjgkl	33efjgkl	33efjgkl	fjfkl	123	kjj	666	4236	4425
12349	ajklfff	33efjgkl	efjgklk	33efjgkl	33efjgkl	33efjgkl	fjfkl	123	kjj	666	4236	4425
12350	ajklfff	33efjgkl	33efklk	33efjgkl	33efjgkl	33efjgkl	fjfkl	123	kjj	666	4236	4425
12351	ajklfff	33efjgkl	#%$GG	33efjgkl	33efjgkl	33efjgkl	fjfkl	123	kjj	666	4236	4425

Raw ADS

- Potential variables required to solve business problem
- Very wide data set
- Derived, Joined and Lightly Transformed

3. UNDERSTAND AND PREPARE THE DATA

- Profile Data
- Clean Data (missing / corrupted values)
- Outlier Analysis
- Transform Data
- Generate New Features
- Visualize Data
- Analyze Data (*EDA)
- Create Analytic Data Set (ADS)



Web Activity Table			Precision	Sensitivity	Acc+Sen
			47.00%	0.25%	94.00%
2	Binary	78.00%	22.00%	1.00%	94.40%
3	Binary	51.00%	43.65%	0.15%	94.04%
4	Binary	91.00%	None	0.00%	94.00%
5	Binary	91.00%	None	0.00%	94.00%
6	Binary	92.00%	None	0.00%	#VALUE!
7	Binary	91.00%	None	0.00%	82.40%
8	Binary	91.00%	None	0.00%	82.42%
9	Binary	91.00%	None	0.00%	84.52%
10	Binary	91.00%	25.00%	0.61%	82.78%
11	Binary	77.00%	20.37%	64.56%	84.57%

A	B	C	D	E	F	G	H	I	J	K	L	M
Acct ID	Att 1	Att 2	Number Add		Age	SSN	SSN	SSN	SSN	SSN	SSN	SSN
12345	ajklfff	33efjgkl	33efjgk	33efjgkl	33efjgkl	33efjgkl	33efjg	33efj	33ef	33ef	33ef	33ef
12346	ajklfff	33efjgkl	33efjgk	33efjgkl	33efjgkl	33efjgkl	fjfkl	123	kjj	666	4236	4425
12347	ajklfff	33efjgkl	33efjgk	33efjgkl	33efjgkl	33efjgkl	fjfkl	123	kjj	666	4236	4425
12348	ajklfff	33efjgkl	33efjgk	33efjgkl	33efjgkl	33efjgkl	fjfkl	123	kjj	666	4236	4425
12349	ajklfff	33efjgkl	efjgklk	33efjgkl	33efjgkl	33efjgkl	fjfkl	123	kjj	666	4236	4425
12350	ajklfff	33efjgkl	33efklk	33efjgkl	33efjgkl	33efjgkl	fjfkl	123	kjj	666	4236	4425
12351	ajklfff	33efjgkl	#%$GG	33efjgkl	33efjgkl	33efjgkl	fjfkl	123	kjj	666	4236	4425

Modeling Ready ADS

A	C	E	F	K	M
Acct ID	Att 2	Att 31	Age	SSN	REV
12345	33efjgkl	33efjgkl	33efjgkl	33ef	33ef
12346	33efjgkl	33efjgkl	33efjgkl	666	4425
12347	33efjgkl	33efjgkl	33efjgkl	666	4425
12348	33efjgkl	33efjgkl	33efjgkl	666	4425
12349	33efjgkl	33efjgkl	33efjgkl	666	4425
12350	33efjgkl	33efjgkl	33efjgkl	666	4425
12351	33efjgkl	33efjgkl	33efjgkl	666	4425
12345	33efjgkl	33efjgkl	33efjgkl	33ef	2223
12346	33efjgkl	33efjgkl	33efjgkl	666	4425
12347	33efjgkl	33efjgkl	33efjgkl	666	4425
12348	33efjgkl	33efjgkl	33efjgkl	666	4425
12349	33efjgkl	33efjgkl	33efjgkl	666	4425
12350	33efjgkl	33efjgkl	33efjgkl	666	4425
12351	33efjgkl	33efjgkl	33efjgkl	666	4425
2345	Fjlkd	Jlkjf	Fjfjf	345	0123

PANDAS DATA PROFILING

Function	Purpose	Best For	Output
df.info()	Provides a technical summary of the DataFrame.	Quickly checking data types and finding columns with missing values.	A text summary of columns, non-null counts, and data types.
df.describe()	Generates descriptive statistics for numerical columns.	Understanding the distribution, scale, and potential outliers of numerical data.	A table of statistics (mean, median, min, max, etc.).
df.isnull().sum()	Counts the exact number of missing (NaN) values in each column.	Quantifying the severity of missing data for each feature.	A Series showing each column name and its null count.
df['col'].value_counts()	Counts the unique values in a categorical column.	Analyzing the distribution of categories and spotting inconsistencies.	A Series of unique values and their frequencies.
df.duplicated().sum()	Counts the total number of duplicate rows.	Checking for data redundancy and integrity issues.	A single integer representing the number of duplicate rows.

AUTOMATED DATA PROFILING

These libraries are designed to give you a comprehensive overview of your dataset with minimal code, saving you a significant amount of time during the initial exploratory phase.

- ydata-profiling: The most popular choice for generating a detailed, interactive HTML report that covers everything from missing values and correlations to data distributions.
- Sweetviz: Excellent for creating beautiful, visual reports and is especially powerful for comparing two datasets (like a training and a testing set).

BYU

THE ZYBOOK DATA PREPARATION PROCESS

Table 3.1.1: Steps of data wrangling.

Step	Description
Step 1: Discovering	Discovery, also called data exploration, familiarizes the data scientist with source data in preparation for subsequent steps.
Step 2: Structuring	Structuring data transforms features to uniform formats, units, ar
Step 3: Cleaning	Cleaning data removes or replaces missing and outlier data.
Step 4: Enriching	Enriching data derives new features from existing features and ap
Step 5: Validating	Validating data verifies that the dataset is internally consistent and accurate.
Step 6: Publishing	Publishing data makes the dataset available to other data scientists by storing data in a database, uploading data to the cloud, or distributing data files.

Not in the correct sequence. You should not do any transformations on the data until after it has been initially cleaned. Otherwise, you are wasting operations and may generate errors

THE UPDATED DATA PREPARATION PROCESS

Step	Description
Step 1: Profiling	Data Profiling, also called data exploration, familiarizes the data scientist with source data in preparation for subsequent steps.
Step 2: Cleaning	Cleaning data removes or replaces missing and outlier data.
Step 3: Feature Gen	Enriching data derives new features from existing features and appends new data from external sources.
Step 4: Transform	Structuring data transforms to uniform formats, units, and scales.
Step 5: Validating	Validating data verifies that the dataset is internally consistent and accurate.
Step 6: Publishing	Publishing data makes the dataset available to other data scientists by storing data in a database, uploading data to the cloud, or distributing data files.

This is an iterative process. As new features/data are received, cycle through each step again.

BYU

DATA CLEANING

- Missing Values
  - Code snippet with fillna().
  - Visual: Table before/after missing value imputation.
- Inconsistencies
  - Example: “CA” vs “California.”
  - Visual: Table showing cleaned categories.
- Duplicates
  - Code: drop_duplicates().
  - Visual: Highlight duplicate rows.
- Outliers
  - Z-score detection.
  - Visual: Boxplot before/after removing outliers.

BYU

DATA TRANSFORMATION & FEATURE ENGINEERING

- Feature Scaling: Normalization vs Standardization
  - Code: MinMaxScaler vs StandardScaler.
- Aggregation
  - Example: Sales per month.
- Feature Engineering
  - Example: “DayOfWeek” from timestamp.

FEATURE SCALING: NORMALIZATION VS. STANDARDIZATION

- A dataset's numeric features often have different scales, and in some datasets, scales may differ by orders of magnitude.
- This incongruity may bias some algorithms by giving more weight to larger numbers.
- *Feature scaling* converts numeric features to uniform ranges. Two of the most common feature scaling methods are standardization and normalization.

BYU

NORMALIZATION (AKA MIN-MAX SCALING)

- There are many types of data “normalization”
- Min-Max Normalization scales values between [0, 1] using the following formula:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

```
# Manually normalize each column to [0, 1] range
# Sample data
data = {'feature1': [1, 2, 3, 4, 5],
        'feature2': [10, 15, 20, 25, 30]}
df = pd.DataFrame(data)

df_normalized_manual = (df - df.min()) / (df.max() - df.min())
print(df_normalized_manual)
```

```
   feature1  feature2
0      0.00      0.00
1      0.25      0.25
2      0.50      0.50
3      0.75      0.75
4      1.00      1.00
```

STANDARDIZATION (AKA Z-SCORE NORMALIZATION)

- Rescales value to have a mean of 0 and standard deviation of 1:

$$x' = \frac{x - \mu}{\sigma}$$

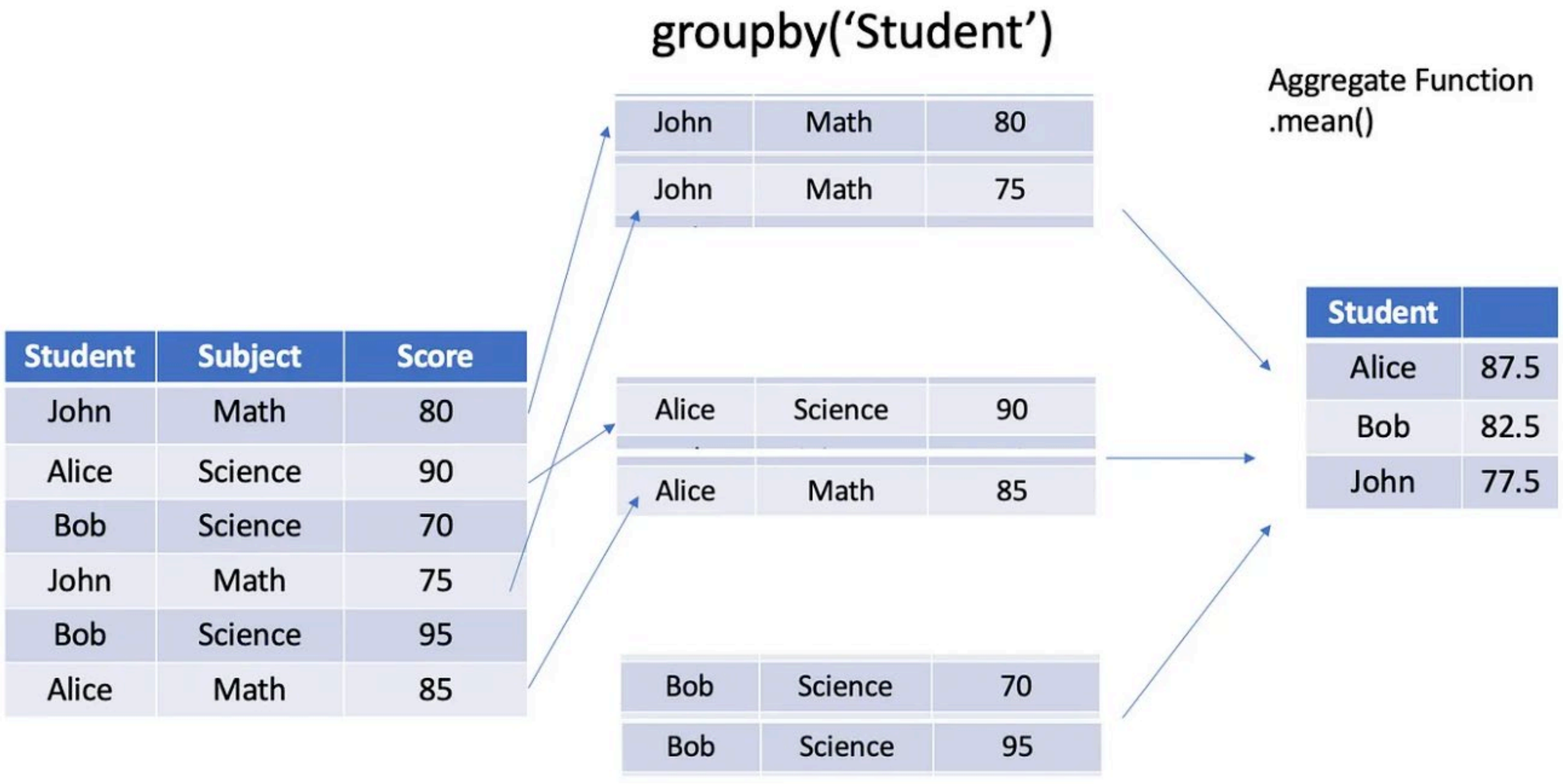
```python
# Standardize the data (Z-score normalization)
df_standardized = (df - df.mean()) / df.std()

print(df_standardized)
```

```
   feature1  feature2
0 -1.264911 -1.264911
1 -0.632456 -0.632456
2  0.000000  0.000000
3  0.632456  0.632456
4  1.264911  1.264911
```

BYU

GROUPBY

- The groupby() function in pandas is one of the most powerful and frequently used tools for data analysis.
- The process is best understood by the term **"Split-Apply-Combine."**
- **Split:** The data is split into groups based on some criteria (e.g., all rows with the same student).
- **Apply:** A function is applied to each group independently (e.g., mean score for each student).
- **Combine:** The results of these operations are combined into a new DataFrame or Series.



BYU

PIVOT TABLE

- df.pivot_table() takes several parameters.
- **value** specifies the values in the pivot table's elements.
- The feature in the pivot table's rows is specified using **index** and the feature in the pivot table's columns is **columns**.
- **aggfunc** specifies a function to apply to the values in each row/column combination within the pivot table.
- The default aggregate function is np.mean. Other functions include: sum, count, min, max, median, mode, var (variance), first, last, size, custom functions (e.g., aggfunc=lambda x: x.max() - x.min()
- You can also apply multiple aggregation functions by passing them as a list: pd.pivot_table(df, values='Sales', index='Region', aggfunc=['mean', 'sum', 'count'])

CROSSTAB / CONTINGENCY / FREQUENCY TABLE

- **Purpose**: Primarily used to calculate the frequency (or other statistics) between two or more categorical variables.
- **Default Output**: Crosstabs typically result in contingency tables, where each row and column represents a different category, and the cells show the frequency of their co-occurrence.
- **Common Use**: It is most commonly used for counting and can also apply some basic aggregation functions.
- **Crosstab** is more limited and generally used for frequency counts, while **pivot tables** provide more flexibility, allowing for various types of aggregations (mean, sum, etc.).
- Crosstabs are usually used for two categorical variables, whereas pivot tables can handle both categorical and numerical data with multiple levels of summarization.

BYU