SCREEN SENSE – KIDS' SCREENTIME VISUALIZATION

WEEK 2 REPORT – PREPROCESSING AND FEATURE ENGINEERING

1. OBJECTIVE

The main goal for Week 2 was to **clean, preprocess, and enhance the dataset** through feature engineering to enable meaningful visualization and analysis.
This phase focused on handling inconsistencies, creating derived metrics such as risk and behaviour indicators, and preparing the data for analytical and dashboard-ready insights.
These transformations will help identify patterns in kids' screen usage behaviour and potential risk factors affecting their well-being.

2. IMPLEMENTATION

Step 1: Importing Required Libraries

Imported essential Python libraries such as **pandas** and **NumPy** for data manipulation and feature creation.
These form the foundation for cleaning, transforming, and analysing the dataset efficiently.

```
import pandas as pd
import numpy as np
```

Step 2: Dataset Loading

Loaded the **'Indian Kids Screentime 2025'** dataset into the notebook and verified successful import.
The dataset contains **9,712 records and 8 columns**, ensuring a strong foundation for analysis.

```
# Load the Dataset

file_path = "D:\Indian_Kids_Screen_Time.csv"
df = pd.read_csv(file_path)
print(" Dataset loaded successfully for preprocessing.")
print("Shape:", df.shape)
```

```
 Dataset loaded successfully for preprocessing.
Shape: (9712, 8)
```

Step 3: Handling Missing and Inconsistent Values

- Trimmed extra spaces and standardized text casing across all object columns.
- Unified category names such as **'male'** → **'Male'**, **'urban'** → **'Urban'**, ensuring uniformity.
- Replaced missing values in *Primary_Device* and *Health_Impacts* with default entries ('Unknown', 'None') to maintain completeness.

```
# Handle Missing / Inconsistent Values

# Trim whitespace and standardize text casing
for col in df.select_dtypes(include='object').columns:
    df[col] = df[col].str.strip().str.title()
```

```
# Check for missing values

print("\n Missing values before handling:\n", df.isnull().sum())
```

```
 Missing values before handling:
 Age                                0
Gender                              0
Avg_Daily_Screen_Time_hr            0
Primary_Device                      0
Exceeded_Recommended_Limit          0
Educational_to_Recreational_Ratio   0
Health_Impacts                   3218
Urban_or_Rural                      0
dtype: int64
```

Step 4: Handling Inconsistent Categories

Replaced short or lowercase entries with standardized forms for **Gender** and **Urban or Rural**.
This step ensures accurate grouping and aggregation in visual analysis.

```
# (If any missing values exist)

df.fillna({
    'Primary_Device': 'Unknown',
    'Health_Impacts': 'None'
}, inplace=True)
```

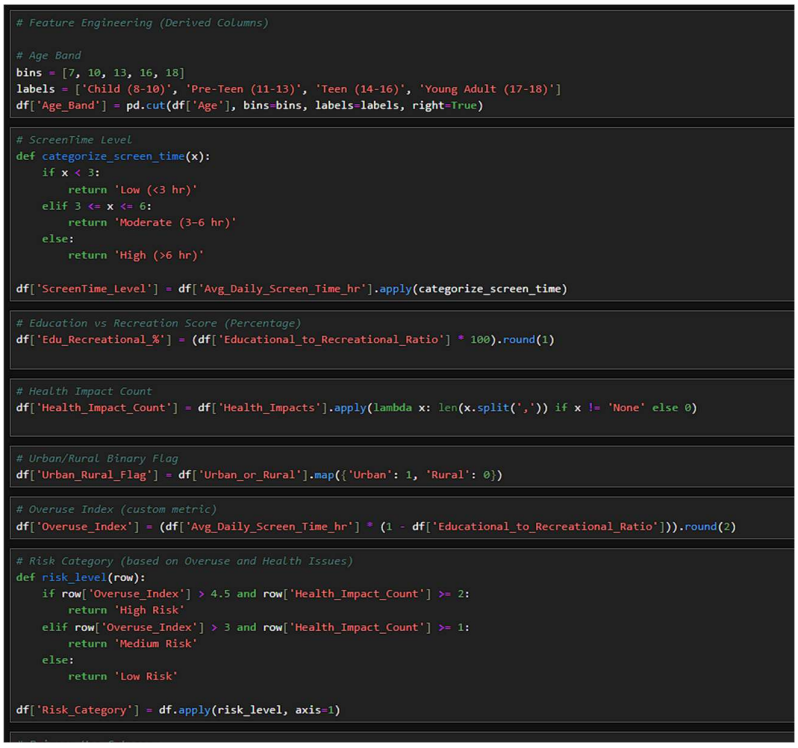
```
# Handle Inconsistent Categories

# Standardize gender categories
df['Gender'] = df['Gender'].replace({
    'M': 'Male', 'F': 'Female', 'male': 'Male', 'female': 'Female'
})
```

```
# Standardize Urban/Rural field
df['Urban_or_Rural'] = df['Urban_or_Rural'].replace({
    'urban': 'Urban', 'rural': 'Rural'
})
```

Step 5: Feature Engineering (Derived Columns Creation)

Multiple new analytical features were added to improve interpretability and enable advanced insights:

- Age Band – Groups kids by age range (Child, Pre-Teen, Teen, Young Adult).
- Screen Time Level – Categorizes users as *Low (<3 hr)*, *Moderate (3–6 hr)*, or *High (>6 hr)* based on daily screen hours.
- Edu Recreational% – Converts educational-to-recreational ratio into percentage for better understanding.
- Health Impact Count – Counts health issues reported per child.
- Urban Rural Flag – Encodes location type numerically (Urban = 1, Rural = 0).
- Overuse Index – A custom metric combining screen time and recreation ratio to detect overuse.
- Risk Category – Automatically labels users as *Low*, *Medium*, or *High Risk* based on health impact and overuse index.
- Primary Use Category – Classifies main device usage as *Educational*, *Entertainment*, or *Gaming*.
- Device Usage Context – Combines device and location for comparative trend analysis.
- Weekday Weekend Usage – Indicates whether screen time is higher on weekends.
- Usage Behaviour Type – Categorizes users as *Study Focused*, *Balanced*, or *Recreational Heavy*.
- Health Risk Text – Converts numeric risk levels into dashboard-friendly text such as *Needs Attention* or *Healthy Usage*.



```
# Feature Engineering (Derived Columns)

# Age Band
bins = [7, 10, 13, 16, 18]
labels = ['Child (8-10)', 'Pre-Teen (11-13)', 'Teen (14-16)', 'Young Adult (17-18)']
df['Age_Band'] = pd.cut(df['Age'], bins=bins, labels=labels, right=True)

# ScreenTime Level
def categorize_screen_time(x):
    if x < 3:
        return 'Low (<3 hr)'
    elif 3 <= x <= 6:
        return 'Moderate (3-6 hr)'
    else:
        return 'High (>6 hr)'

df['ScreenTime_Level'] = df['Avg_Daily_Screen_Time_hr'].apply(categorize_screen_time)

# Education vs Recreation Score (Percentage)
df['Edu_Recreational_%'] = (df['Educational_to_Recreational_Ratio'] * 100).round(1)

# Health Impact Count
df['Health_Impact_Count'] = df['Health_Impacts'].apply(lambda x: len(x.split(',')) if x != 'None' else 0)

# Urban/Rural Binary Flag
df['Urban_Rural_Flag'] = df['Urban_or_Rural'].map({'Urban': 1, 'Rural': 0})

# Overuse Index (custom metric)
df['Overuse_Index'] = (df['Avg_Daily_Screen_Time_hr'] * (1 - df['Educational_to_Recreational_Ratio'])).round(2)

# Risk Category (based on Overuse and Health Issues)
def risk_level(row):
    if row['Overuse_Index'] > 4.5 and row['Health_Impact_Count'] >= 2:
        return 'High Risk'
    elif row['Overuse_Index'] > 3 and row['Health_Impact_Count'] >= 1:
        return 'Medium Risk'
    else:
        return 'Low Risk'

df['Risk_Category'] = df.apply(risk_level, axis=1)
```

```
# Quick Check

print("\n Sample of Processed Data:\n")
display(df.head(10))
```

Sample of Processed Data:

_Ratio	Health_Impacts	Urban_or_Rural	Age_Band	ScreenTime_Level	Edu_Recreational_%	Health_Impact_Count	Urban_Rural_Flag	Overuse_Index	Risk_Category	Primary
0.42	Poor Sleep, Eye Strain	Urban	Teen (14-16)	Moderate (3–6 hr)	42.0	2	1	2.31	Low Risk	
0.30	Poor Sleep	Urban	Pre-Teen (11-13)	Moderate (3–6 hr)	30.0	1	1	3.23	Medium Risk	
0.32	Poor Sleep	Urban	Young Adult (17-18)	Moderate (3–6 hr)	32.0	1	1	2.54	Low Risk	
0.39	None	Urban	Teen (14-16)	Low (<3 hr)	39.0	0	1	0.74	Low Risk	
0.49	Poor Sleep, Anxiety	Urban	Pre-Teen (11-13)	Moderate (3–6 hr)	49.0	2	1	3.00	Low Risk	
0.44	Poor Sleep	Urban	Teen (14-16)	Moderate (3–6 hr)	44.0	1	1	2.73	Low Risk	
0.48	None	Rural	Young Adult (17-18)	Low (<3 hr)	48.0	0	0	1.54	Low Risk	
0.54	None	Urban	Child (8-10)	Low (<3 hr)	54.0	0	1	1.26	Low Risk	
0.36	Poor Sleep, Anxiety	Rural	Teen (14-16)	Moderate (3–6 hr)	36.0	2	0	2.95	Low Risk	
	Poor Sleep,		Young							

Step 6: Saving the Processed Dataset:

1. The final cleaned and enriched dataset was saved as **screensense_cleaned_textbased.csv**, containing all original and derived features. This version will be used in Week 3 for visual analysis and pattern discovery.

```
df.to_csv("screensense_cleaned.csv", index=False)
```

```
import os
print(os.getcwd())
```

```
C:\Users\geeky
```

```
df.to_csv(r"C:\Users\geeky\Downloads\screensense_cleaned.csv", index=False)
```

3. CONCLUSION

The dataset is now standardized, clean, and enriched with multiple behavioral and risk-based features.
These engineered metrics will enable more meaningful visualizations and insights — such as identifying **which age groups or device types show high screen dependence**, or **how urban vs. rural usage differs**.
This structured dataset forms the analytical backbone for Week 3's **Univariate and Bivariate Visual Analysis**, directly supporting the project's goal of understanding children's digital habits and promoting healthier screen time balance.

Report By – Yash Verma