Class 4: Advanced Pandas for Data Processing

Overview

Pandas is a powerful Python library for data processing, critical for data science workflows. This class note covers advanced Pandas techniques for data cleaning, exploratory data analysis (EDA), feature engineering, data transformation, and data splitting, preparing data for machine learning applications.

Objectives

- Clean data by handling missing values, duplicates, and outliers.
- Perform exploratory data analysis to understand data patterns.
- Apply feature engineering to select relevant features.
- Transform data through normalization and encoding categorical variables.
- Split data into training, testing, and validation sets.

Key Concepts

1. **Data Cleaning**:
   - *Missing Values*: Impute with mean/median or drop rows.
   - *Duplicates*: Remove using `drop_duplicates()`.
   - *Outliers*: Detect and remove using InterQuartile Range (IQR).
   - *Data Types*: Convert columns (e.g., `astype('datetime64[ns]')`).
2. **Exploratory Data Analysis (EDA)**: Use `describe()` for summary statistics and `plot()` for visualization.
3. **Feature Engineering**: Select relevant features based on domain knowledge or statistical methods.
4. **Data Transformation**:
   - *Normalization*: Scale numeric data (e.g., (x - mean)/std).
   - *Encoding*: Convert categorical data using one-hot or label encoding.
5. **Data Splitting**: Divide data into train (80%), test (10%), and validation (10%) sets for unbiased model training and evaluation.

Example Code

```
import pandas as pd
import numpy as np

% Loading data
health_data = pd.read_csv(”health_monitor_data.csv”)

% Handling missing values
avg_calories = health_data[”Calories”].mean()
health_data[”Calories”] = health_data[”Calories”].fillna(avg_calories)
health_data = health_data.dropna()  % Drop remaining missing values

% Removing duplicates
health_data = health_data.drop_duplicates()

% Converting data types
health_data[”Date”] = health_data[”Date”].astype(”datetime64[ns]”)

% Removing outliers using IQR
Q1 = health_data[”Pulse”].quantile(0.25)
Q3 = health_data[”Pulse”].quantile(0.75)
IQR = Q3 - Q1
L, R = Q1 - 1.5 * IQR, Q3 + 1.5 * IQR
health_data = health_data[health_data[”Pulse”].between(L, R)]

% Exploratory data analysis
print(health_data.describe())
health_data.plot(kind=”line”, x=”Pulse”)

% Normalizing numeric columns
def normalize(col):
    return (col - col.mean()) / col.std()

health_data[”Pulse”] = health_data[”Pulse”].agg(normalize)
health_data[”Calories”] = health_data[”Calories”].agg(normalize)

% One-hot encoding categorical data
health_data = pd.get_dummies(health_data, columns=[”Type”], sparse=True)

% Label encoding categorical data
health_data[”label_encoded_type”] = pd.Categorical(
    health_data[”Type”],
    categories=[”Easy”, ”Moderate”, ”Heavy”],
    ordered=True
).codes
```

Key Takeaways

Advanced Pandas techniques enable robust data preprocessing for machine learning. Cleaning, transforming, and splitting data ensures high-quality inputs for training unbiased and effective models.