import ast

*# Load the dataset*

X\_train = pd.read\_csv('your\_file.csv')

*# Define a function to count the number of languages*

def count\_languages(language\_list\_str):

*# Convert the string representation of the list to an actual list*

language\_list = ast.literal\_eval(language\_list\_str)

*# Return the length of the list*

return len(language\_list)

*# Apply the function to the column, assuming the column name is 'languages'*

X\_train['Supported languages'] = X\_train['Supported languages'].apply(count\_languages)

In [ ]:

def count\_items(cell\_content):

*# Split the string by commas and count the resulting items*

return len(cell\_content.split(','))

*# Apply the function to the column, assuming the column name is 'features'*

X\_train['Categories'] = X\_train['Categories'].apply(count\_items)

In [ ]:

*# Split the genres column into separate tags and apply one-hot encoding*

genres\_split = X\_train['Genres'].str.get\_dummies(sep=',')

*# Concatenate the original DataFrame with the new one-hot encoded DataFrame*

X\_train = pd.concat([X\_train, genres\_split], axis=1)

*# Optionally, drop the original 'genres' column if it's no longer needed*

X\_train.drop('Genres', axis=1, inplace=True)

In [ ]:

*#### VARIANTA 1*

unique\_values = X\_train['Estimated Owners'].unique()

*# Print the unique values*

print("Unique values in 'Estimated Owners' column:")

print(unique\_values)

*# Define the order of ranges for ordinal encoding (based on the unique values found)*

range\_order = sorted(unique\_values, key=lambda x: int(x.split('-')[0]))

*# Create a mapping from range to ordinal value*

range\_mapping = {range\_value: idx for idx, range\_value **in** enumerate(range\_order)}

*# Display the mapping*

print("**\n**Range to Ordinal Mapping:")

print(range\_mapping)

*# Apply the mapping to the 'estimated\_peak' column*

X\_train['estimated\_peak\_encoded'] = X\_train['estimated\_peak'].map(range\_mapping)

*# Display the transformed DataFrame*

print("**\n**Transformed DataFrame:")

print(X\_train)

*##### VARIANTA 2 kindaaa (vezi care merge ca idk)*

range\_order = ["0-2000", "2000-5000", "5000-10000"]

*# Create a mapping from range to ordinal value*

range\_mapping = {range\_value: idx for idx, range\_value **in** enumerate(range\_order)}

*# Display the mapping*

print("**\n**Range to Ordinal Mapping:")

print(range\_mapping)

*# Apply the mapping to the 'estimated\_peak' column*

X\_train['estimated\_peak\_encoded'] = X\_train['estimated\_peak'].map(range\_mapping)

*# Display the transformed DataFrame*

print("**\n**Transformed DataFrame:")

print(X\_train)