TP2- Regression (1)

Parcours Progis

Etudes, Medias, communication, Marketing

06.10.2025

Bahareh Afshinpour

Université Grenoble Alpes

Sciences Po Grenoble UGA

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Customer Lifetime Value (CLV)

- **CLV** is the **total amount of money** a customer is expected to spend on your business **over their entire relationship** with you.

- **Why is CLV Important?**
  - **Marketing:** Helps decide how much to spend to acquire a new customer.
  - **Social Media:** Helps identify which customers to target with ads or promotions.
  - **Business Strategy:** Helps focus on high-value customers and improve their experience.

If a customer buys a $50 product every month and stays with your brand for 2 years, their CLV is about $1,200.

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Why Predict CLV with Regression?

- **Regression** is a way to predict a number (like CLV) based on other numbers (like how often a customer buys, how much they spend, etc.).
- If you can predict CLV, you can:
  - ❖Spend more to acquire high-CLV customers.
  - ❖Offer special deals to keep them.
  - ❖Focus your social media ads on similar people.

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

For a marketing/social media dataset, you might have columns like:

Column Name	What It Means	Example Value
customer_id	Unique ID for each customer	1001
avg_purchase_value	Average amount spent per purchase	$50
purchase_frequency	How often they buy (e.g., per month)	2
campaign_response_rate	How often they respond to marketing campaigns	0.75
total_spend	Total amount spent so far	$600
Customer_Lifetime_Value	**Target:** Total predicted value over time	$1,200

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

How Does Regression Work Here?

- **Input (X):** Columns like avg_purchase_value, purchase_frequency, campaign_response_rate
- **Output (y):** Customer_Lifetime_Value
- **Model:** Finds the relationship between X and y, so you can predict y for new customers.

If a new customer has:
avg_purchase_value = $60, purchase_frequency = 1.5
campaign_response_rate = 0.6
The model can predict their **CLV** (e.g., $900)

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Data Frames attributes

Python objects have *attributes* and *methods*.

df.attribute	description
dtypes	list the types of the columns
columns	list the column names
axes	list the row labels and column names
ndim	number of dimensions
size	number of elements
shape	return a tuple representing the dimensionality
values	numpy representation of the data

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Data Frames methods

Unlike attributes, python methods have *parenthesis.*
All attributes and methods can be listed with a *dir()* function: `dir(df)`

df.method()	description
head([n]), tail([n])	first/last n rows
describe()	generate descriptive statistics (for numeric columns only)
max(), min()	return max/min values for all numeric columns
mean(), median()	return mean/median values for all numeric columns
std()	standard deviation
sample([n])	returns a random sample of the data frame
dropna()	drop all the records with missing values

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Exploring data frames

You can have a look at the first five rows of your dataframe with **.head()**:

```
#List first 5 records
df.head()
```

	rank	discipline	phd	service	sex	salary
0	Prof	B	56	49	Male	186960
1	Prof	A	12	6	Male	93000
2	Prof	A	23	20	Male	110515
3	Prof	A	40	31	Male	131205
4	Prof	B	20	18	Male	104800

You also use the **.shape** attribute of the DataFrame to see its **dimensionality**. The result is a tuple containing the number of rows and columns.

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

LOADING PYTHON LIBRARIES

In [ ]:
```
#Import Python Libraries
import numpy as np
import scipy as sp
import pandas as pd
import matplotlib as mpl
import seaborn as sns
```

Press `Shift+Enter` to execute the *jupyter* cell

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

READING DATA USING PANDAS

```
#Read csv file
df = pd.read_csv(”filename.csv")
```

Note: The above command has many optional arguments to fine-tune the data import process.

There is a number of pandas commands to read other data formats:

```
pd.read_excel('myfile.xlsx',sheet_name='Sheet1', index_col=None, na_values=['NA'])
pd.read_stata('myfile.dta')
pd.read_sas('myfile.sas7bdat')
pd.read_hdf('myfile.h5','df')
```

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv('./WA_Fn-UseC_-Marketing-Customer-Value-Analysis.csv', encoding='latin1')
data.columns
```

[1]:
```
Index(['Customer', 'State', 'Customer Lifetime Value', 'Response', 'Coverage',
       'Education', 'Effective To Date', 'EmploymentStatus', 'Gender',
       'Income', 'Location Code', 'Marital Status', 'Monthly Premium Auto',
       'Months Since Last Claim', 'Months Since Policy Inception',
       'Number of Open Complaints', 'Number of Policies', 'Policy Type',
       'Policy', 'Renew Offer Type', 'Sales Channel', 'Total Claim Amount',
       'Vehicle Class', 'Vehicle Size'],
      dtype='object')
```

[3]:
```
data.head()
```

[3]:

	Customer	State	Customer Lifetime Value	Response	Coverage	Education	Effective To Date	EmploymentStatus	Gender	Income	...
0	BU79786	Washington	2763.519279	No	Basic	Bachelor	2/24/11	Employed	F	56274	...
1	QZ44356	Arizona	6979.535903	No	Extended	Bachelor	1/31/11	Unemployed	F	0	...

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Selecting a column in a Data Frame

Method 1: Subset the data frame using column name:

```
df['sex']
```

Method 2: Use the column name as an attribute:

```
df.sex
```

Note: there is an attribute *rank* for pandas data frames, so to select a column with a name "rank" we should use method 1.

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Data Frame data types

```
#Check a particular column type
df['salary'].dtype
```

```
#Check types for all the columns
df.dtypes
```

```
rank          object
discipline    object
phd           int64
service       int64
sex           object
salary        int64
dtype: object
```

```
data.dtypes
```

```
Customer                      object
State                         object
Customer Lifetime Value      float64
Response                      object
Coverage                      object
Education                     object
Effective To Date             object
EmploymentStatus              object
Gender                        object
Income                         int64
Location Code                 object
Marital Status                object
```

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

DataFrame.map()

Used for substituting each value in a Series with another value

```
# Nunmerical Mapping of response in dataset
response_mapping = {
    'No' : 0,
    'Yes' : 1
}

data['Response'] = data['Response'].map(response_mapping)
data
```

	Customer	State	Customer Lifetime Value	Response	Coverage	Education	Effective To Date	Empl
0	BU79786	Washington	2763.519279	0	Basic	Bachelor	2/24/11	
1	QZ44356	Arizona	6979.535903	0	Extended	Bachelor	1/31/11	
2	AI49188	Nevada	12887.431650	0	Premium	Bachelor	2/19/11	

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Checking for the Missing values

Missing values can be imputed with a provided constant value, or using the statistics (mean, median or most frequent) of each column in which the missing values are located.

Numerical features

A	B
22	34
36	78
15	66
	78

Replace with mean

A	B
22	34
36	78
15	66
24,3	78
24,3	64

Categorical features

C	D
A	P
A	P
A	Q
B	R
	p

Replace with mode

C	D
A	P
A	P
A	Q
B	R
A	P

isnull(), Detect missing values for an array-like object.

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Checking for the Missing values

- Mean simply returns the sum of the values divided by the number of values.

```
[29]: data.isnull().mean()
```

```
[29]: Customer                  0.0
      State                     0.0
      Customer Lifetime Value   0.0
      Response                  0.0
      Coverage                  0.0
      Education                 0.0
      Effective To Date         0.0
      EmploymentStatus          0.0
      Gender                    0.0
      Income                    0.0
```

data.isnull().mean()

This data does not have any missing data.

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Select_dtypes (1)

- Here, the goal is to split the data into two parts, **numeric** and **categorical**.
- continous_var_data = data.select_dtypes(include=['int64','float'])

```
num=data.select_dtypes(include='number')
num.head()
```

	Customer Lifetime Value	Income	Monthly Premium Auto	Months Since Last Claim
0	2763.519279	56274	69	32
1	6979.535903	0	94	13
2	12887.431650	48767	108	18
3	7645.861827	0	106	18
4	2813.692575	43836	73	12

```
cate=data.select_dtypes(include='object')
cate.head()
```

	Customer	State	Response	Coverage	Education	Effective To Date	Empl
0	BU79786	Washington	No	Basic	Bachelor	2/24/11	
1	QZ44356	Arizona	No	Extended	Bachelor	1/31/11	

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Select_dtypes (2)

num=data.select_dtypes(include='number')

num.head()

```
num=data.select_dtypes(include='number')
num.head()
```

	Customer Lifetime Value	Income	Monthly Premium Auto	Months Since Last Claim	Months Since Policy Inception
0	2763.519279	56274	69	32	5
1	6979.535903	0	94	13	42
2	12887.431650	48767	108	18	38
3	7645.861827	0	106	18	65
4	2813.692575	43836	73	12	44

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

DataFrame drop() Method

- Suppose one column x has many missing values, so we want to drop it.
- we can drop a single column from a DataFrame using the **.drop()** .

```
[21]: data.shape

[21]: (9134, 24)

[25]: data_drop=data.drop('Location Code', axis=1)
      data_drop.shape

[25]: (9134, 23)
```

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Defining Independent and Target features

How to take our data frame and split it into two important parts.

```
y = continous_var_data['Customer Lifetime Value']
X = continous_var_data.drop(columns=['Customer Lifetime Value'])
```

End