Data Collection and Preprocessing Phase

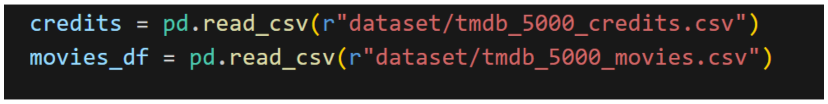
Date	15 June 2025
Team ID	SWTID1749705685
Project Title	Movie Box Office Gross Prediction using Machine Learning
Maximum Marks	6 Marks

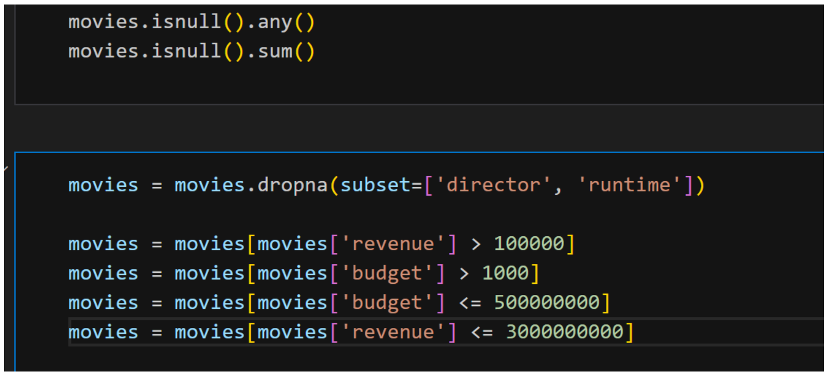
Data Exploration and Preprocessing Template

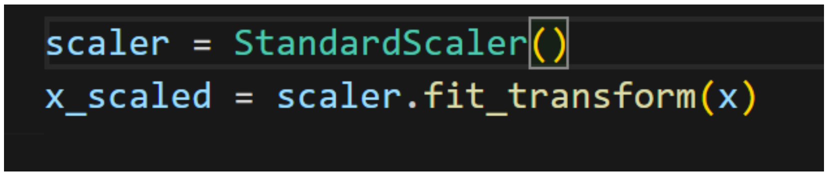
The dataset was sourced from TMDB and merged using movie IDs from two CSV files. Missing values in key fields like `director` and `runtime` were dropped, and date fields were imputed using the mode. Outliers in `budget` and `revenue` were filtered using logical thresholds. These steps ensured clean, reliable data for analysis and modeling.

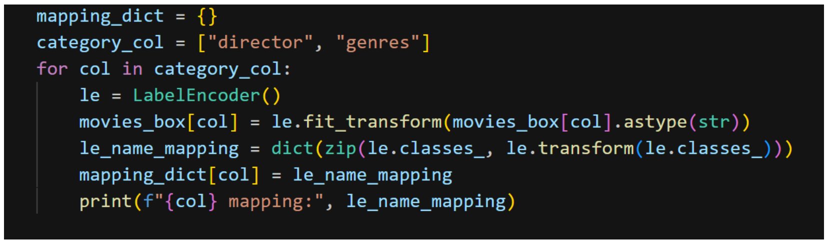
Section	**Description**
Data Overview	We used the TMDB 5000 Movie Dataset consisting of two CSV files: **`tmdb_5000_movies.csv`** and **`tmdb_5000_credits.csv`**. After merging both on the movie ID, the final dataset had 4800+ records and over 20+ features. Key features include budget, revenue, runtime, genres, popularity, vote average, and director.
Univariate Analysis	Summary statistics like mean, median, mode, min, max, and standard deviation were computed for numerical columns (e.g., budget, revenue, popularity). We also visualized distributions using boxplots and histograms.
Bivariate Analysis	Correlation heatmaps and scatter plots revealed strong positive correlation between budget and revenue. Other relationships (e.g., vote count vs. revenue) were also analyzed.
Multivariate Analysis	Analyzed how multiple features (e.g., budget, genre, release month) collectively influence revenue. Applied regression-based insights

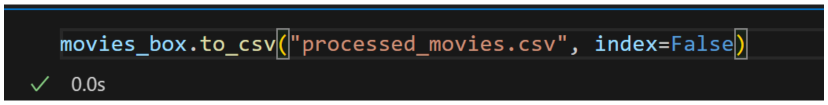
Outliers and Anomalies	Removed extreme outliers (e.g., movies with budget > $500M or revenue > $3B). Dropped rows with revenue < $100K or budget < $1000.
Data Preprocessing Code Screenshots	

Loading Data	
	`credits = pd.read_csv(r"dataset/tmdb_5000_credits.csv")`
	`movies_df = pd.read_csv(r"dataset/tmdb_5000_movies.csv")`

Handling Missing Data	
	`movies.isnull().any()`
	`movies.isnull().sum()`
	
	`movies = movies.dropna(subset=['director', 'runtime'])`
	
	`movies = movies[movies['revenue'] > 100000]`
	`movies = movies[movies['budget'] > 1000]`
	`movies = movies[movies['budget'] <= 500000000]`
	`movies = movies[movies['revenue'] <= 3000000000]`

Data Transformation	
	`scaler = StandardScaler()`
	`x_scaled = scaler.fit_transform(x)`

Feature Engineering	
	`mapping_dict = {}`
	`category_col = ["director", "genres"]`
	`for col in category_col:`
	`    le = LabelEncoder()`
	`    movies_box[col] = le.fit_transform(movies_box[col].astype(str))`
	`    le_name_mapping = dict(zip(le.classes_, le.transform(le.classes_)))`
	`    mapping_dict[col] = le_name_mapping`
	`    print(f"{col} mapping:", le_name_mapping)`

Save Processed Data	 `movies_box.to_csv("processed_movies.csv", index=False)` ✓ 0.0s