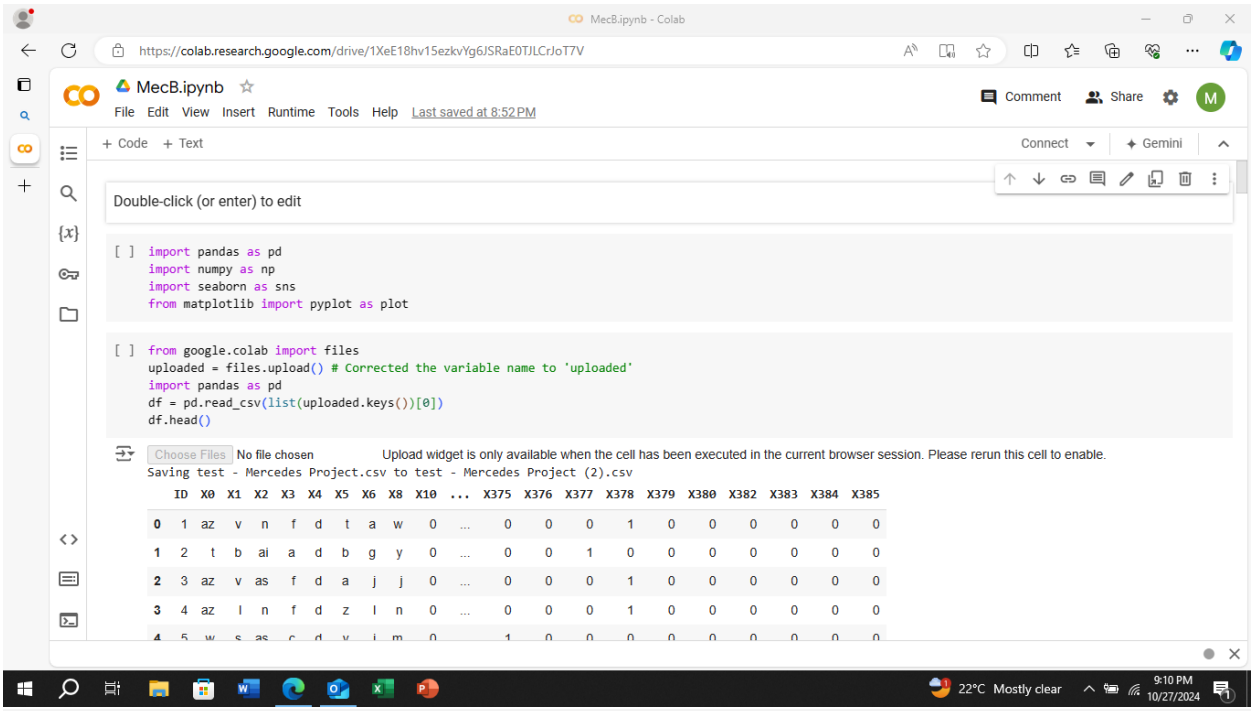
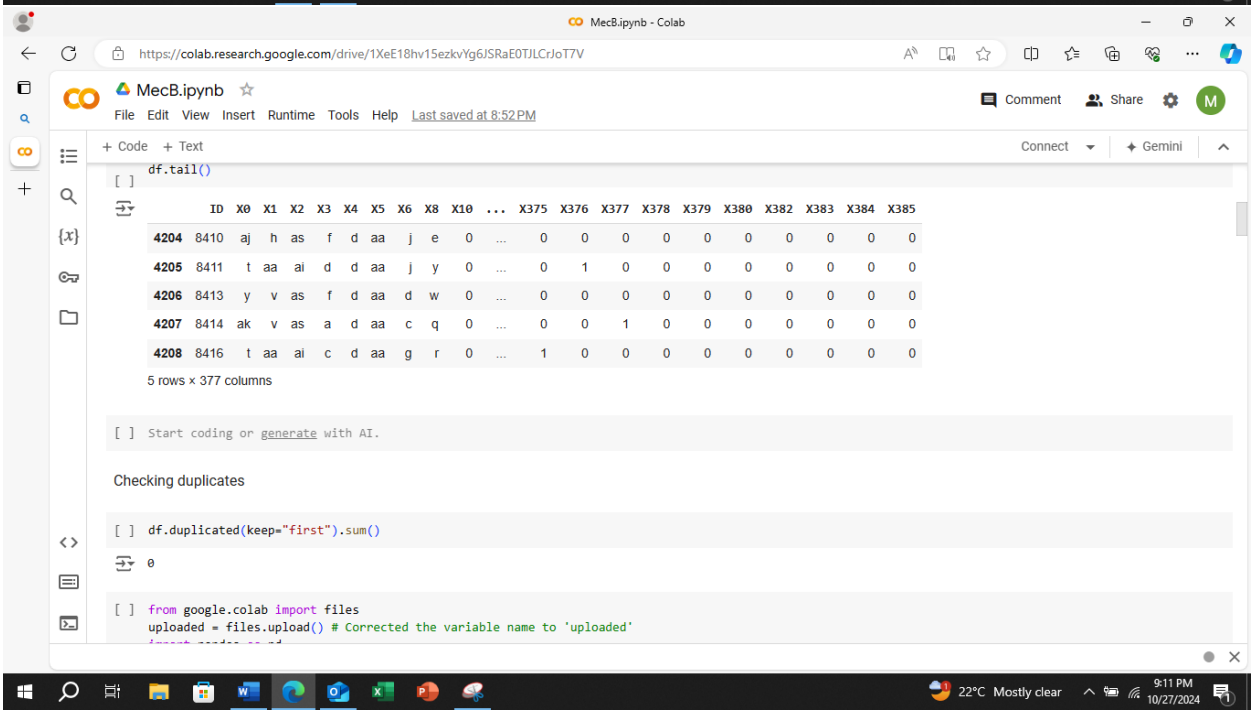
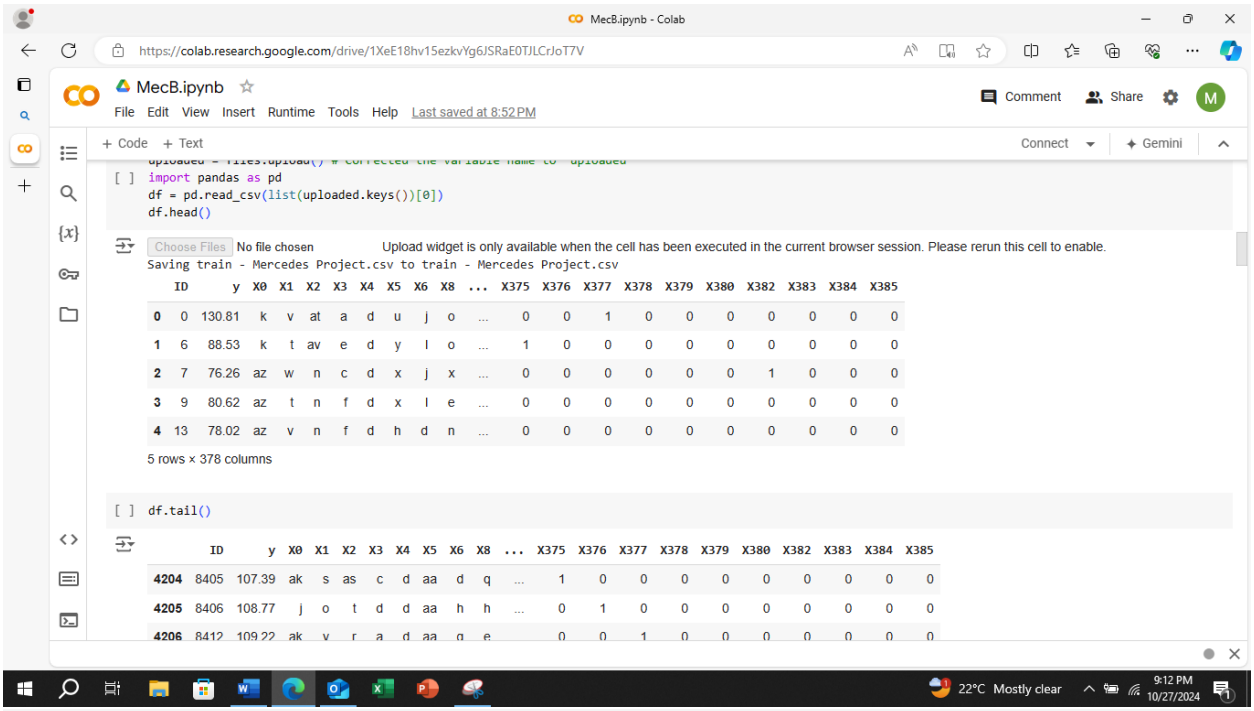
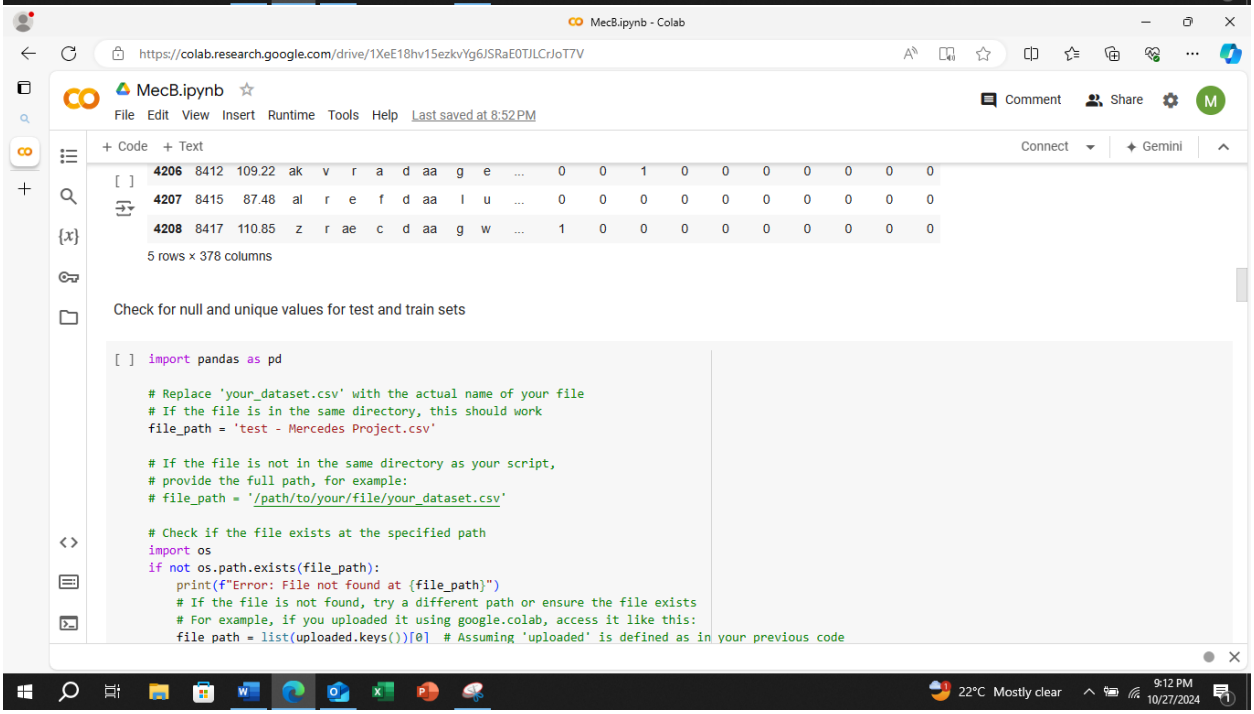
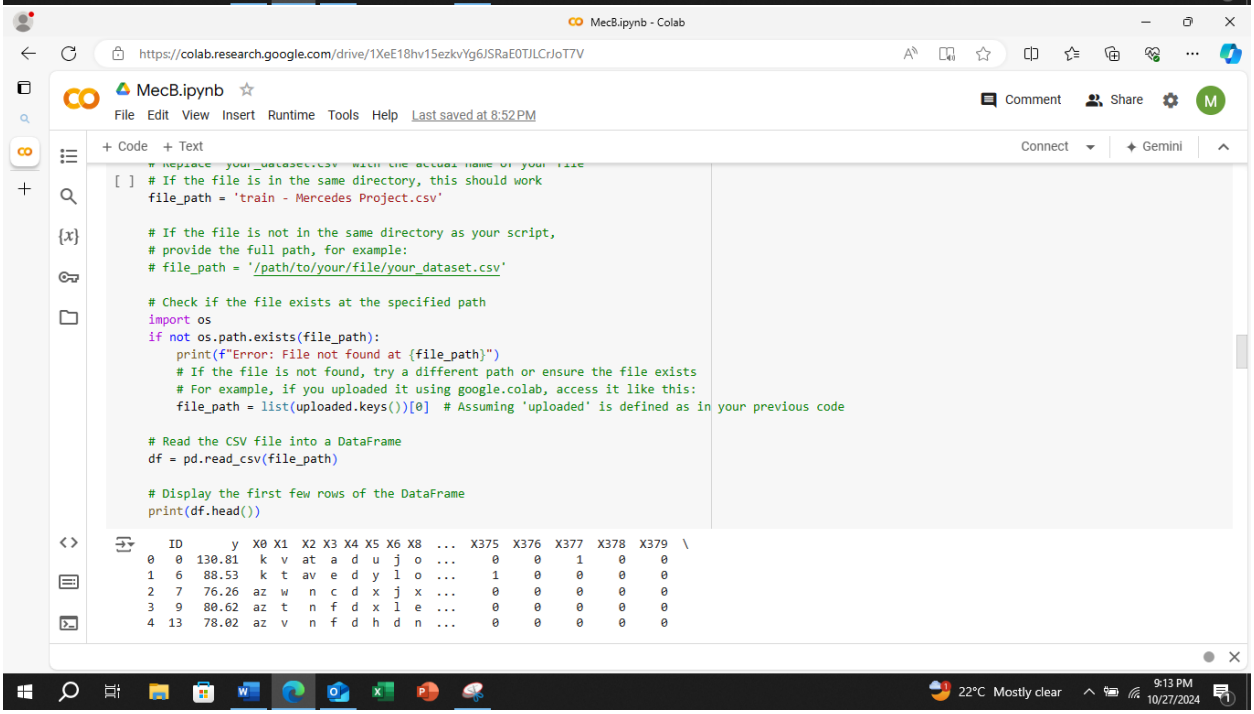

MecB.ipynb - Colab
https://colab.research.google.com/drive/1XeE18hv15ezkvYg6JSRaE0TJLCrJoT7V
MecB.ipynb
File Edit View Insert Runtime Tools Help Last saved at 8:52 PM
Comment
Share
+ Code + Text
Connect
Gemini
Double-click (or enter) to edit
import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plot
from google.colab import files
uploaded = files.upload() # Corrected the variable name to 'uploaded'
import pandas as pd
df = pd.read_csv(list(uploaded.keys())[0])
df.head()
Choose Files No file chosen
Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving test - Mercedes Project.csv to test - Mercedes Project (2).csv
ID X0 X1 X2 X3 X4 X5 X6 X8 X10 ... X375 X376 X377 X378 X379 X380 X382 X383 X384 X385
0 1 az v n f d t a w 0 ... 0 0 0 1 0 0 0 0 0 0
1 2 t b ai a d b g y 0 ... 0 0 1 0 0 0 0 0 0 0
2 3 az v as f d a j j 0 ... 0 0 0 1 0 0 0 0 0 0
3 4 az l n f d z l n 0 ... 0 0 0 1 0 0 0 0 0 0
22°C Mostly clear
9:10 PM
10/27/2024

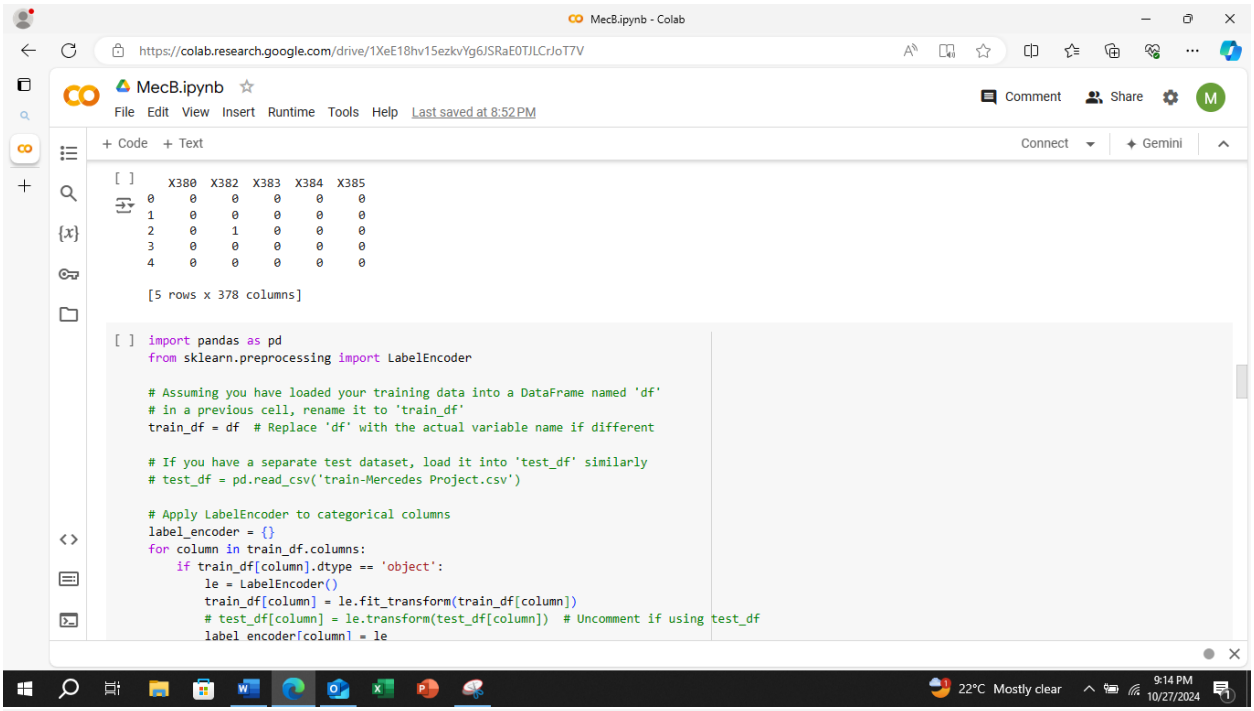

MecB.ipynb - Colab
https://colab.research.google.com/drive/1XeE18hv15ezkvYg6JSRaE0TJLCrJoT7V
MecB.ipynb
File Edit View Insert Runtime Tools Help Last saved at 8:52 PM
Comment
Share
+ Code + Text
Connect
Gemini
df.tail()
ID X0 X1 X2 X3 X4 X5 X6 X8 X10 ... X375 X376 X377 X378 X379 X380 X382 X383 X384 X385
4204 8410 aj h as f d aa j e 0 ... 0 0 0 0 0 0 0 0 0 0
4205 8411 t aa ai d d aa j y 0 ... 0 1 0 0 0 0 0 0 0 0
4206 8413 y v as f d aa d w 0 ... 0 0 0 0 0 0 0 0 0 0
4207 8414 ak v as a d aa c q 0 ... 0 0 1 0 0 0 0 0 0 0
4208 8416 t aa ai c d aa g r 0 ... 1 0 0 0 0 0 0 0 0 0
5 rows × 377 columns
Start coding or generate with AI.
Checking duplicates
df.duplicated(keep="first").sum()
0
from google.colab import files
uploaded = files.upload() # Corrected the variable name to 'uploaded'
22°C Mostly clear
9:11 PM
10/27/2024

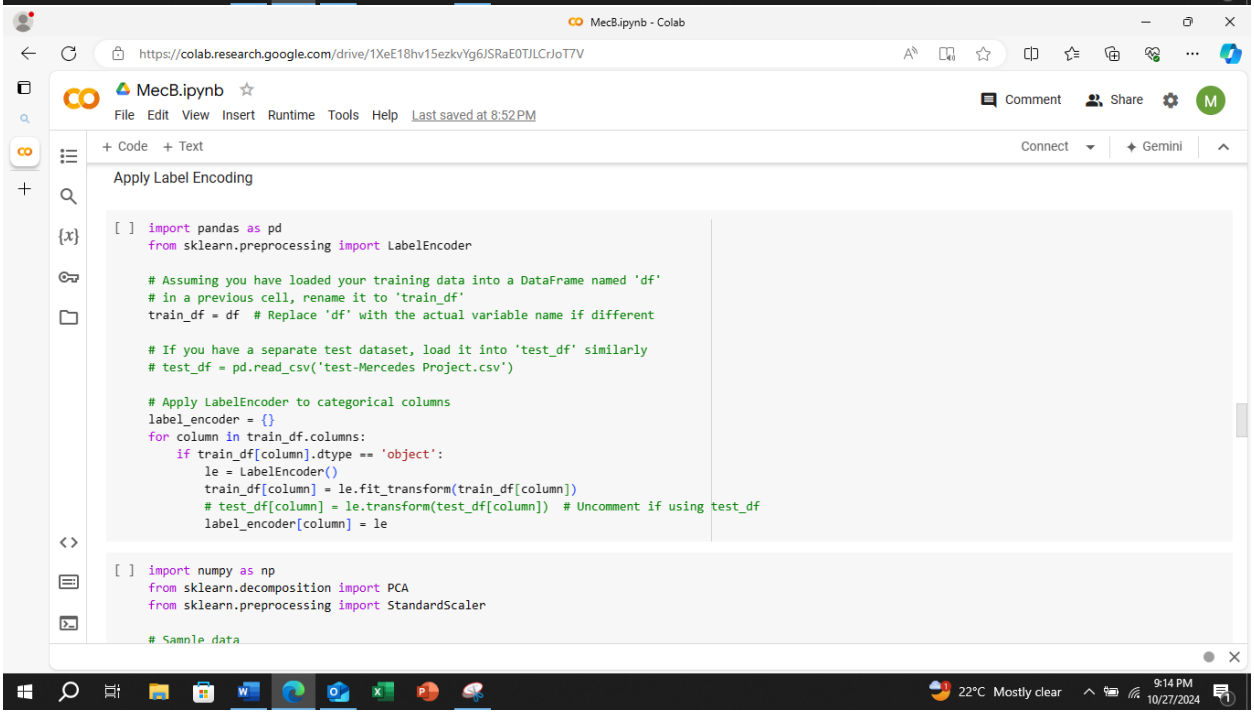

MecB.ipynb - Colab
https://colab.research.google.com/drive/1XeE18hv15ezkvYg6JSRaE0TJLCrJoT7V
MecB.ipynb
File Edit View Insert Runtime Tools Help Last saved at 8:52 PM
Comment
Share
+ Code + Text
Connect
Gemini
[ ] import pandas as pd
df = pd.read_csv(list(uploaded.keys())[0])
df.head()
Choose Files No file chosen
Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving train - Mercedes Project.csv to train - Mercedes Project.csv
ID y X0 X1 X2 X3 X4 X5 X6 X8 ... X375 X376 X377 X378 X379 X380 X382 X383 X384 X385
0 0 130.81 k v at a d u j o ... 0 0 1 0 0 0 0 0 0 0
1 6 88.53 k t av e d y l o ... 1 0 0 0 0 0 0 0 0 0
2 7 76.26 az w n c d x j x ... 0 0 0 0 0 0 1 0 0 0
3 9 80.62 az t n f d x l e ... 0 0 0 0 0 0 0 0 0 0
4 13 78.02 az v n f d h d n ... 0 0 0 0 0 0 0 0 0 0
5 rows × 378 columns
[ ] df.tail()
ID y X0 X1 X2 X3 X4 X5 X6 X8 ... X375 X376 X377 X378 X379 X380 X382 X383 X384 X385
4204 8405 107.39 ak s as c d aa d q ... 1 0 0 0 0 0 0 0 0 0
4205 8406 108.77 j o t d d aa h h ... 0 1 0 0 0 0 0 0 0 0
4206 8412 109.22 ak v r a d aa g e ... 0 0 1 0 0 0 0 0 0 0
22°C Mostly clear
9:12 PM
10/27/2024

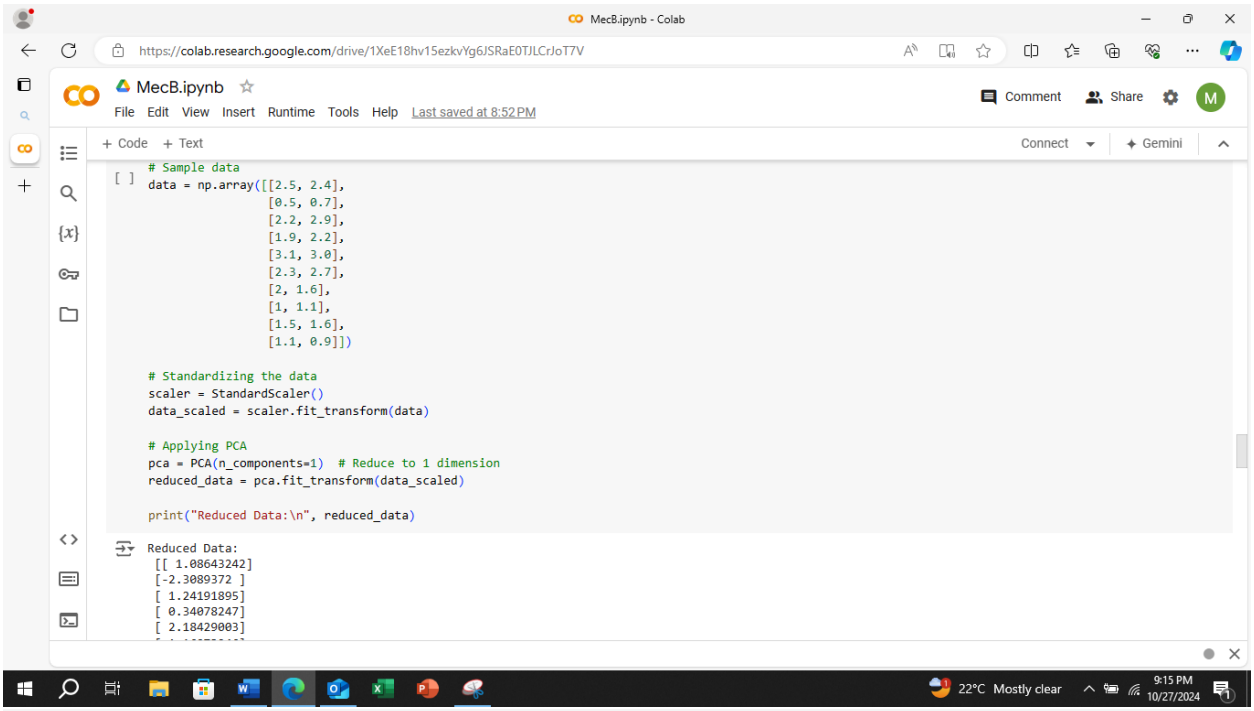

MecB.ipynb - Colab
https://colab.research.google.com/drive/1XeE18hv15ezkvYg6JSRaE0TJLCrJoT7V
MecB.ipynb
File Edit View Insert Runtime Tools Help Last saved at 8:52 PM
Comment
Share
+ Code + Text
Connect
Gemini
[ ] 4206 8412 109.22 ak v r a d aa g e ... 0 0 1 0 0 0 0 0 0 0
4207 8415 87.48 al r e f d aa l u ... 0 0 0 0 0 0 0 0 0 0
4208 8417 110.85 z r ae c d aa g w ... 1 0 0 0 0 0 0 0 0 0
5 rows × 378 columns
Check for null and unique values for test and train sets
[ ] import pandas as pd
# Replace 'your_dataset.csv' with the actual name of your file
# If the file is in the same directory, this should work
file_path = 'test - Mercedes Project.csv'
# If the file is not in the same directory as your script,
# provide the full path, for example:
# file_path = '/path/to/your/file/your_dataset.csv'
# Check if the file exists at the specified path
import os
if not os.path.exists(file_path):
print(f"Error: File not found at {file_path}")
# If the file is not found, try a different path or ensure the file exists
# For example, if you uploaded it using google.colab, access it like this:
file_path = list(uploaded.keys())[0] # Assuming 'uploaded' is defined as in your previous code
22°C Mostly clear
9:12 PM
10/27/2024

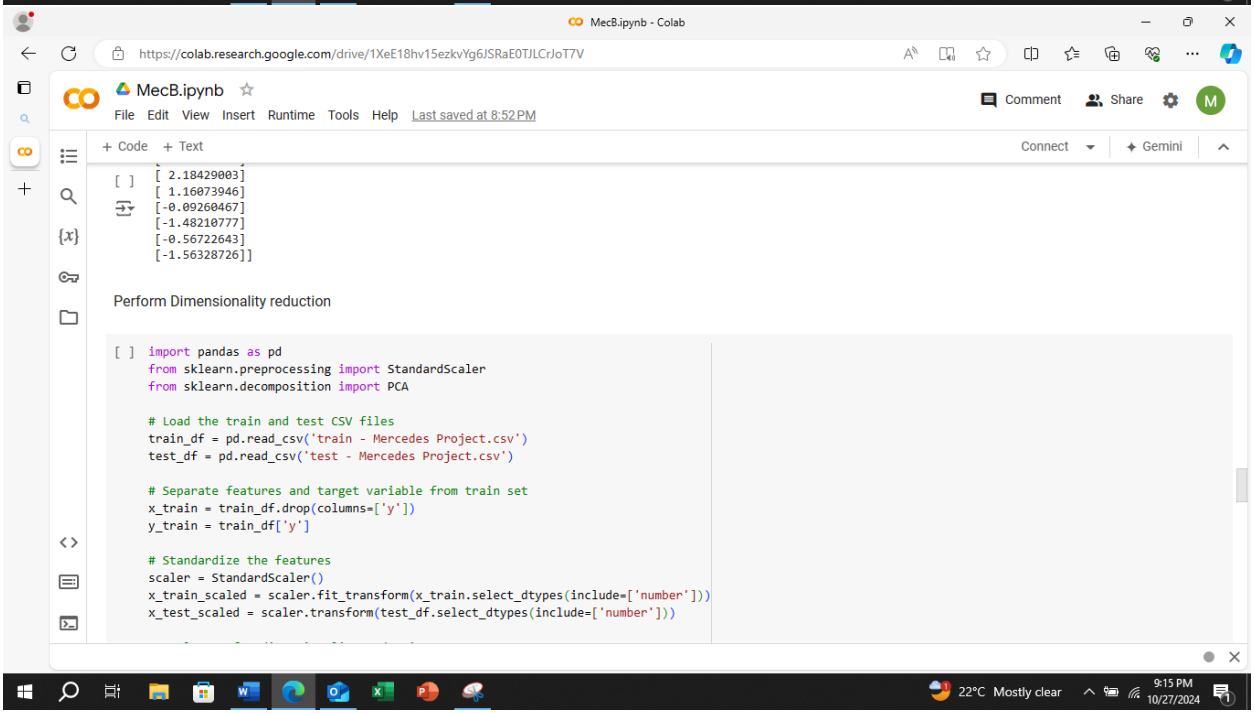

MecB.ipynb - Colab
https://colab.research.google.com/drive/1XeE18hv15ezkvYg6JSRaE0TJLCrJoT7V
MecB.ipynb
File Edit View Insert Runtime Tools Help Last saved at 8:52 PM
Comment
Share
+ Code + Text
Connect
Gemini
[ ] # Read the CSV file into a DataFrame
df = pd.read_csv(file_path)
# Display the first few rows of the DataFrame
print(df.head())
ID X0 X1 X2 X3 X4 X5 X6 X8 X10 ... X375 X376 X377 X378 X379 X380 \
0 1 az v n f d t a w 0 ... 0 0 0 1 0 0
1 2 t b ai a d b g y 0 ... 0 0 1 0 0 0
2 3 az v as f d a j j 0 ... 0 0 0 1 0 0
3 4 az l n f d z l n 0 ... 0 0 0 1 0 0
4 5 w s as c d y i m 0 ... 1 0 0 0 0 0
X382 X383 X384 X385
0 0 0 0 0
1 0 0 0 0
2 0 0 0 0
3 0 0 0 0
4 0 0 0 0
[5 rows x 377 columns]
[ ] Start coding or generate with AI.
[ ] import pandas as pd
Replace 'your_dataset.csv' with the actual name of your file
22°C Mostly clear
9:13 PM
10/27/2024

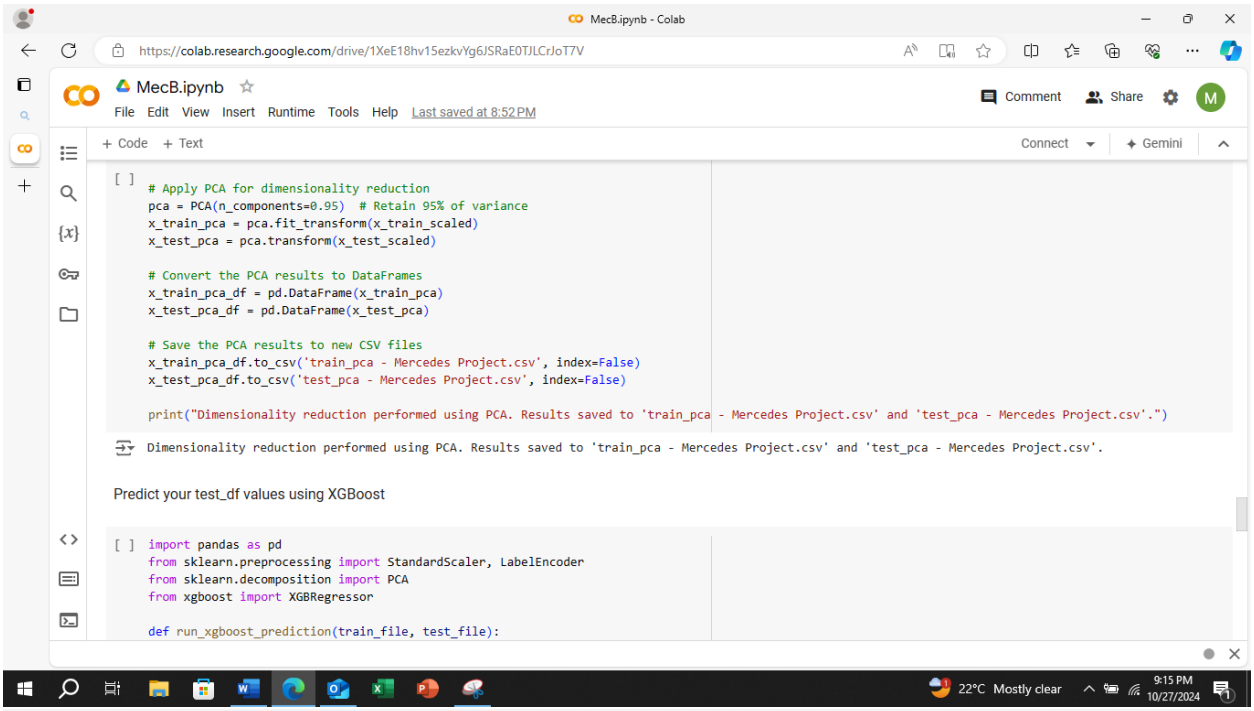

MecB.ipynb - Colab
https://colab.research.google.com/drive/1XeE18hv15ezkvYg6JSRaE0TJLCrJoT7V
MecB.ipynb
File Edit View Insert Runtime Tools Help Last saved at 8:52 PM
Comment
Share
+ Code + Text
Connect
Gemini
[ ] # If the file is in the same directory, this should work
file_path = 'train - Mercedes Project.csv'
# If the file is not in the same directory as your script,
# provide the full path, for example:
# file_path = '/path/to/your/file/your_dataset.csv'
# Check if the file exists at the specified path
import os
if not os.path.exists(file_path):
print(f"Error: File not found at {file_path}")
# If the file is not found, try a different path or ensure the file exists
# For example, if you uploaded it using google.colab, access it like this:
file_path = list(uploaded.keys())[0] # Assuming 'uploaded' is defined as in your previous code
# Read the CSV file into a DataFrame
df = pd.read_csv(file_path)
# Display the first few rows of the DataFrame
print(df.head())
ID y X0 X1 X2 X3 X4 X5 X6 X8 ... X375 X376 X377 X378 X379 \
0 0 130.81 k v at a d u j o ... 0 0 1 0 0
1 6 88.53 k t av e d y l o ... 1 0 0 0 0
2 7 76.26 az w n c d x j x ... 0 0 0 0 0
3 9 80.62 az t n f d x l e ... 0 0 0 0 0
4 13 78.02 az v n f d h d n ... 0 0 0 0 0
22°C Mostly clear
9:13 PM
10/27/2024

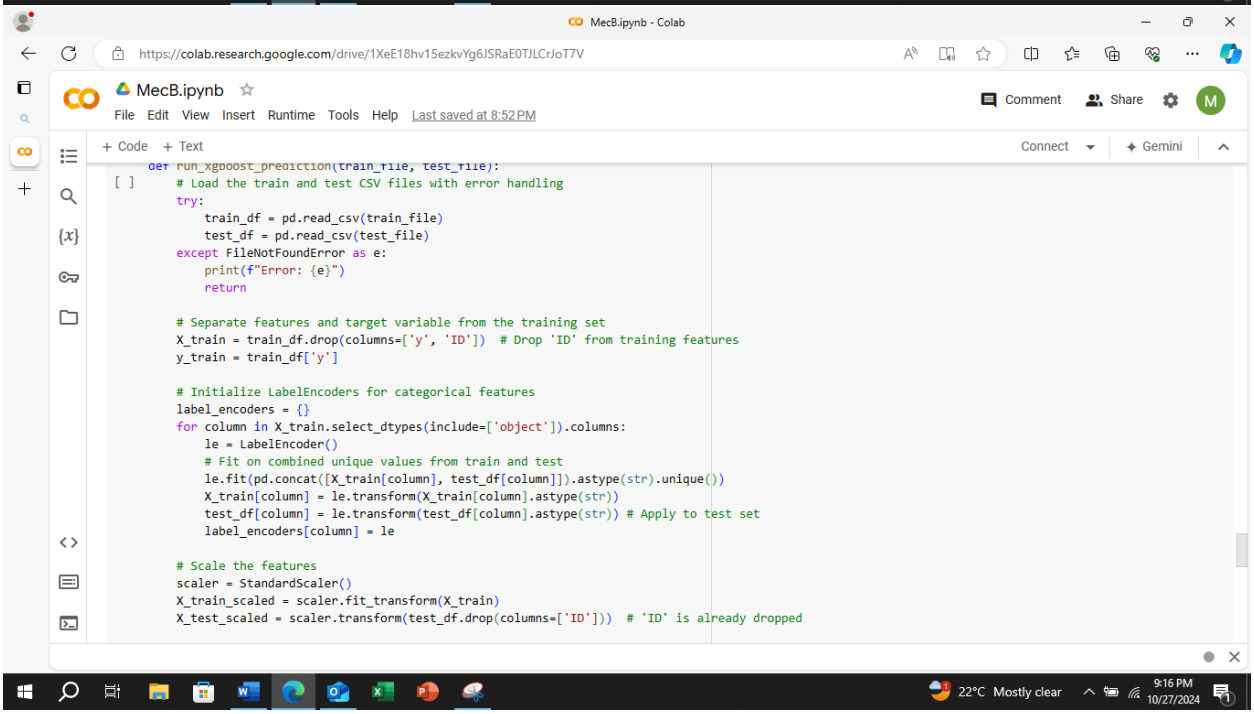

MecB.ipynb - Colab
https://colab.research.google.com/drive/1XeE18hv15ezkvYg6JSRaE0TJLCrJoT7V
MecB.ipynb
File Edit View Insert Runtime Tools Help Last saved at 8:52 PM
Comment
Share
M
+ Code + Text
Connect
Gemini
[ ]
X380 X382 X383 X384 X385
0 0 0 0 0 0
1 0 0 0 0 0
2 0 1 0 0 0
3 0 0 0 0 0
4 0 0 0 0 0
[5 rows x 378 columns]
[ ] import pandas as pd
from sklearn.preprocessing import LabelEncoder
# Assuming you have loaded your training data into a DataFrame named 'df'
# in a previous cell, rename it to 'train_df'
train_df = df # Replace 'df' with the actual variable name if different
# If you have a separate test dataset, load it into 'test_df' similarly
# test_df = pd.read_csv('train-Mercedes Project.csv')
# Apply LabelEncoder to categorical columns
label_encoder = {}
for column in train_df.columns:
if train_df[column].dtype == 'object':
le = LabelEncoder()
train_df[column] = le.fit_transform(train_df[column])
# test_df[column] = le.transform(test_df[column]) # Uncomment if using test_df
label_encoder[column] = le
22°C Mostly clear
9:14 PM
10/27/2024

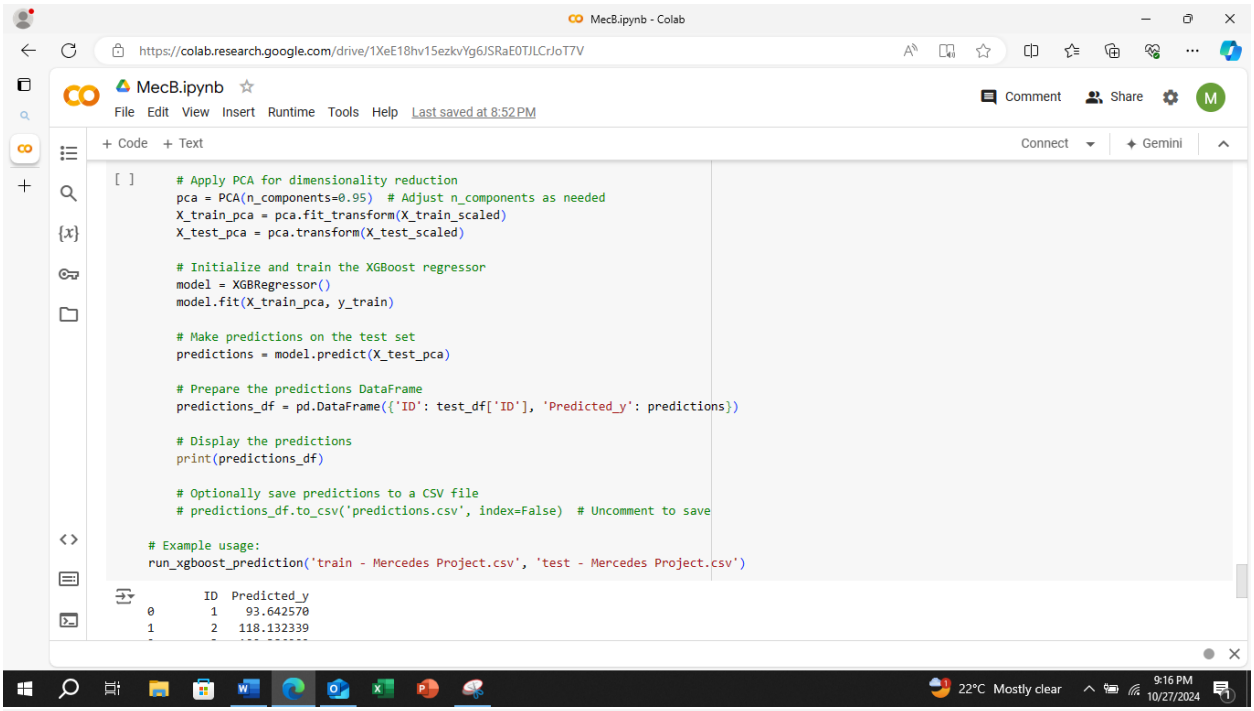

MecB.ipynb - Colab
https://colab.research.google.com/drive/1XeE18hv15ezkvYg6JSRaE0TJLCrJoT7V
MecB.ipynb
File Edit View Insert Runtime Tools Help Last saved at 8:52 PM
Comment
Share
M
+ Code + Text
Connect
Gemini
Apply Label Encoding
[ ] import pandas as pd
from sklearn.preprocessing import LabelEncoder
# Assuming you have loaded your training data into a DataFrame named 'df'
# in a previous cell, rename it to 'train_df'
train_df = df # Replace 'df' with the actual variable name if different
# If you have a separate test dataset, load it into 'test_df' similarly
# test_df = pd.read_csv('test-Mercedes Project.csv')
# Apply LabelEncoder to categorical columns
label_encoder = {}
for column in train_df.columns:
if train_df[column].dtype == 'object':
le = LabelEncoder()
train_df[column] = le.fit_transform(train_df[column])
# test_df[column] = le.transform(test_df[column]) # Uncomment if using test_df
label_encoder[column] = le
[ ] import numpy as np
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
# Sample data
22°C Mostly clear
9:14 PM
10/27/2024

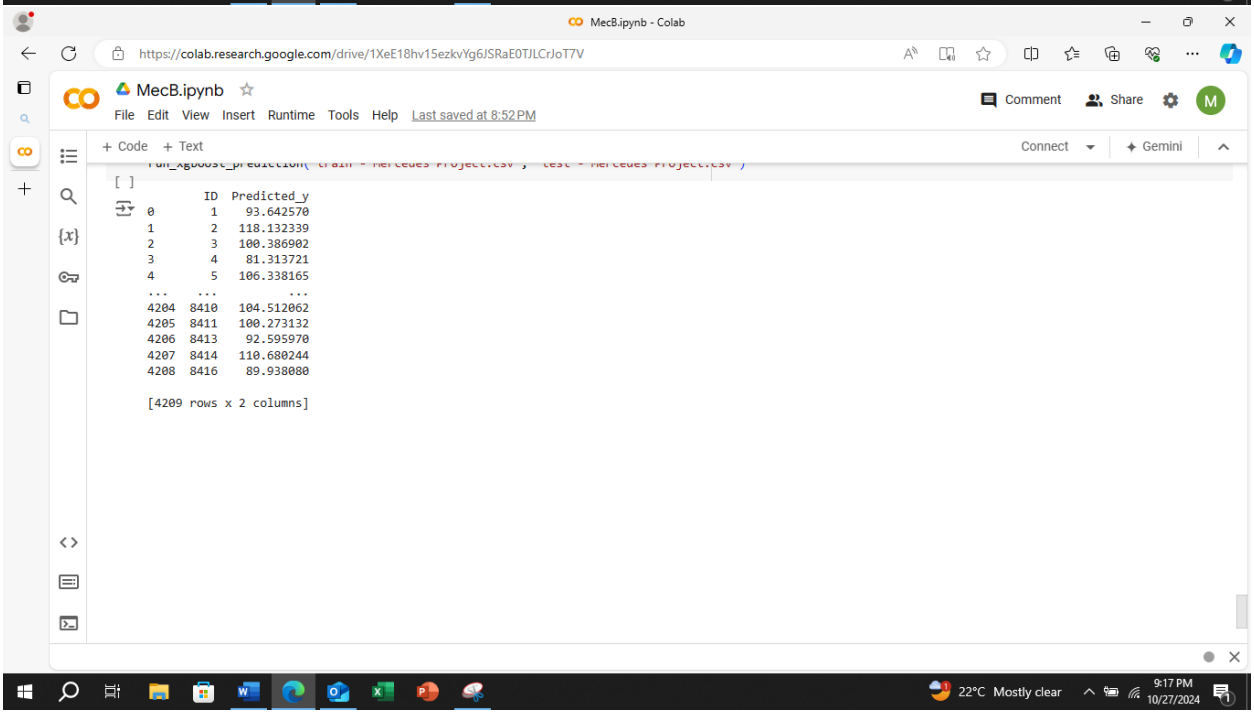

MecB.ipynb - Colab
https://colab.research.google.com/drive/1XeE18hv15ezkvYg6JSRaE0TJLCrJoT7V
MecB.ipynb
File Edit View Insert Runtime Tools Help Last saved at 8:52 PM
Comment
Share
+ Code + Text
Connect
Gemini
[ ]
# Sample data
data = np.array([[2.5, 2.4],
[0.5, 0.7],
[2.2, 2.9],
[1.9, 2.2],
[3.1, 3.0],
[2.3, 2.7],
[2, 1.6],
[1, 1.1],
[1.5, 1.6],
[1.1, 0.9]])
# Standardizing the data
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)
# Applying PCA
pca = PCA(n_components=1) # Reduce to 1 dimension
reduced_data = pca.fit_transform(data_scaled)
print("Reduced Data:\n", reduced_data)
Reduced Data:
[[1.08643242]
[-2.3089372]
[1.24191895]
[0.34078247]
[2.18429003]
22°C Mostly clear
9:15 PM
10/27/2024


MecB.ipynb - Colab
https://colab.research.google.com/drive/1XeE18hv15ezkvYg6JSRaE0TJLCrJoT7V
MecB.ipynb
File Edit View Insert Runtime Tools Help Last saved at 8:52 PM
Comment
Share
+ Code + Text
Connect
Gemini
[ ]
[2.18429003]
[1.16073946]
[-0.09260467]
[-1.48210777]
[-0.56722643]
[-1.56328726]]
Perform Dimensionality reduction
[ ]
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
# Load the train and test CSV files
train_df = pd.read_csv('train - Mercedes Project.csv')
test_df = pd.read_csv('test - Mercedes Project.csv')
# Separate features and target variable from train set
x_train = train_df.drop(columns=['y'])
y_train = train_df['y']
# Standardize the features
scaler = StandardScaler()
x_train_scaled = scaler.fit_transform(x_train.select_dtypes(include=['number']))
x_test_scaled = scaler.transform(test_df.select_dtypes(include=['number']))
22°C Mostly clear
9:15 PM
10/27/2024


MecB.ipynb - Colab
https://colab.research.google.com/drive/1XeE18hv15ezkvYg6JSRaE0TJLCrJoT7V
MecB.ipynb
File Edit View Insert Runtime Tools Help Last saved at 8:52 PM
Comment
Share
+ Code + Text
Connect
Gemini
[ ]
# Apply PCA for dimensionality reduction
pca = PCA(n_components=0.95) # Retain 95% of variance
x_train_pca = pca.fit_transform(x_train_scaled)
x_test_pca = pca.transform(x_test_scaled)
# Convert the PCA results to DataFrames
x_train_pca_df = pd.DataFrame(x_train_pca)
x_test_pca_df = pd.DataFrame(x_test_pca)
# Save the PCA results to new CSV files
x_train_pca_df.to_csv('train_pca - Mercedes Project.csv', index=False)
x_test_pca_df.to_csv('test_pca - Mercedes Project.csv', index=False)
print("Dimensionality reduction performed using PCA. Results saved to 'train_pca - Mercedes Project.csv' and 'test_pca - Mercedes Project.csv'.")
Dimensionality reduction performed using PCA. Results saved to 'train_pca - Mercedes Project.csv' and 'test_pca - Mercedes Project.csv'.
Predict your test_df values using XGBoost
[ ]
import pandas as pd
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.decomposition import PCA
from xgboost import XGBRegressor
def run_xgboost_prediction(train_file, test_file):
22°C Mostly clear
9:15 PM
10/27/2024


MecB.ipynb - Colab
https://colab.research.google.com/drive/1XeE18hv15ezkvYg6JSRaE0TJLCrJoT7V
MecB.ipynb
File Edit View Insert Runtime Tools Help Last saved at 8:52 PM
Comment
Share
+ Code + Text
Connect
Gemini
def run_xgboost_prediction(train_file, test_file):
[ ]
# Load the train and test CSV files with error handling
try:
train_df = pd.read_csv(train_file)
test_df = pd.read_csv(test_file)
except FileNotFoundError as e:
print(f"Error: {e}")
return
# Separate features and target variable from the training set
X_train = train_df.drop(columns=['y', 'ID']) # Drop 'ID' from training features
y_train = train_df['y']
# Initialize LabelEncoders for categorical features
label_encoders = {}
for column in X_train.select_dtypes(include=['object']).columns:
le = LabelEncoder()
# Fit on combined unique values from train and test
le.fit(pd.concat([X_train[column], test_df[column]]).astype(str).unique())
X_train[column] = le.transform(X_train[column].astype(str))
test_df[column] = le.transform(test_df[column].astype(str)) # Apply to test set
label_encoders[column] = le
# Scale the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(test_df.drop(columns=['ID'])) # 'ID' is already dropped
22°C Mostly clear
9:16 PM
10/27/2024


MecB.ipynb - Colab
https://colab.research.google.com/drive/1XeE18hv15ezkvYg6JSRaE0TJLCrJoT7V
MecB.ipynb
File Edit View Insert Runtime Tools Help Last saved at 8:52 PM
Comment
Share
+ Code + Text
Connect
Gemini
[ ]
# Apply PCA for dimensionality reduction
pca = PCA(n_components=0.95) # Adjust n_components as needed
X_train_pca = pca.fit_transform(X_train_scaled)
X_test_pca = pca.transform(X_test_scaled)
# Initialize and train the XGBoost regressor
model = XGBRegressor()
model.fit(X_train_pca, y_train)
# Make predictions on the test set
predictions = model.predict(X_test_pca)
# Prepare the predictions DataFrame
predictions_df = pd.DataFrame({'ID': test_df['ID'], 'Predicted_y': predictions})
# Display the predictions
print(predictions_df)
# Optionally save predictions to a CSV file
# predictions_df.to_csv('predictions.csv', index=False) # Uncomment to save
# Example usage:
run_xgboost_prediction('train - Mercedes Project.csv', 'test - Mercedes Project.csv')
ID Predicted_y
0 1 93.642570
1 2 118.132339
22°C Mostly clear
9:16 PM
10/27/2024


MecB.ipynb - Colab
https://colab.research.google.com/drive/1XeE18hv15ezkvYg6JSRaE0TJLCrJoT7V
MecB.ipynb
File Edit View Insert Runtime Tools Help Last saved at 8:52 PM
Comment
Share
+ Code + Text
Connect
Gemini
[ ]
ID Predicted_y
0 1 93.642570
1 2 118.132339
2 3 100.386902
3 4 81.313721
4 5 106.338165
... ...
4204 8410 104.512062
4205 8411 100.273132
4206 8413 92.595970
4207 8414 110.680244
4208 8416 89.938080
[4209 rows x 2 columns]
22°C Mostly clear
9:17 PM
10/27/2024