2025

Home Project 2

Data pre-processing with Python

DANIEL DADZIE APPIAH | 156801227 | DDAPPIAH

2ND APRIL, 2025.

Table of Contents

1 Introduction 2

2 0-Screen for Starting Project in Google Colab 2

3 Step 1: Load Data in Pandas 3

4 Display of headers 5

5 Explore Data: data format below 5

6 Check for Missing Values 6

7 Step 2: Drop Unnecessary Columns and Validating result after deleting the unnecessary columns 6

8 Step 2: Validating to see the deleted columns 7

9 Step 3: Drop Rows with Missing Values 7

10 Step 3: Drop Rows (handling) With Missing Values and check dataset size after dropping missing values 8

10.1 Explanation for Result 8

11 Step 4: Create Dummy Variables and Check Transformed Dataset 8

12 Step 5: Interpolate Missing Numeric Values and Confirm Missing Values Handled 9

12.1 Explanation of Results 9

13 Convert the Data Frame to NumPy and show the effect of conversion to NumPy 10

13.1 Summary & Explanation of Results 11

14 Step 7: Split Dataset into Training and Test Sets 12

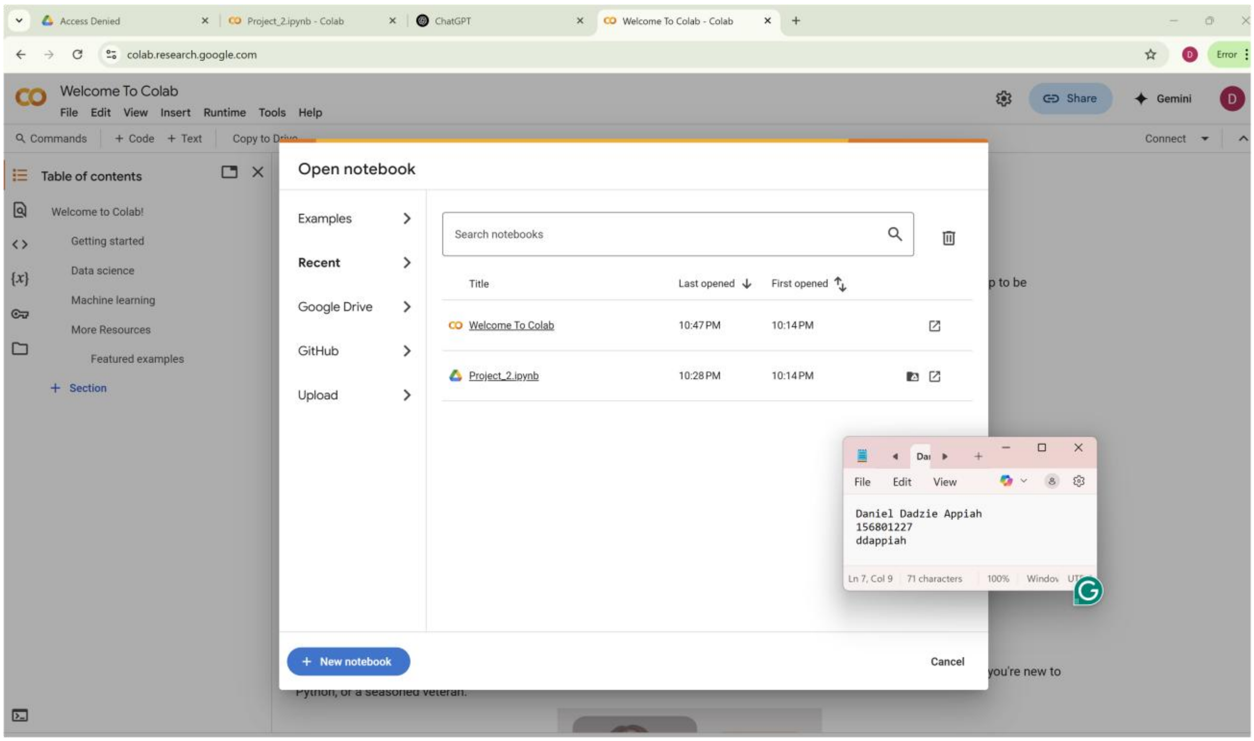
14.1 Explanation of the Results 12

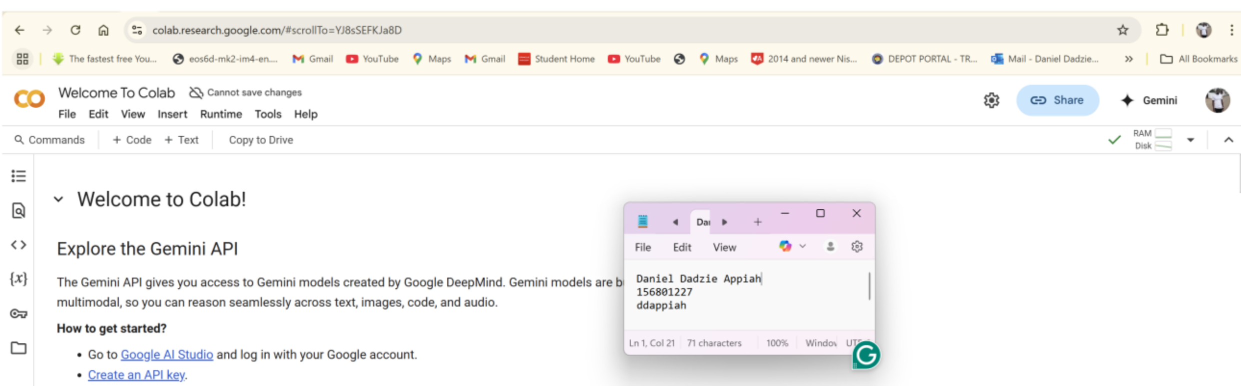
1 Introduction

Data pre-processing is a crucial step in data analysis and machine learning, ensuring that raw data is cleaned, transformed, and structured for accurate and efficient modeling. Real-world data is often incomplete, inconsistent, and noisy, making pre-processing essential for improving data quality and optimizing model performance.

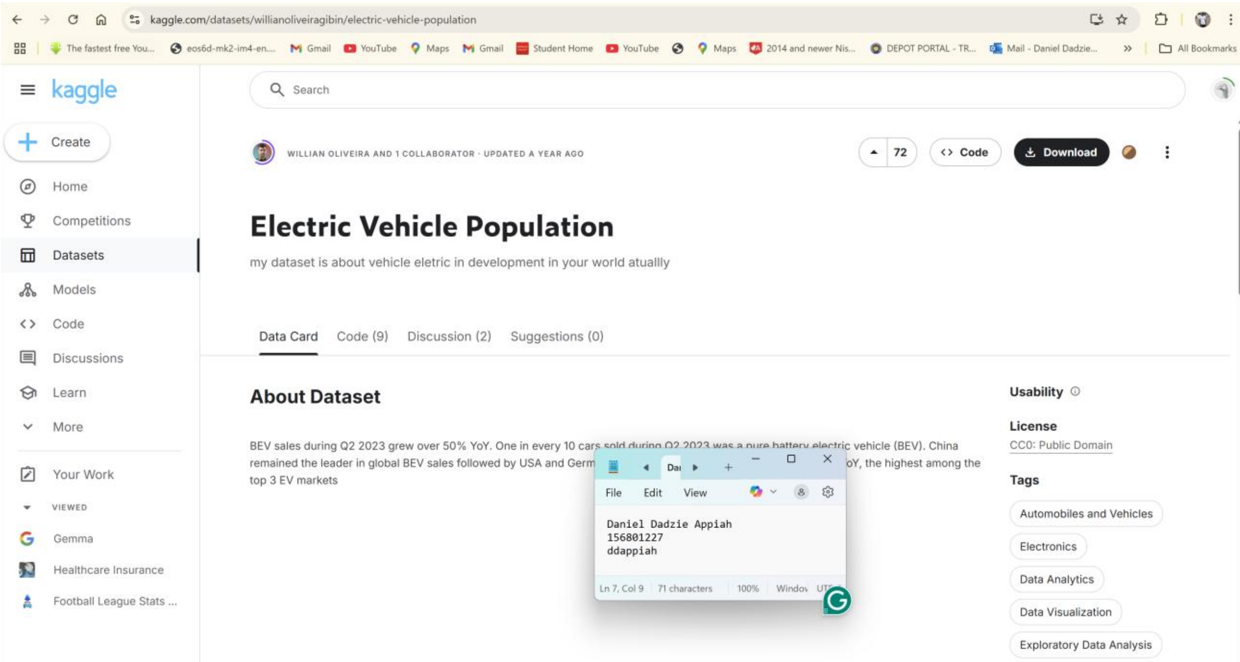
2 0-Screen for Starting Project in Google Colab

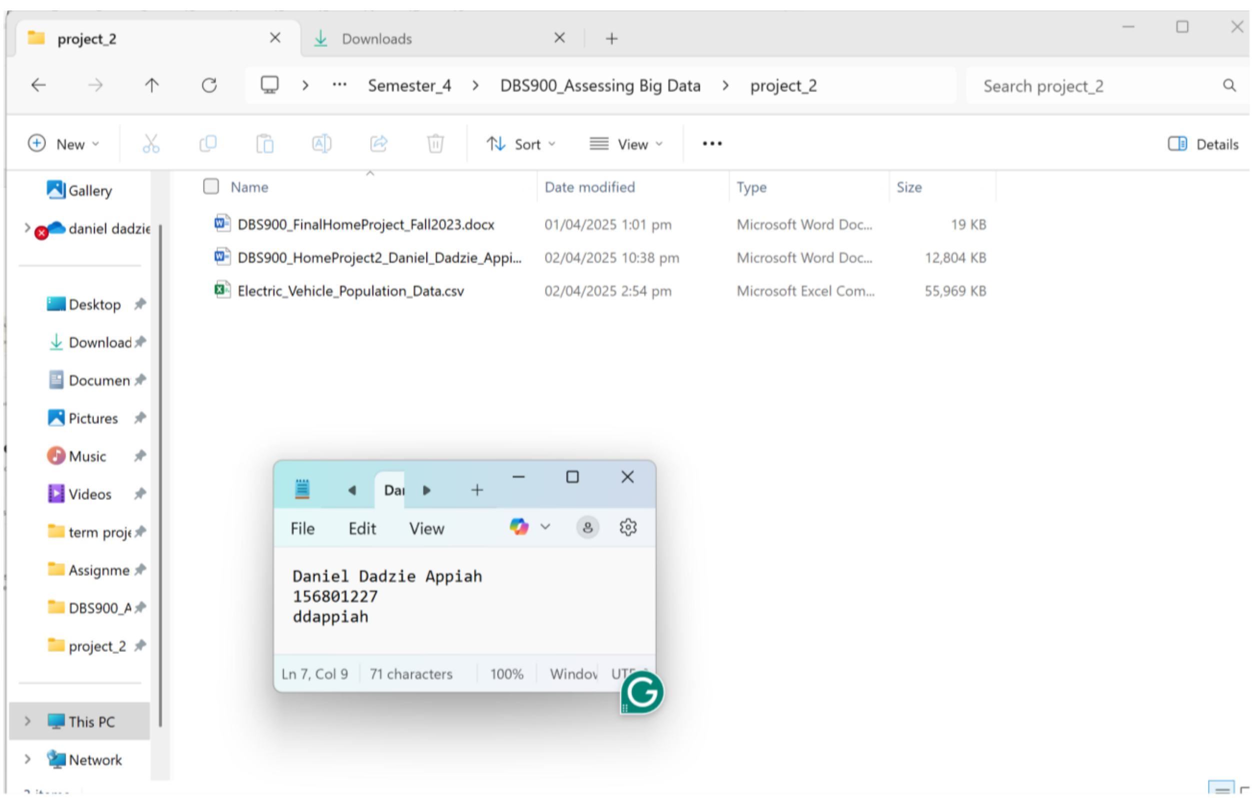
Starting up the command line as an administrator. Use of Windows Terminal.

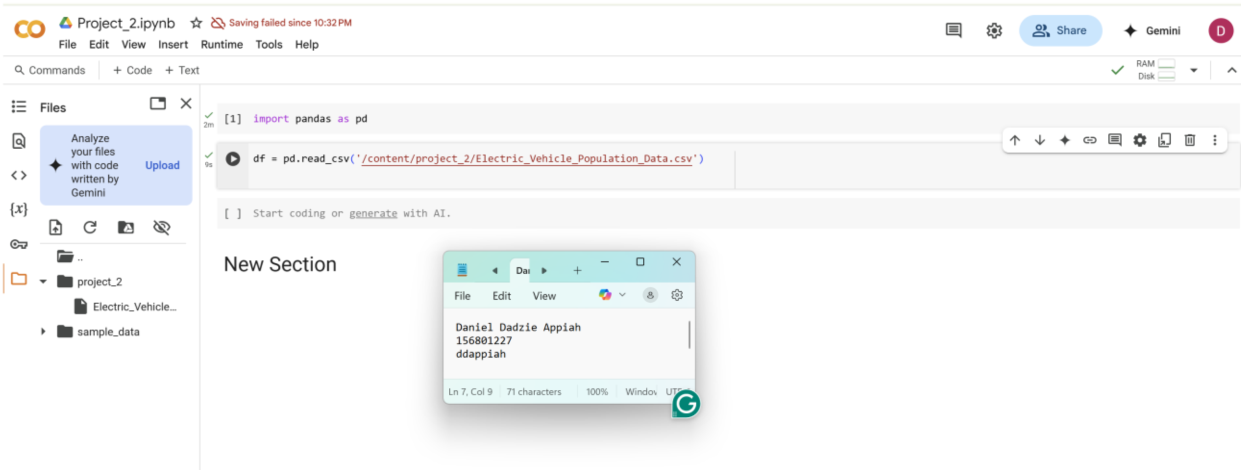




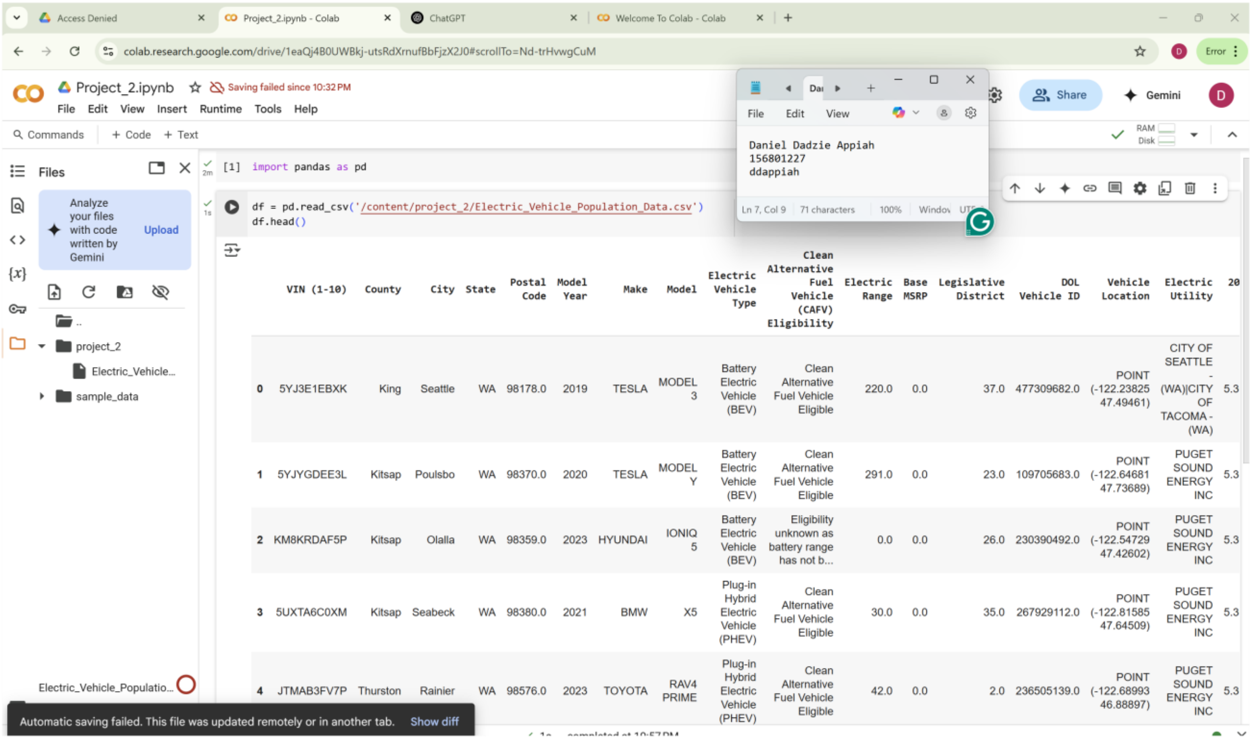
3 Step 1: Load Data in Pandas



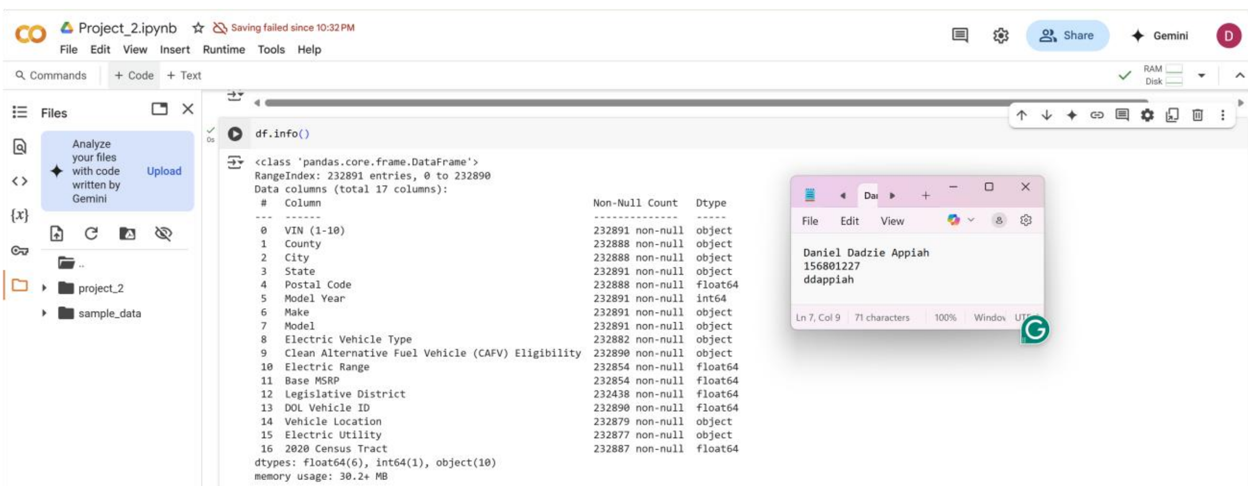



Project_2.ipynb
Saving failed since 10:32 PM
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
+ Code
+ Text
RAM
Disk
Files
Analyze your files with code written by Gemini
Upload
project_2
Electric_Vehicle...
sample_data
[1] import pandas as pd
df = pd.read_csv('/content/project_2/Electric_Vehicle_Population_Data.csv')
[ ] Start coding or generate with AI.
New Section
File Edit View
Daniel Dadzie Appiah
156801227
ddappiah
Ln 7, Col 9
71 characters
100%

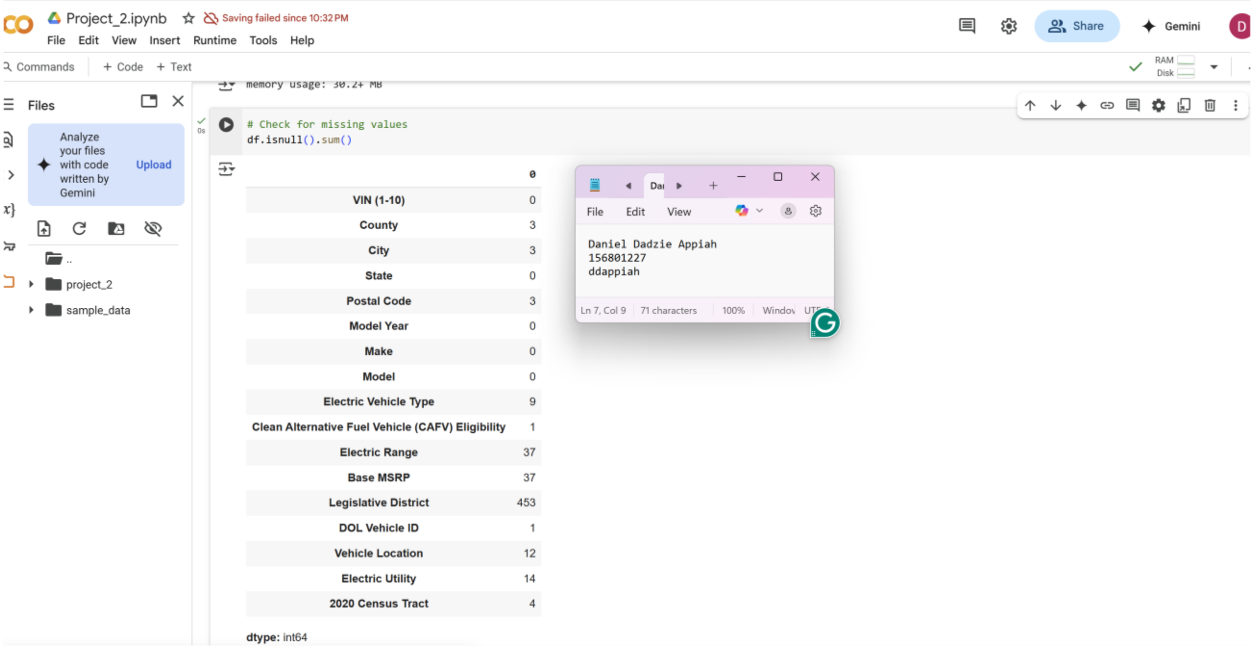
4 Display of headers



5 Explore Data: data format below

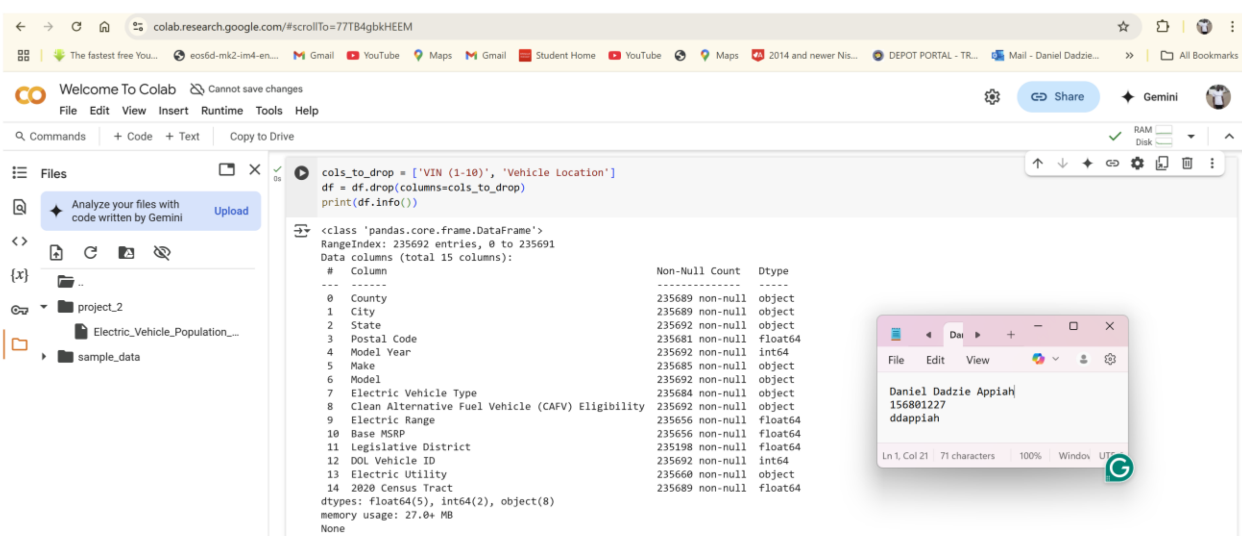


6 Check for Missing Values

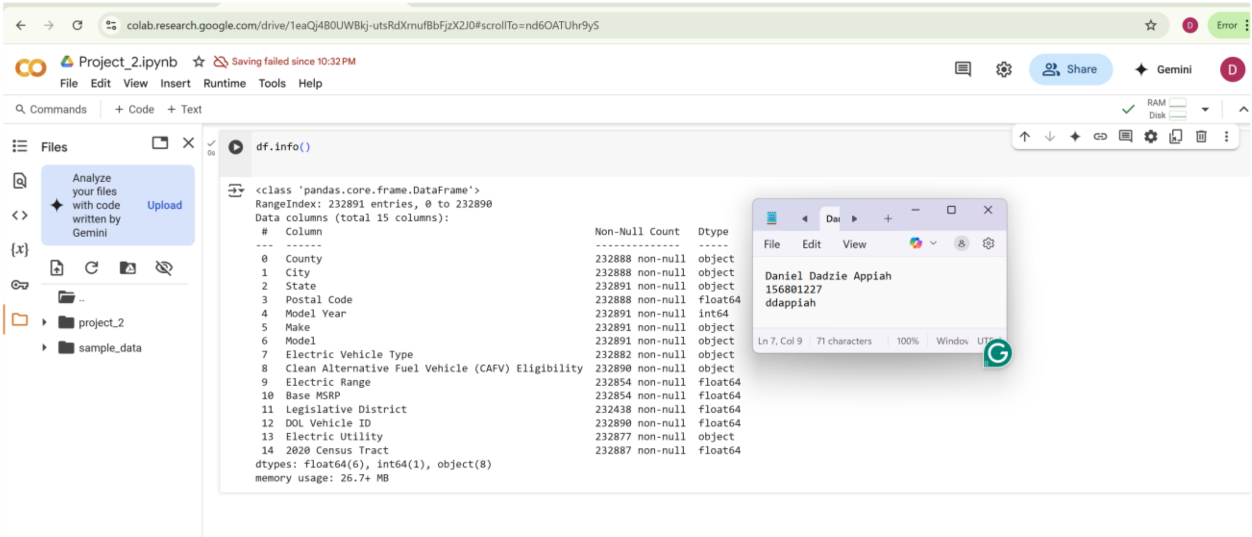


7 Step 2: Drop Unnecessary Columns and Validating result after deleting the unnecessary columns

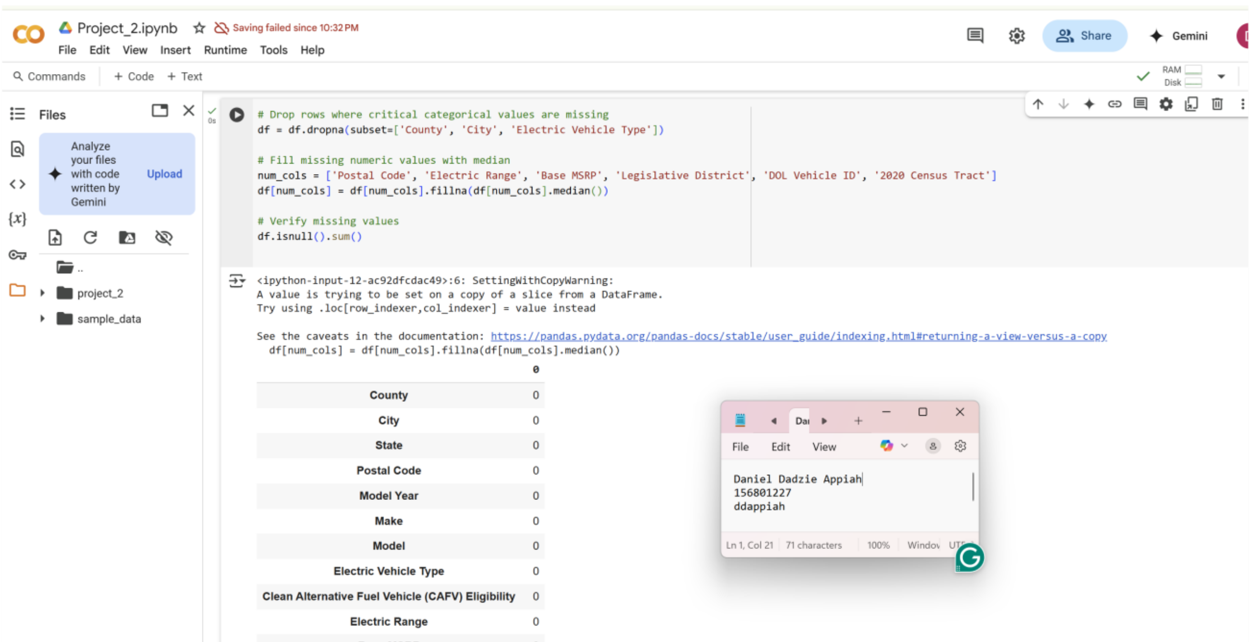
Issues the command twice. I too picture after it has been deleted



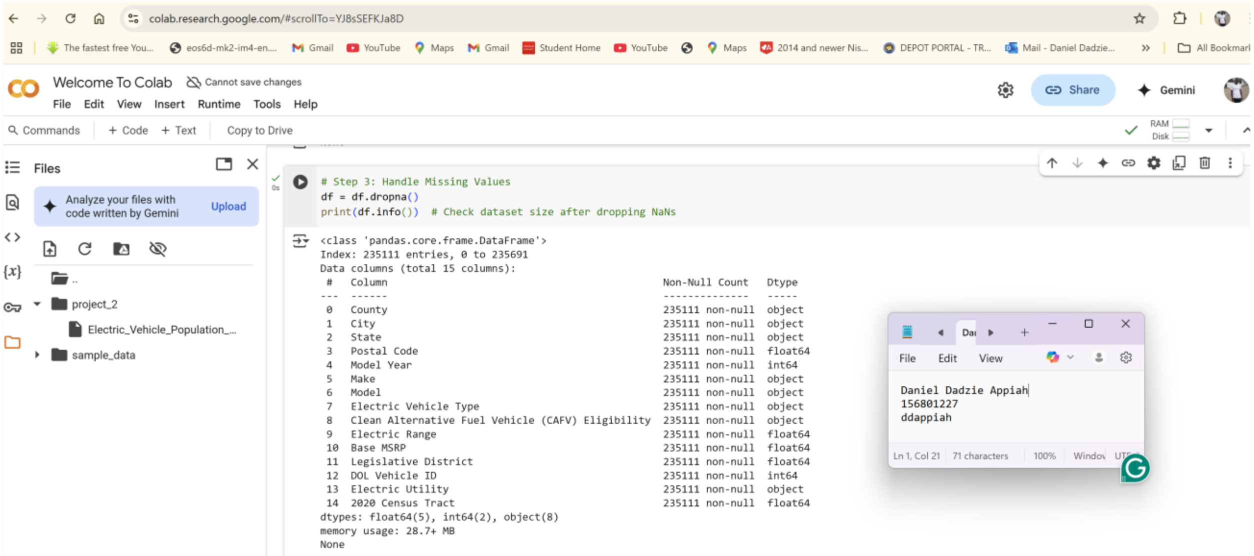
8 Step 2: Validating to see the deleted columns



9 Step 3: Drop Rows with Missing Values



10 Step 3: Drop Rows (handling) With Missing Values and check dataset size after dropping missing values

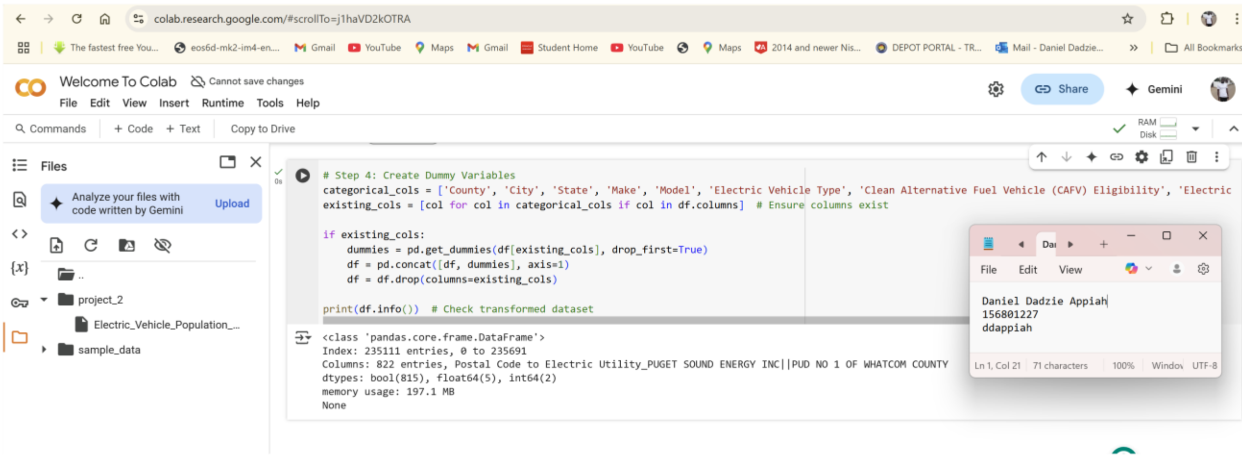


10.1 Explanation for Result

This line uses Pandas' dropna() method to remove any row in the DataFrame df that contains at least one missing value (NaN).
This prints an updated summary of the DataFrame after dropping rows with missing values.

11 Step 4: Create Dummy Variables and Check Transformed Dataset

12 Step 5: Interpolate Missing Numeric Values and Confirm Missing Values Handled



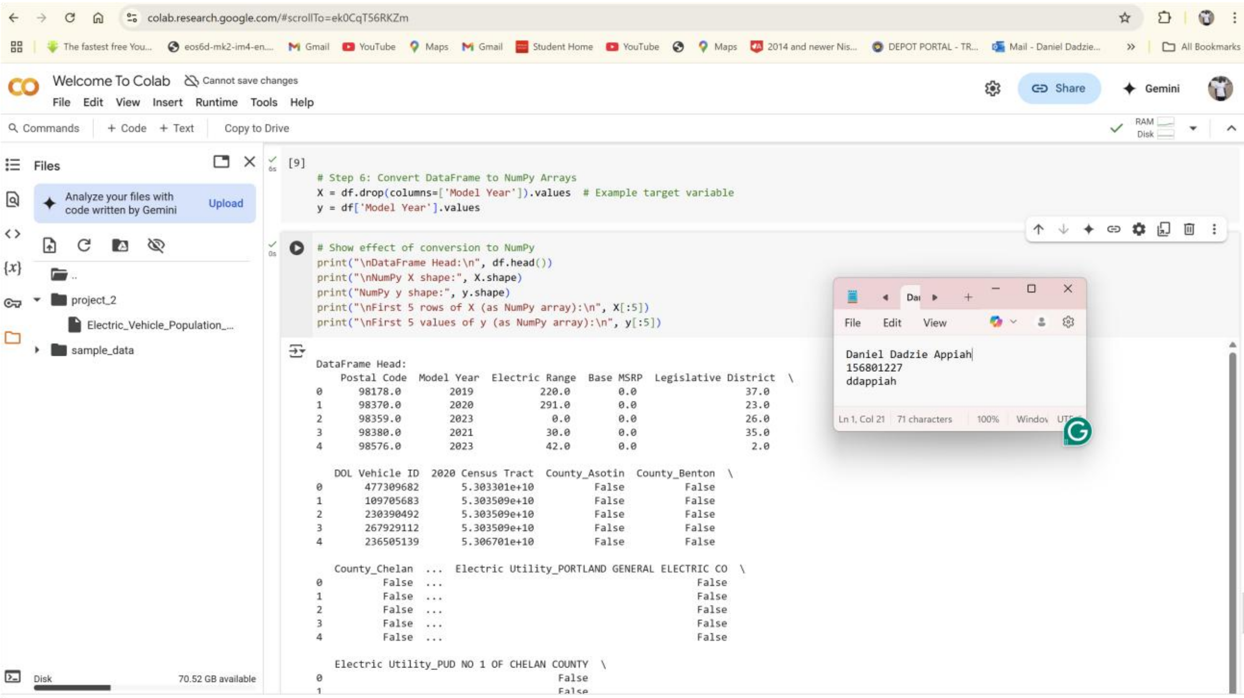
12.1 Explanation of Results

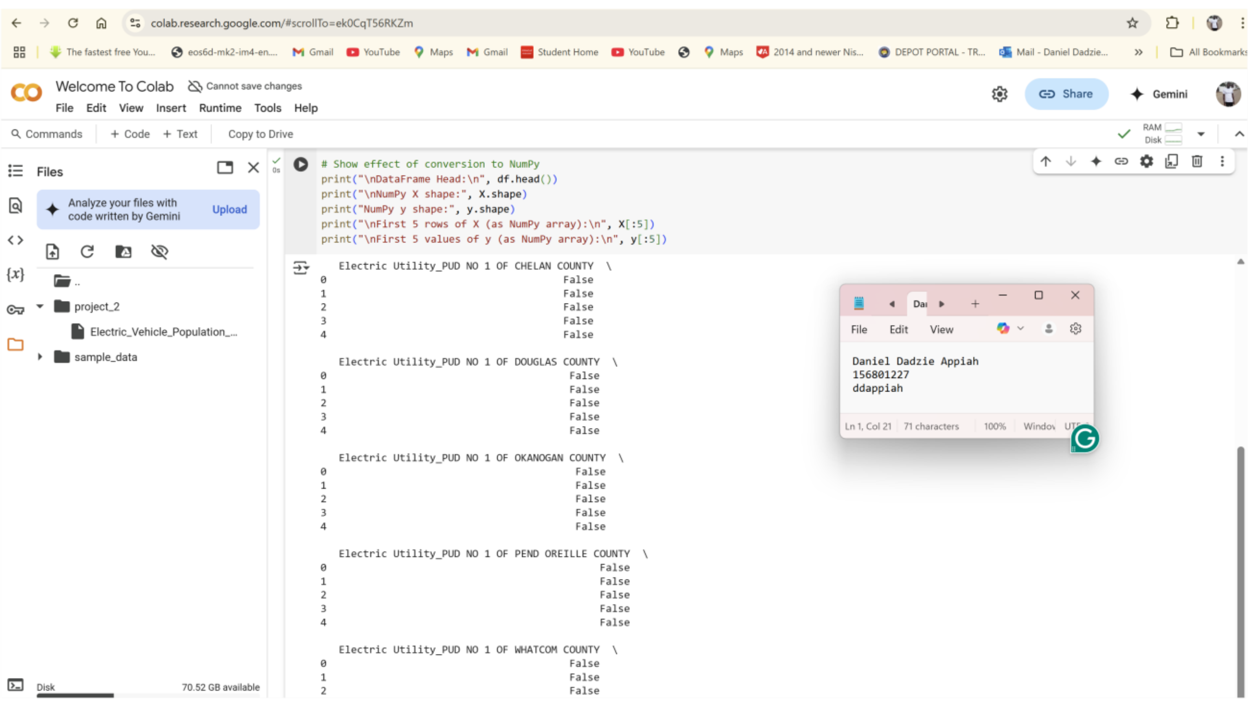
The dataset contains 235,111 entries. After one-hot encoding categorical variables, the total number of columns increased to 822. The dataset now has:

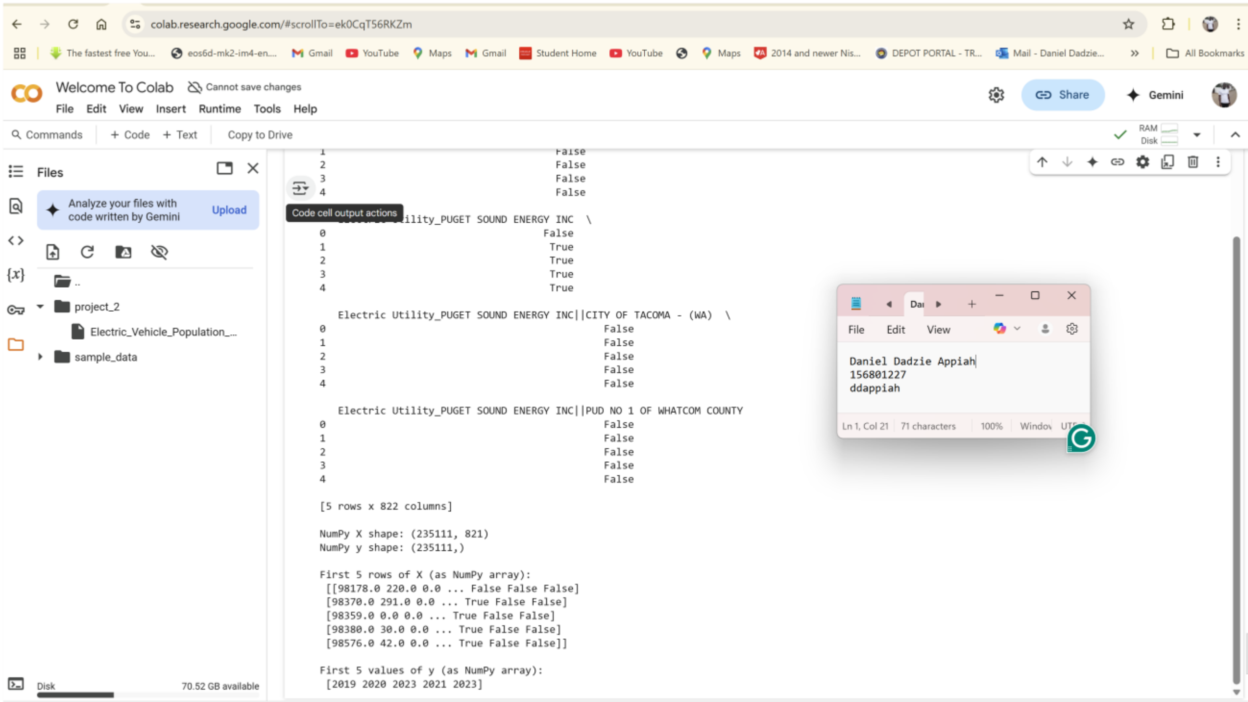
- 815 boolean columns (dummy variables for categorical features).
- 5 float64 columns (numeric data, including interpolated missing values).
- 2 int64 columns (discrete numerical values).
- The memory usage is 197.1 MB, indicating a significant increase due to the expansion of dummy variables.

This transformation ensures that the dataset is fully numerical and ready for machine learning.

13 Convert the Data Frame to NumPy and show the effect of conversion to NumPy







13.1 Summary & Explanation of Results

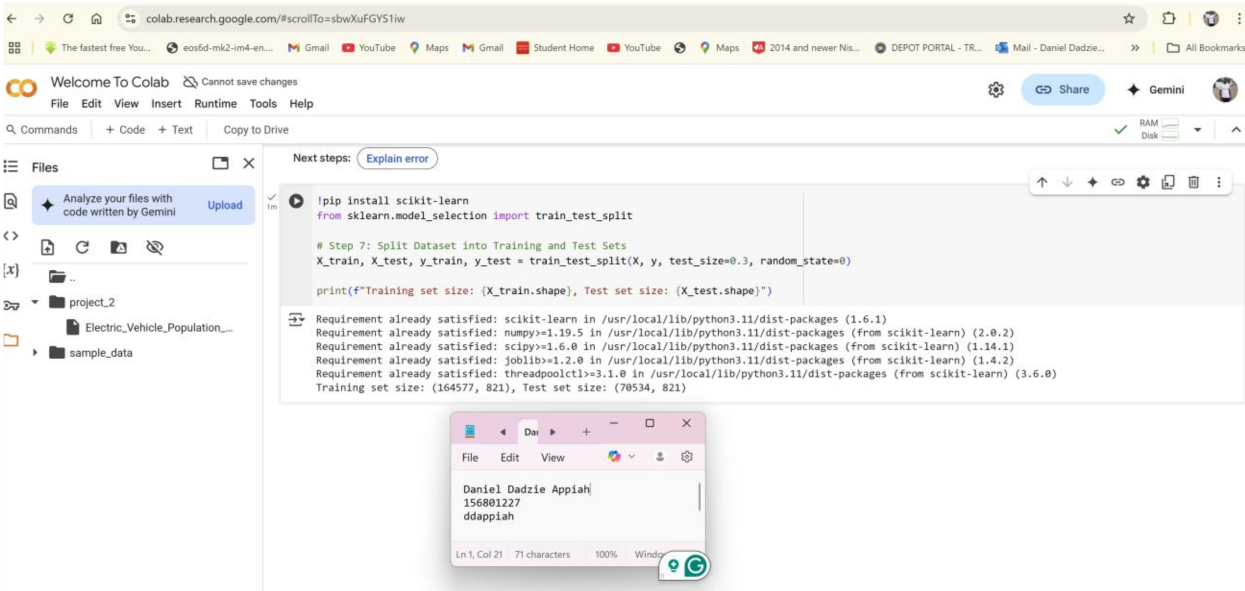
DataFrame Overview:
The dataset initially contained multiple columns, including Postal Code, Model Year, Electric Range, Base MSRP and many more. Unnecessary columns (DOL Vehicle ID) were removed.
Missing values were dropped or interpolated for numerical columns.

Effect of Converting to NumPy Arrays:
X (features) and y (target variable: Model Year) were extracted from the DataFrame.
The conversion to NumPy arrays results in:
X shape: (235,111 rows, 821 features), y shape: (235,111 rows, 1 target variable)

Verifying the Conversion:
The first 5 rows of X show numerical and categorical (dummy) variables converted to True/False values. The first 5 values of y confirm that the Model Year is correctly extracted.

Splitting Data into Training & Test Sets:
The dataset was split into:
Training set: 70% of the data (164,577 rows, 821 features). Test set: 30% of the data (70,534 rows, 821 features). This ensures a balanced dataset for machine learning models.

14 Step 7: Split Dataset into Training and Test Sets



14.1 Explanation of the Results

Data Preprocessing Steps:
The dataset was loaded from Google Drive and checked for missing values.
Irrelevant columns (DOL Vehicle ID) were dropped. Rows with missing values were removed.
Categorical variables (County, City, State) were converted into dummy variables.
Missing numeric values (Postal Code, Electric Range) were interpolated.

Conversion to NumPy Arrays:
The df DataFrame was converted into two NumPy arrays: X (features), which contains all columns except Model Year. y (target variable) contains only the Model Year. The shapes of X and Y confirm a successful conversion.

Dataset Splitting for Machine Learning:
The dataset was split into:
Training set : 70% of the data (X_train and y_train) : 164,577 samples with 821 features
Test set : 30% of the data (X_test and y_test) : 70,534 samples with 821 features
The split ensures the model has enough data for training and evaluation.