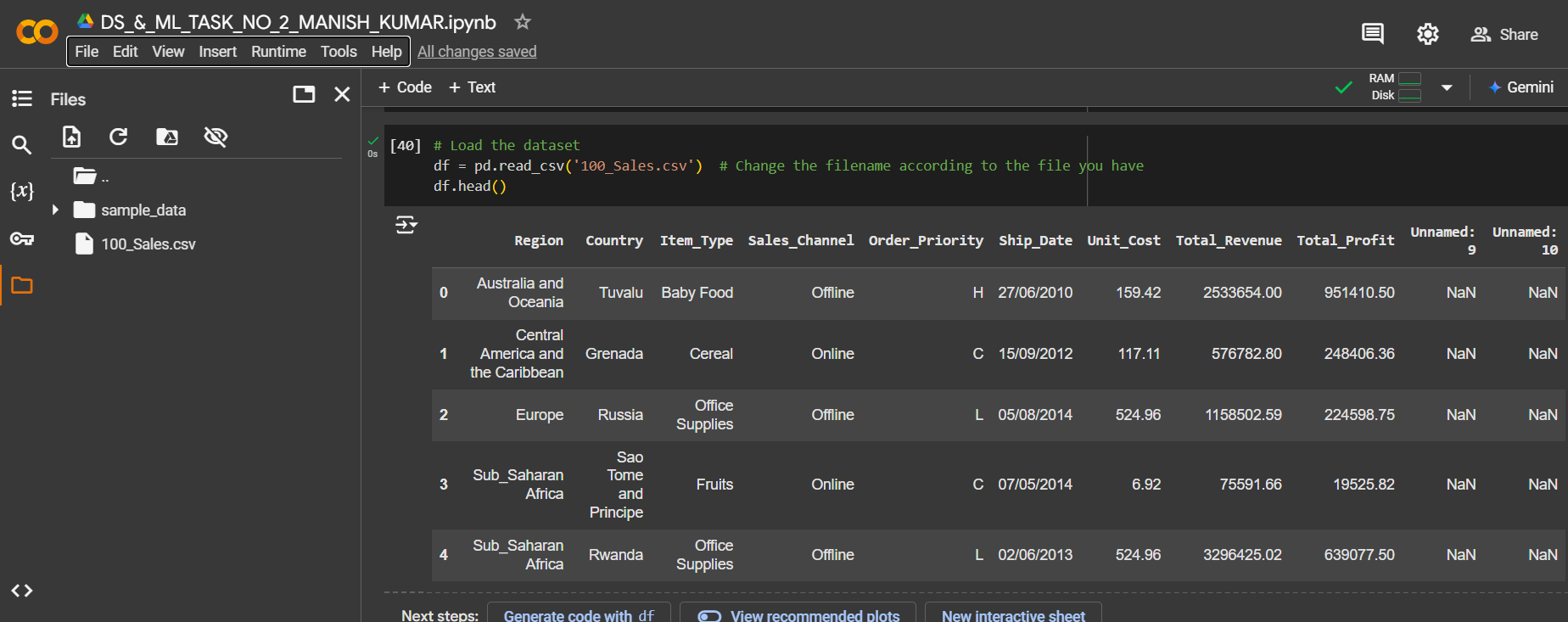
1. **Task Description**

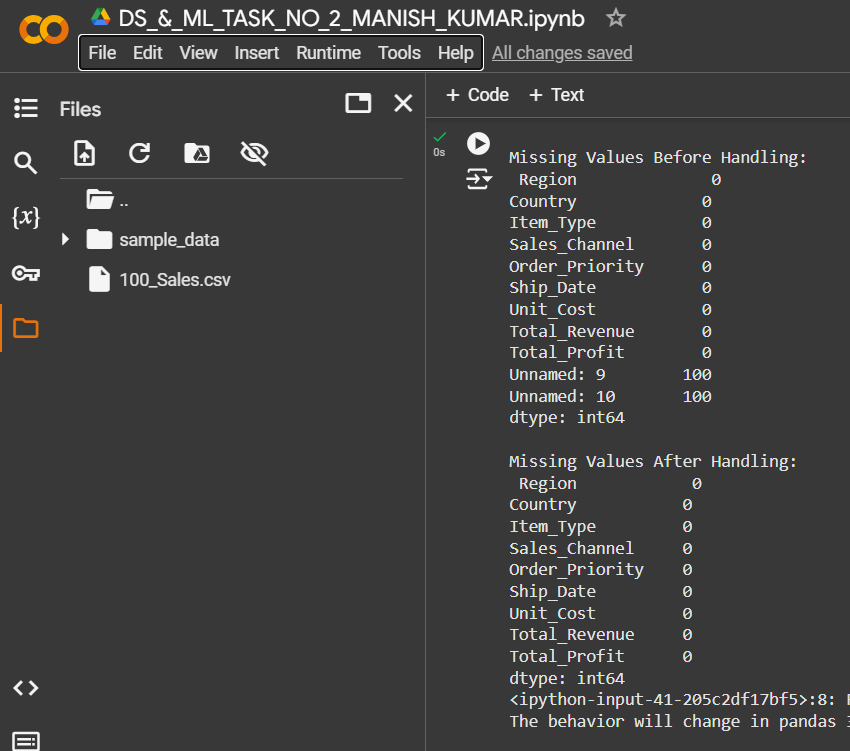
* The objective of this task is to apply two machine learning algorithms, Decision Tree and Random Forest, to predict the target variable Total\_Revenue in the provided dataset.
* This dataset contains various features such as Region, Country, Item\_Type, Unit\_Cost, Total\_Profit, and others that can influence sales revenue. The task involves several stages, including data preprocessing (handling missing values and encoding categorical variables), feature engineering, and model training.
* We will train both Decision Tree and Random Forest models to predict Total\_Revenue and evaluate their performance using appropriate regression metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R²).
* Additionally, model optimization and evaluation will help determine the more accurate model for this prediction task.

1. **Attach Screenshot of Output**

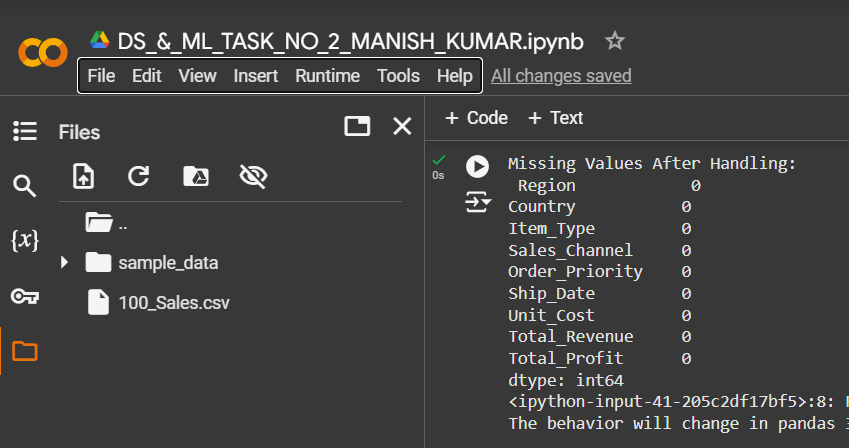
* **Dataset Preview:**



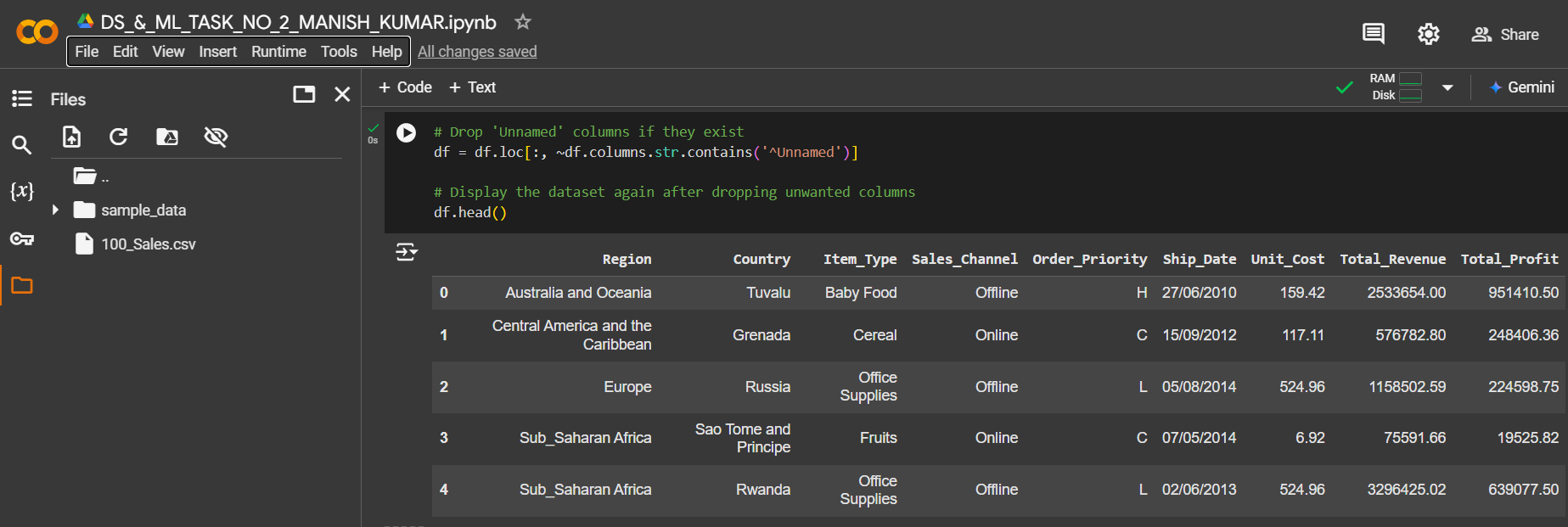
* **Missing values before handling:**

****

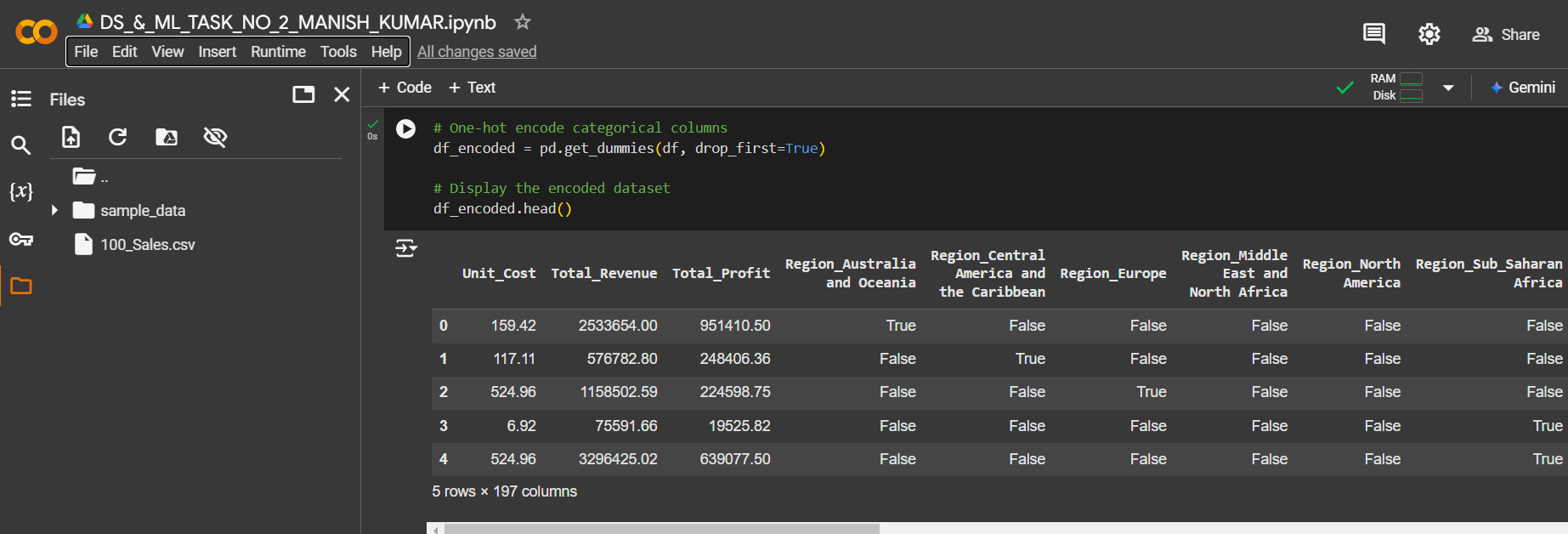
* **Missing values after handling:**



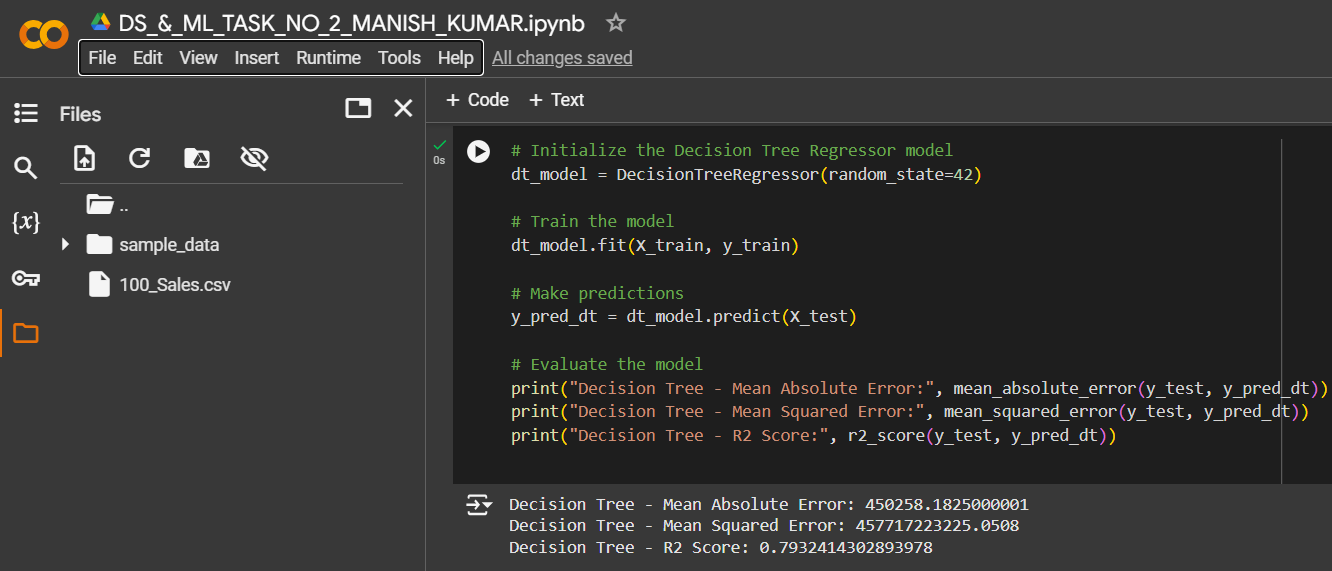
* **Drop unnamed columns:**



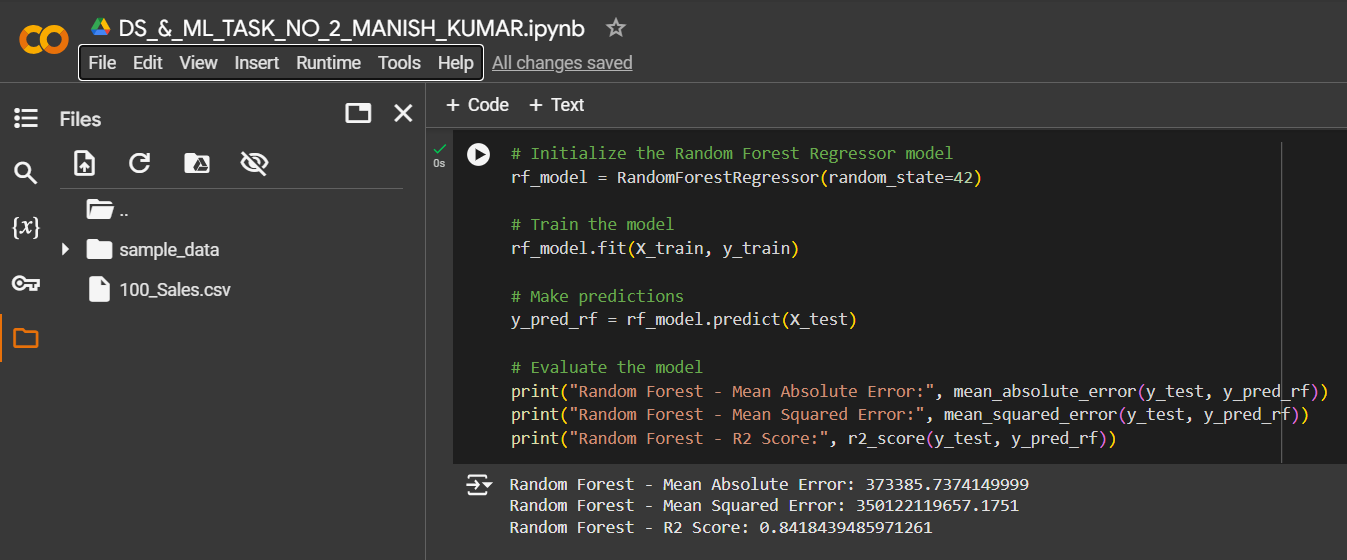
* **Encode categorical columns:**



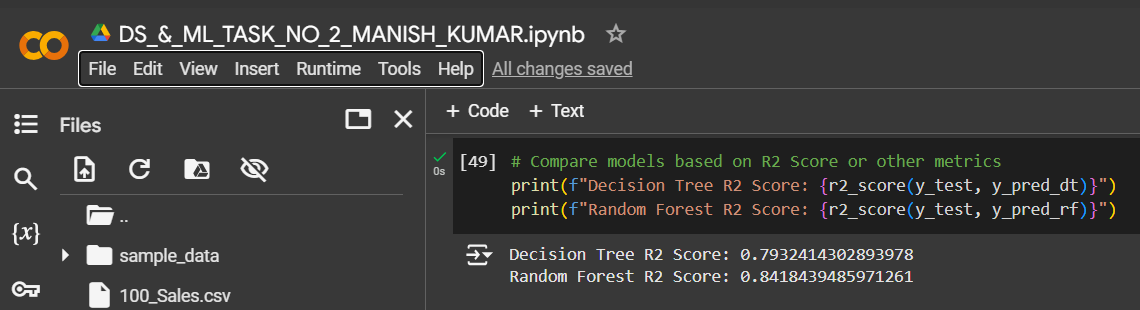
* **Decision Tree Model:**



* **Random Forest Model:**

****

* **Model comparison:**

****

1. **Describe Widget/Algorithm Used in Task**

**Algorithms Used:**

* **Decision Tree Regressor**: A non-linear model used for regression tasks that splits the data based on feature values to predict the target. It's interpretable and works well on both categorical and continuous data.
  + **Process**: The algorithm divides the dataset into subsets based on the best splits to predict a continuous target.
* **Random Forest Regressor**: An ensemble method that creates multiple decision trees and averages their predictions to improve performance and reduce overfitting.
  + **Process**: Random Forest builds multiple decision trees and outputs the average prediction from all trees. It's a more robust algorithm compared to a single decision tree.

**Steps Involved:**

1. **Data Preprocessing**:
   * Load the dataset and inspect it for missing values.
   * Handle missing data by filling numeric columns with mean or median values.
   * Drop columns that contain completely missing data or are not relevant (e.g., "Unnamed" columns).
2. **Feature Engineering**:
   * Apply one-hot encoding to categorical columns (such as Region, Country).
3. **Model Training**:
   * Split the dataset into training and testing sets (80%-20%).
   * Train Decision Tree and Random Forest models on the training data.
4. **Model Evaluation**:
   * Use metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R2) to evaluate both models.

**Libraries/Tools Used:**

* **Pandas**: Utilized for data manipulation and cleaning, including handling missing values, encoding categorical variables, and splitting datasets.
* **Scikit-Learn**: This library is used for implementing machine learning models such as Decision Tree and Random Forest, as well as evaluating their performance using regression metrics like MAE, MSE, and R².
* **NumPy**: Employed for numerical operations such as handling missing values with mean imputation and performing mathematical calculations during model evaluation.

**\*\*\* The End \*\*\***