Universidad Nacional Autónoma de México

Diplomado en Ciencia de Datos

Módulo I:

Introducción, Manipulación, Exploración y Visualización de Datos.

Profesora:
Act. Carla Paola Malerva Reséndiz

Facultad de Estudios Superiores de Acatlán

Presenta:
Francisco Roman Peña de la Rosa

Reporte Examen II

Ciudad de México, 2021

2.- Calidad de Datos

En el apartado de '**revisar y eliminar duplicados, manteniendo el primer elemento**', se dividió el dataframe inicial, en 2: **df1** y **df2** con el objetivo de testear el método de "**drop.duplicates**" en **df1** y en **df2** aplicar el método de eliminación de duplicados por llave con las columnas (**v_id_client** y **v_id_shop**):

A) En el primer enfoque de eliminación de registros duplicados, se encontraron 5 duplicados, pasando de un dataframe de dimensión: (51000, 32) => (50995, 32).

B) Por otra parte, en el método de eliminación utilizando columnas **id** antes citadas, se identificaron y eliminaron 1,000 registros. La razón por la que también se consideró trabajara con este segundo método, se debe a que se dan y existen casos en donde la persona que solicita un crédito lo realiza por más de una ocasión dentro de la misma tienda/sucursal, pues la misma institución bancaria o encargada de otorgar los créditos brinda esa opción y permite al cliente solicitar un nuevo préstamo dentro de un determinado periodo de tiempo o intentando con nueva documentación, avales, comprobantes de ingresos por mencionar algunos.

Para continuar con el análisis de nuestros datos, se decidió continuar con el enfoque de eliminación de duplicados por id (aplicado a df2).

- **Etiquetado de Variables**

Variables de Tipo Numérico:

c_age
c_quant_dependants
c_months_in_residence
c_months_in_the_job
c_mate_income
c_payment_day
c_shop_rank
c_quant_banking_accounts
c_personal_net_income
c_quant_additional_cards_in_the_application

Variables de Tipo Categórico

v_id_client'
v_id_shop
v_sex
v_marital_status
v_education
v_flag_residencial_phone
v_area_code_residencial_phone
v_residence_type
v_flag_mothers_name
v_flag_fathers_name
v_flag_residence_town=working_town
v_flag_residence_state=working_state
v_flag_residencial_address=postal_address
v_profession_code
v_flag_other_card

v_flag_mobile_phone
v_flag_contact_phone
v_cod_application_booth
v_flag_card_insurance_option
v_tgt

Variables de Tipo Texto

t_personal_reference_#1
t_personal_reference_#2

- **Completitud**

En esta sección se muestra una comparación entre ambos dataframes utilizados para eliminar duplicados, a través del método drop.duplicates (**df1**) y el dataframe en donde se eliminan duplicados a través de id (**df2**):

Para **df1** tenemos:

	columna	total	completitud
0	v_education	50995	0.000000
1	t_personal_reference_#1	20625	59.554858
2	t_personal_reference_#2	13886	72.769879
3	c_age	2472	95.152466
4	v_flag_contact_phone	1279	97.491911
5	c_mate_income	306	99.399941
6	v_sex	3	99.994117
7	v_id_client	0	100.000000
8	v_flag_residencial_address=postal_address	0	100.000000
9	v_flag_other_card	0	100.000000
10	c_quant_banking_accounts	0	100.000000
11	v_flag_mobile_phone	0	100.000000
12	c_personal_net_income	0	100.000000
13	v_cod_application_booth	0	100.000000
14	c_quant_additional_cards_in_the_application	0	100.000000
15	v_profession_code	0	100.000000
16	c_months_in_the_job	0	100.000000
17	v_flag_residence_town=working_town	0	100.000000
18	v_flag_card_insurance_option	0	100.000000
19	v_flag_fathers_name	0	100.000000
20	v_flag_mothers_name	0	100.000000
21	c_months_in_residence	0	100.000000
22	v_residence_type	0	100.000000
23	v_shop_rank	0	100.000000
24	v_payment_day	0	100.000000
25	v_area_code_residencial_phone	0	100.000000
26	v_flag_residencial_phone	0	100.000000
27	c_quant_dependants	0	100.000000
28	v_marital_status	0	100.000000
29	v_id_shop	0	100.000000
30	v_flag_residence_state=working_state	0	100.000000
31	v_tgt	0	100.000000

Para **df2** tenemos:

	columna	total	completitud
0	v_education	50000	0.000000
1	t_personal_reference_#1	20624	58.752000
2	t_personal_reference_#2	13792	72.416000
3	c_age	2471	95.058000
4	v_flag_contact_phone	1279	97.442000
5	c_mate_income	306	99.388000
6	v_sex	3	99.994000
7	v_id_client	0	100.000000
8	v_flag_residencial_address=postal_address	0	100.000000
9	v_flag_other_card	0	100.000000
10	c_quant_banking_accounts	0	100.000000
11	v_flag_mobile_phone	0	100.000000
12	c_personal_net_income	0	100.000000
13	v_cod_application_booth	0	100.000000
14	c_quant_additional_cards_in_the_application	0	100.000000
15	v_profession_code	0	100.000000
16	c_months_in_the_job	0	100.000000
17	v_flag_residence_town=working_town	0	100.000000
18	v_flag_card_insurance_option	0	100.000000
19	v_flag_fathers_name	0	100.000000
20	v_flag_mothers_name	0	100.000000
21	c_months_in_residence	0	100.000000
22	v_residence_type	0	100.000000
23	v_shop_rank	0	100.000000
24	v_payment_day	0	100.000000
25	v_area_code_residencial_phone	0	100.000000
26	v_flag_residencial_phone	0	100.000000
27	c_quant_dependants	0	100.000000
28	v_marital_status	0	100.000000
29	v_id_shop	0	100.000000
30	v_flag_residence_state=working_state	0	100.000000
31	v_tgt	0	100.000000

Como podemos ver, si tenemos una diferencia al comparar ambas tablas resultantes: el número de registros faltantes para las primeras 4 variables: v_education, t_personal_reference_#1, t_personal_reference_#2 y c_age disminuye en df2.

- **Consistencia y Conformidad**

Se aplicó el formato correspondiente a cada variable: integer, float o string de acuerdo su tipo. Ejemplo de algunos de los cambios aplicados son:

- **v_marital_status**: se homologó a las 5 categorías correspondientes, pues dentro de los registros se tenían espacios adicionales al inicio o final de cada categoría, lo que causaba que se contabilizara y visualizará como una diferente siendo la misma. Por ejemplo: **array (['S', 'C', 'O', 'V', 'D', 'S ', 'C ', 'D ', 'V ', 'O '] =** > array (['S', 'C', 'O', 'V', 'D'].
- **v_residence_type**: presento el mismo comportamiento que v_marital_status, y a través de las tranformaciones necesarias, la variable quedó definida como se presenta: **array (['P', ' ', 'A', 'C', 'O', ' a', 'p', ' p', 'P ', 'a', ' c', 'C ', ' o', 'O ', 'A ', 'c', 'o', 'p '] =>** array (['P', 'NaN', 'C', 'O'].
- c_mate_income: se realizaron conversiones de flotantes -> enteros, y los valores de tipo: nan o n.a. fueron reemplazados por NaN, utilizando np.nan para ser imputados posteriormente.

- **Completitud**

Después de llevar acabo el primer análisis de completitud, se aplican las trasnformaciones mencionadas en la sección anterior y obtenemos como resultado la siguiente tabla:

	columna	total	completitud
0	v_education	50000	0.000000
1	v_id_client	0	100.000000
2	c_quant_additional_cards_in_the_application	0	100.000000
3	v_cod_application_booth	0	100.000000
4	c_personal_net_income	0	100.000000
5	v_flag_contact_phone	0	100.000000
6	v_flag_mobile_phone	0	100.000000
7	t_personal_reference_#2	0	100.000000
8	t_personal_reference_#1	0	100.000000
9	c_quant_banking_accounts	0	100.000000
10	v_flag_other_card	0	100.000000
11	v_flag_residencial_address=postal_address	0	100.000000
12	c_mate_income	0	100.000000
13	v_profession_code	0	100.000000
14	c_months_in_the_job	0	100.000000
15	v_flag_residence_state=working_state	0	100.000000
16	v_flag_residence_town=working_town	0	100.000000
17	v_flag_fathers_name	0	100.000000
18	v_flag_mothers_name	0	100.000000
19	c_months_in_residence	0	100.000000
20	v_residence_type	0	100.000000
21	v_shop_rank	0	100.000000
22	v_payment_day	0	100.000000
23	v_area_code_residencial_phone	0	100.000000
24	v_flag_residencial_phone	0	100.000000
25	c_quant_dependants	0	100.000000
26	c_age	0	100.000000
27	v_marital_status	0	100.000000
28	v_sex	0	100.000000
29	v_id_shop	0	100.000000
30	v_flag_card_insurance_option	0	100.000000
31	v_tgt	0	100.000000

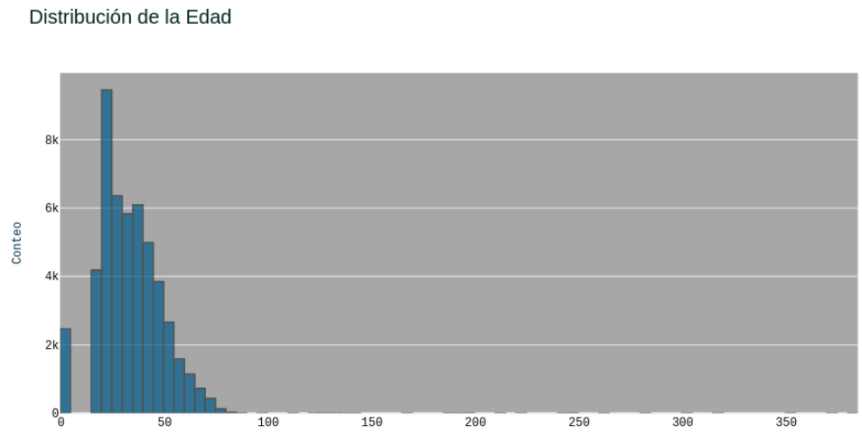
- **Eliminación de Variables – Completitud < 80%**

De la tabla anterior, vemos que la variable que no cumple con este criterio es "**v_education**" y procedemos a elminarla.

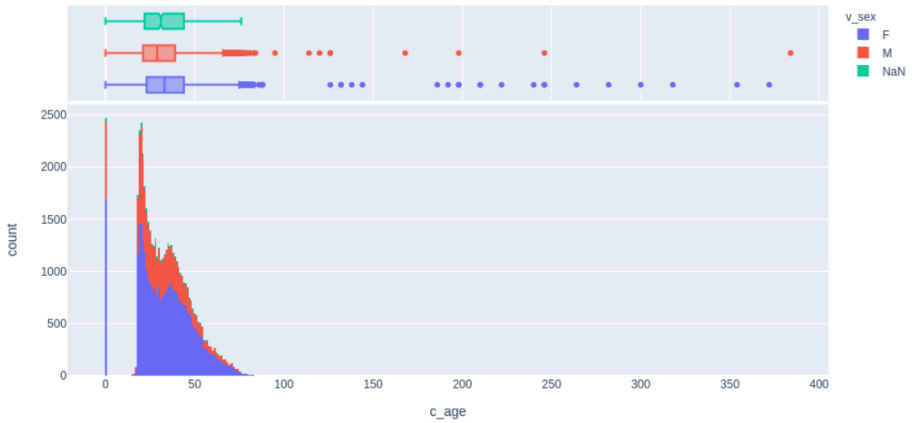
- **Análisis Exploratorio de Datos (EDA)**

Como objeto de nuestro análisis, se realizaron algunas gráfcas que permitieran visualizar el comportamiento de algunas variables, y para algunos casos su relación con la variable objetivo. Dichos gráficos se muestran a continuación:

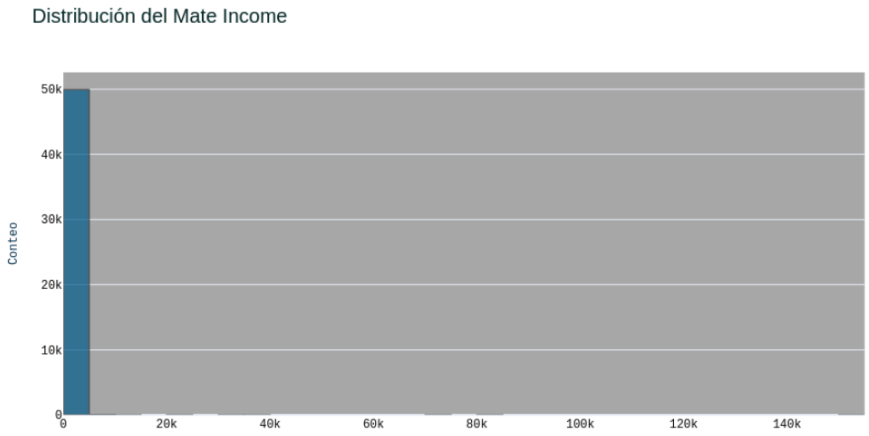
- **c_age:** En el gráfico siguiente se muestra la distribución de la edad:



Distribución de la Edad VS el Sexo



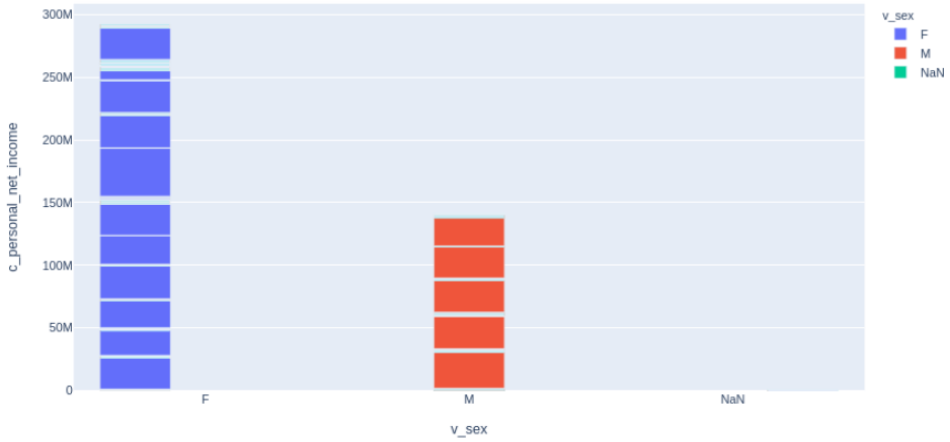
- **c_mate_income:** En los gráficos siguientes se muestra el comportamiento de la distribución de ésta y algunas relaciones con respecto de otras.



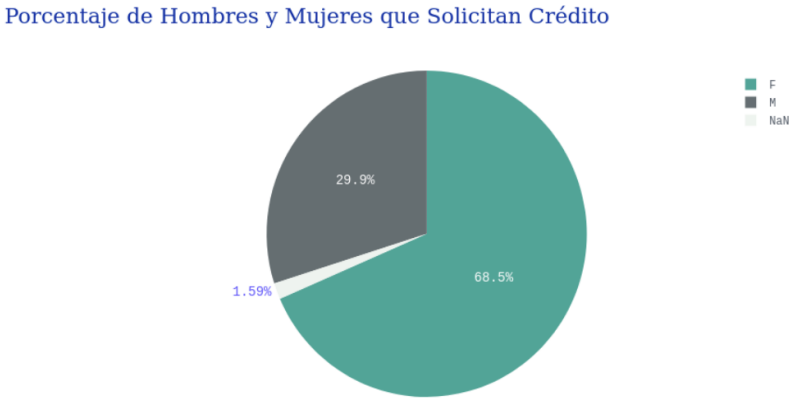
Contraste entre c_mate_income VS v_sex

- **c_personal_net_income**: En el siguiente gráfico se muestra la comparación entre la variable en cuestión y **v_sex.**

Contraste entre c_personal_net_income VS v_sex

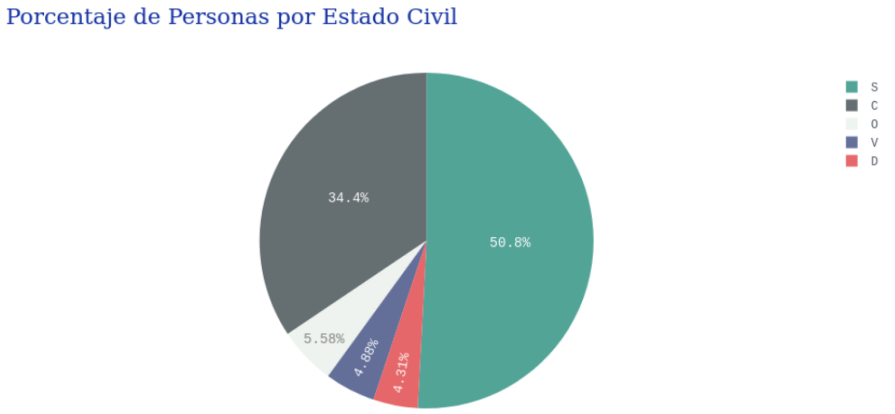


- **v_sex:** En el gráfico que se presenta a continuación tenemos el porcentaje de hombres y mujeres que solicitan un crédito.



Del gráfico anterior podemos ver que del total personas que solicitan un crédito, el 68.5% son mujeres, siendo esto importante de resaltar pues de las gráficas en donde comparamos el comportamiento de las variables: **c_mate_income** y **c_personal_net_income,** observamos que las mujeres son quienes mayor volumen en ingresos en ambas variables: en algunos casos se tienen registros con 80K y 26.6M, respectivamente para ambas tipos de ingreso.

- **v_marital_status**: En el siguiente gráfico se muestra la proporción de personal que solicitaron un crédito de acuerdo a su estado civil.



Podemos ver que en primera posición con el 50.8% tenemos a personas solteras seguido de personas caasadas con el 34.4%. Es importante señalar que de el 30.7% de las personas solteras que soliticitaron son hombres.

TRATAMIENTO I

- **Datos Anómalos**

Para la variable **c_age**, se aplicaron los métodos vistos en clase: **IQR**, **Percentiles** y **Z-Score**. A continuación se presentan los resultados obtenidos para cada uno de los métodos, y se explica cuáles de ellos se decidió implementar.

Para el método de de "**IQR"**, se obtuvieron los siguientes resultados:

	n_outliers_IQR	n_outliers_IQR_%
0	318	0.64

Por otra parte, mediante el método de "**Percentiles"** los resultados fueron los siguientes:

	n_outliers_Percentil	n_outliers_Percentil_%
0	4759	9.52

Finalmente, con "**Z-Score"** se detectó:

n_outliers_Z_Score	n_outliers_Z_Score_%
64	0.13

*** Los 2 métodos seleccionados para esta variable fueron: **IQR** y **Percentiles,** el Z-Score no se utilizó dado que los datos no son similares a una distribución normal.

Por otra parte para las variables: **c_mate_income** y **c_personal_net_income** también se utilizaron los métodos de **IQR** y **Percentiles**, y Z-Score fue descartado por la misma razón que se mencionó anteriormente (la distribución de los datos no tiene un comportamiento normal según las pruebas).

- **c_mate_income**

Utilizando IQR

	n_outliers_IQR	n_outliers_IQR_%
0	1962	3.85

Utilizando Percentiles

	n_outliers_Percentil	n_outliers_Percentil_%
0	1962	3.92

Utilizando Z-Score

n_outliers_Z_Score	n_outliers_Z_Score_%
165	0.33

- **c_personal_net_income**

Utilizando IQR

	n_outliers_IQR	n_outliers_IQR_%
0	4120	8.24

Utilizando Percentiles

	n_outliers_Percentil	n_outliers_Percentil_%
0	4352	8.7

Utilizando Z-Score

n_outliers_Z_Score	n_outliers_Z_Score_%
16	0.03

- **Datos Faltantes**

Al llegar a este punto, encontramos que todas las variables se encuentran en un 100% de completitud, razón por la cual no se consideró aplicar un tratamiento adicional para imputar missings.

	columna	total	completitud
0	v_id_client	0	100.000000
1	v_flag_card_insurance_option	0	100.000000
2	c_quant_additional_cards_in_the_application	0	100.000000
3	v_cod_application_booth	0	100.000000
4	c_personal_net_income	0	100.000000
5	v_flag_contact_phone	0	100.000000
6	v_flag_mobile_phone	0	100.000000
7	t_personal_reference_#2	0	100.000000
8	t_personal_reference_#1	0	100.000000
9	c_quant_banking_accounts	0	100.000000
10	v_flag_other_card	0	100.000000
11	v_flag_residencial_address=postal_address	0	100.000000
12	c_mate_income	0	100.000000
13	v_profession_code	0	100.000000
14	c_months_in_the_job	0	100.000000
15	v_flag_residence_state=working_state	0	100.000000
16	v_flag_residence_town=working_town	0	100.000000
17	v_flag_fathers_name	0	100.000000
18	v_flag_mothers_name	0	100.000000
19	c_months_in_residence	0	100.000000
20	v_residence_type	0	100.000000
21	c_shop_rank	0	100.000000
22	c_payment_day	0	100.000000
23	v_area_code_residencial_phone	0	100.000000
24	v_flag_residencial_phone	0	100.000000
25	c_quant_dependants	0	100.000000
26	c_age	0	100.000000
27	v_marital_status	0	100.000000
28	v_sex	0	100.000000
29	v_id_shop	0	100.000000
30	v_tgt	0	100.000000
31	profession	0	100.000000

- **Ingeniería de Datos**

En esta sección, utilizando sklearn, se lleva a cabo la partición entre el conjunto de prueba y entrenamiento del dataframe en cuestión (tratamiento_1), asignando 30% para el test.

Se empleó **"One-Hot Encoding"** para las variables categóricas que se listan a continuación:

'v_sex',
'v_marital_status',
'v_flag_residencial_phone',
'v_area_code_residencial_phone',
'v_residence_type',
'v_flag_mothers_name',
'v_flag_fathers_name',
'v_flag_residence_town=working_town',
'v_flag_residence_state=working_state',
'v_profession_code',
'v_flag_residencial_address=postal_address',
'v_flag_other_card',
'v_flag_mobile_phone',
'v_flag_contact_phone',
'v_cod_application_booth',
'v_flag_card_insurance_option',

'v_tgt'

Las dimensiones de los dataframes de entrenamiento y prueba, quedó como se muestra enseguida:
X_train => (35,000, 390)
X_test => (15,000, 390)

- **Reducción de Dimensiones**

Dentro de esta etapa del proceso, las variables que fueron eliminadas fueron:

*** Para identificar que las variables descritas en la parte de abajo debían ser eliminadas, se utilizó el **Filtro de Baja Varianza**. La tabla mostrada enseguida muestra el comportamiento de las variables continuas dentro de nuestro análisis (obtenida a través del método df.describe()).

Comportamiento de la Varianza en las Variables

	c_age	c_quant_dependants	c_payment_day	c_shop_rank	c_months_in_the_job	c_mate_income	c_quant_banking_accounts	c_pers
count	35000.000000	35000.0	35000.000000	35000.000000	35000.000000	35000.000000	35000.0	
mean	33.029600	0.0	15.329000	0.015200	50.490514	54.438229	0.0	
std	15.481688	0.0	7.150476	0.206602	74.219634	1055.991737	0.0	
min	0.000000	0.0	1.000000	0.000000	0.000000	0.000000	0.0	
10%	19.000000	0.0	8.000000	0.000000	0.000000	0.000000	0.0	
20%	21.000000	0.0	8.000000	0.000000	0.000000	0.000000	0.0	
30%	24.000000	0.0	12.000000	0.000000	12.000000	0.000000	0.0	
40%	27.000000	0.0	12.000000	0.000000	12.000000	0.000000	0.0	
50%	32.000000	0.0	12.000000	0.000000	12.000000	0.000000	0.0	
60%	36.000000	0.0	18.000000	0.000000	24.000000	0.000000	0.0	
70%	40.000000	0.0	20.000000	0.000000	48.000000	0.000000	0.0	
80%	45.000000	0.0	20.000000	0.000000	72.000000	0.000000	0.0	
90%	53.000000	0.0	25.000000	0.000000	144.000000	0.000000	0.0	
100%	372.000000	0.0	28.000000	3.000000	1176.000000	150000.000000	0.0	
max	372.000000	0.0	28.000000	3.000000	1176.000000	150000.000000	0.0	

- c_quant_dependants
- c_shop_rank
- c_quant_banking_accounts
- c_quant_additional_cards_in_the_application
- c_mate_income

Por otra parte, en el PCA se utilizaron las siguientes variables:

- c_age
- c_payment_day
- c_months_in_residence
- c_months_in_the_job
- c_personal_net_income

La varianza explicada por los 3 componentes definidos en este análisis explican el 0.9999999698745548 de la varianza. En la siguiente tabla se muestran los resultados arrojados por el PCA.

Valores Obtenidos para los 3 Componentes

	PC1	PC2	PC3	v_tgt
0	-7663.431991	-119.692152	-19.178480	0.0
1	-7626.432541	325.367867	-38.007451	NaN
2	-7153.431926	-156.616328	-38.453557	0.0
3	-7639.432069	-36.815063	-47.252575	NaN
4	-7953.432378	226.636864	-67.094084	0.0
...	...	...	...	...
34995	-7713.431967	-132.102039	-29.097697	0.0
34996	-6753.431957	-143.797000	-27.499236	NaN
34997	-7323.432416	262.177806	-70.694249	0.0
34998	-7603.432183	3.213324	8.581963	NaN
34999	-7593.432194	48.380228	-30.766996	1.0

Visualización 3D del PCA

Total de Varianza Explicada: 99.9999998616%