**Python code for the automatic data types convert to needed types**

**import unicodecsv**

**import pandas as pd**

**import numpy as np**

**enrollments\_filename = '/datasets/ud170/udacity-students/enrollments.csv'**

**[[**## Longer version of code (replaced with shorter, equivalent version below)

# enrollments = []

# f = open(enrollments\_filename, 'rb')

# reader = unicodecsv.DictReader(f)

# for row in reader:

# enrollments.append(row)

# f.close()**]]**

**with open(enrollments\_filename, 'rb') as f:**

**reader = unicodecsv.DictReader(f)**

**enrollments = list(reader)**

**#print first\_line in the csv**

**enrollments[0]**

**############# Convert the list of dictionaries to a Pandas DataFrame**

**df = pd.DataFrame(enrollments)**

**################### Display the data types of each column**

**print(df.dtypes)**

**###############code for do the auto convert at a time repected data types**

* **Date data convert DATE datatype**
* **Numeric convert to INT datatype**
* **Float convert to FLOAT datatype etc……**

############################## Convert data types for each column ##################

**df['account\_key'] = df['account\_key'].astype(int)**

**df['join\_date'] = pd.to\_datetime(df['join\_date'])**

**# Handles empty strings as NaT**

**df['cancel\_date'] = pd.to\_datetime(df['cancel\_date'], errors='coerce')**

**# Handles 'nan' as NaN**

**df['days\_to\_cancel'] = pd.to\_numeric(df['days\_to\_cancel'], errors='coerce')**

**df['is\_udacity'] = df['is\_udacity'].astype(bool) df['is\_canceled'] = df['is\_canceled'].astype(bool)**

**print(df.dtypes)**

**#####help of the function do the data type conversions**

**import unicodecsv**

**import pandas as pd**

**from datetime import datetime as dt**

**# File path**

**enrollments\_filename = '/datasets/ud170/udacity-students/enrollments.csv'**

**# Reading the CSV file using unicodecsv**

**with open(enrollments\_filename, 'rb') as f:**

**reader = unicodecsv.DictReader(f)**

**enrollments = list(reader)**

**# Convert the list of dictionaries to a Pandas DataFrame**

**df = pd.DataFrame(enrollments)**

**# Display the data types of each column before conversion**

**print("Before conversion:")**

**print(df.dtypes)**

**# Function to parse dates**

**def parse\_date(date):**

**if date == '' or pd.isna(date):**

**return None**

**else:**

**return dt.strptime(date, "%Y-%m-%d")**

**# Function to parse maybe int**

**def parse\_maybe\_int(i):**

**if i == '' or pd.isna(i):**

**return None**

**else:**

**return int(i)**

**# Function to parse booleans**

**def parse\_boolean(value):**

**return value.lower() == 'true'**

**# Applying the parsing functions to the DataFrame**

**df['account\_key'] = df['account\_key'].astype(int) # String to int**

**df['join\_date'] = df['join\_date'].apply(parse\_date) # String to date**

**df['cancel\_date'] = df['cancel\_date'].apply(parse\_date) # String to date**

**df['days\_to\_cancel'] = df['days\_to\_cancel'].apply(parse\_maybe\_int) # String to int or None**

**df['is\_udacity'] = df['is\_udacity'].apply(parse\_boolean) # String to boolean**

**df['is\_canceled'] = df['is\_canceled'].apply(parse\_boolean) # String to boolean**

**# Display the data types of each column after conversion**

**print("\nAfter conversion:")**

**print(df.dtypes)**

**####data type conversions and as well as use to inr conversions as well**

**import unicodecsv**

**import pandas as pd**

**from datetime import datetime as dt**

**# File path**

**enrollments\_filename = '/datasets/ud170/udacity-students/enrollments.csv'**

**# Reading the CSV file using unicodecsv**

**with open(enrollments\_filename, 'rb') as f:**

**reader = unicodecsv.DictReader(f)**

**enrollments = list(reader)**

**# Convert the list of dictionaries to a Pandas DataFrame**

**df = pd.DataFrame(enrollments)**

**# Display the data types of each column before conversion**

**print("Before conversion:")**

**print(df.dtypes)**

**# Function to parse dates**

**def parse\_date(date):**

**if date == '' or pd.isna(date):**

**return None**

**else:**

**return dt.strptime(date, "%Y-%m-%d")**

**# Function to parse maybe int**

**def parse\_maybe\_int(i):**

**if i == '' or pd.isna(i):**

**return None**

**else:**

**return int(i)**

**# Function to parse booleans**

**def parse\_boolean(value):**

**return value.lower() == 'true'**

**# Function to convert USD to INR**

**def usd\_to\_inr(amount, conversion\_rate=82.5): # Assuming a conversion rate of 82.5 INR per 1 USD**

**if amount == '' or pd.isna(amount):**

**return None**

**else:**

**return float(amount) \* conversion\_rate**

**# Applying the parsing functions to the DataFrame**

**df['account\_key'] = df['account\_key'].astype(int) # String to int**

**df['join\_date'] = df['join\_date'].apply(parse\_date) # String to date**

**df['cancel\_date'] = df['cancel\_date'].apply(parse\_date) # String to date**

**df['days\_to\_cancel'] = df['days\_to\_cancel'].apply(parse\_maybe\_int) # String to int or None**

**df['is\_udacity'] = df['is\_udacity'].apply(parse\_boolean) # String to boolean**

**df['is\_canceled'] = df['is\_canceled'].apply(parse\_boolean) # String to boolean**

**# Assuming there's a 'tuition\_fee\_usd' column for conversion**

**df['tuition\_fee\_usd'] = df['tuition\_fee\_usd'].apply(lambda x: usd\_to\_inr(x)) # Convert USD to INR**

**# Display the data types of each column after conversion**

**print("\nAfter conversion:")**

**print(df.dtypes)**

**######################help of for loop to do the one time data type conversions**

**import unicodecsv**

**import pandas as pd**

**from datetime import datetime as dt**

**# File path**

**enrollments\_filename = '/datasets/ud170/udacity-students/enrollments.csv'**

**# Reading the CSV file using unicodecsv**

**with open(enrollments\_filename, 'rb') as f:**

**reader = unicodecsv.DictReader(f)**

**enrollments = list(reader)**

**# Convert the list of dictionaries to a Pandas DataFrame**

**df = pd.DataFrame(enrollments)**

**# Display the data types of each column before conversion**

**print("Before conversion:")**

**print(df.dtypes)**

**# Function to parse dates**

**def parse\_date(date):**

**if date == '' or pd.isna(date):**

**return None**

**else:**

**return dt.strptime(date, "%Y-%m-%d")**

**# Function to parse maybe int**

**def parse\_maybe\_int(i):**

**if i == '' or pd.isna(i):**

**return None**

**else:**

**return int(i)**

**# Function to parse booleans**

**def parse\_boolean(value):**

**return value.lower() == 'true'**

**# Function to convert USD to INR**

**def usd\_to\_inr(amount, conversion\_rate=82.5): # Assuming a conversion rate of 82.5 INR per 1 USD**

**if amount == '' or pd.isna(amount):**

**return None**

**else:**

**return float(amount) \* conversion\_rate**

**# Mapping columns to their respective conversion functions**

**conversion\_functions = {**

**'account\_key': int,**

**'join\_date': parse\_date,**

**'cancel\_date': parse\_date,**

**'days\_to\_cancel': parse\_maybe\_int,**

**'is\_udacity': parse\_boolean,**

**'is\_canceled': parse\_boolean,**

**'tuition\_fee\_usd': usd\_to\_inr**

**}**

**# Applying the conversion functions using a for loop**

**for column, func in conversion\_functions.items():**

**df[column] = df[column].apply(func)**

**# Display the data types of each column after conversion**

**print("\nAfter conversion:")**

**print(df.dtypes)**