The notebook includes various code cells for building and evaluating machine learning models. Here's a step-by-step breakdown for the project "BoxOfficeRevenuePrediction":

1. Data Loading and Preprocessing
  Step 1: Importing libraries
  Essential libraries such as `pandas`, `numpy`, and `sklearn` are imported for handling data and building models.
  Step 2: Loading the dataset
  The dataset is loaded into a Pandas DataFrame.
  Step 3: Feature Selection and Cleaning
  Key features relevant to predicting box office revenue are selected, and missing values are handled.

2. Data Splitting
  Step 4: Train-Test Split
  The dataset is split into training and testing sets using `train_test_split` from `sklearn`.

3. Model Building
  Step 5: Linear Regression Model
  A Linear Regression model is built using `LinearRegression()` from `sklearn`.
  - The model is trained on the training set and evaluated on the test set.
  - Evaluation metric: `score()` method, which returns the coefficient of determination (R^2 score).

  Step 6: K-Nearest Neighbors (KNN) Regressor
  The KNN model is created using `KNeighborsRegressor(n_neighbors=5)`.
  - Model training and evaluation on scaled data (using `MinMaxScaler()`).

  Step 7: Decision Tree Regressor
  A Decision Tree model is built using `DecisionTreeRegressor()`.
  - The model is evaluated and shows the highest accuracy (R^2 score).

  Step 8: Support Vector Regressor (SVR)
  A Support Vector Regressor is built using `SVR(kernel='linear')` and tested on both original and scaled datasets.

4. Evaluation
  - For each model, evaluation metrics such as R^2 score are calculated for both training and test sets.
  - Models' performance is compared to identify the best model for the project.

5. Documentation Summary
   - Each step in the notebook is focused on loading, preprocessing, model building, and evaluating multiple regression models.
   - Decision Tree shows the best performance for this particular dataset.

Let me know if you'd like more details on specific steps or documentation!