Diseño de solución analítica predictiva del índice ambiental de energía en una planta de producción de alimentos cárnicos de Bogotá

Juan Esteban Cano Ospina

Monografía presentada para optar al título de Especialista en Analítica y Ciencia de Datos

Asesora
Daniela Serna Buitrago, Especialista (Esp)

Universidad de Antioquia
Facultad de Ingeniería
Especialización en Analítica y Ciencia de Datos
Medellín, Antioquia, Colombia
2022

Tabla de contenido

Tabla de ilustraciones

RESUMEN EJECUTIVO

El término “sostenibilidad” hoy en día, debe ser una realidad en las organizaciones, un pilar estratégico en el posicionamiento en el mercado y relacionamiento con los stakeholders cómo clientes e instituciones, entre otros.

En específico, en las cadenas de suministro de la producción de alimentos, es relevante identificar, analizar y realizar seguimiento a variables de proceso que impacten los indicadores ambientales, cualquier decisión estratégica debe contemplar los efectos sobre estas variables.

En el presente proyecto se desarrolló una solución analítica para la proyección del consumo de energía, utilizando información de colocación (producto en proceso) en los recursos (máquinas y equipos) de una planta de producción.

De acuerdo con el diseño metodológico propuesto, se entendieron los datos, se prepararon, se modelaron utilizando diversas técnicas de aprendizaje automático de tipo regresivo y por último se evaluaron los resultados con las métricas tanto analíticas como de negocio.

Es interesante entender las iteraciones, los cambios y las conclusiones que sirven como insumo para futuros proyectos en el área.

Repositorio GitHub del proyecto: https://github.com/juanes-cano/MONOGRAF-A-EACD

DESCRIPCIÓN DEL PROBLEMA

Problema de negocio

Una empresa de producción de alimentos cárnicos de Bogotá requiere proyectar anualmente los consumos de energía eléctrica que se utilizaran para la producción de productos por mes.

El requerimiento hace parte de los lineamientos estratégicos de la organización desde la perspectiva del área ambiental y mantenimiento, las cuales son áreas transversales en el negocio.

Es un compromiso de la empresa responder adecuadamente con sus stakeholders como lo es el gobierno y seguir cumpliendo con los lineamientos del buen uso de los recursos naturales.

Lo anterior enmarcado en las políticas públicas ambientales y como una huella de generación de valor en la responsabilidad social corporativa.

Aproximación desde la analítica de datos

El ejercicio predictivo tiene el propósito de realizar el seguimiento y mejora continua a procesos para la disminución de los efectos adversos generados al medio ambiente y promover el uso consciente de los recursos naturales.

La necesidad se podría resolver para el ámbito productivo desde la información de colocación de kilogramos y ocupación en horas de los recursos (equipos o máquinas) de las líneas de producción y su relación con la variación del consumo en kWh (kilowatts por hora).

Se desarrollarán modelos predictivos para predecir el consumo diario de la planta productiva en energía, dado el proyectado táctico de colocación de kilogramos y uso en horas de cada recurso además de algunas variables adicionales temporales.

El modelo servirá para aumentar la asertividad de la proyección y asegurar un resultado más confiable al grupo de interesados estratégicos como apuesta de negocio en el seguimiento de indicadores ambientales y en el control de mitigación de impactos.

Origen de los datos

Los datos que se tienen a disposición representan un conjunto de información que reúne la colocación de material en cada uno de los recursos (máquinas o equipos) de producción en kilogramos (kg) y horas (Hrs) a detalle día desde el 12 de febrero del 2018 hasta el mes de noviembre del 2021. Los datos se obtuvieron a partir del equipo de capacidades y rendimientos al igual que de analistas del área ambiental del negocio.

Por lo tanto, los datos son un total de 1372 registros o muestras y 98 variables o columnas, de las cuales 97 hacen parte de las predictoras.

Métricas de desempeño

Métrica de Machine Learning: Por ser un ejercicio de regresión, la métrica de machine learning a evaluar será el "explained_variance_score", la varianza explicada, se usa para medir la discrepancia entre un modelo y los datos reales. En otras palabras, es la parte de la varianza total del modelo que se explica por factores que realmente están presentes y no se debe a la varianza del error.

Los porcentajes más altos de varianza explicada indican una mayor fuerza de asociación. También significa que hace mejores predicciones (Rosenthal & Rosenthal, 2011).

Métrica de negocio: Las métricas para la organización es la asertividad de la predicción con respecto al real y la comparación de la desviación entre el proyectado y el real.

El valor mínimo, aceptado de la métrica de asertividad en machine learning por parte de los stakeholders es del 80%, lo anterior, justifica el esfuerzo del equipo analítico para ajustar y minimizar las diferencias entre el proyectado y el real.

DATOS

Datos Originales

Los datos con lo que se cuentan, se encuentra en un solo archivo en formato CSV (delimitado por comas), en total son 98 columnas, 1372 registros, con un tamaño total del archivo de 610 kB.

A continuación, se describirán las columnas:

'Fecha': Fecha, detalle día.
'Semana': Semana del año.
'DiaMes': Dia del mes.
'Mes': Mes del año.
'DiasSem': Dia de la Semana.
'KG_15EMPA09': Kilogramos colocados en el recurso EMPA09
'KG_15EMPA10': Kilogramos colocados en el recurso EMPA10
'KG_15PREM01': Kilogramos colocados en el recurso PREM01
'KG_15MEZC08': Kilogramos colocados en el recurso MEZC08
'KG_15MOLI07': Kilogramos colocados en el recurso MOLI07
'KG_15AHUM10': Kilogramos colocados en el recurso AHUM10
'KG_15FORM02': Kilogramos colocados en el recurso FORM02
'KG_15SEPA02': Kilogramos colocados en el recurso SEPA02
'KG_15MEZC06': Kilogramos colocados en el recurso MEZC06
'KG_15AHUM01': Kilogramos colocados en el recurso AHUM01
'KG_15FORM01': Kilogramos colocados en el recurso FORM01
'KG_15SEPA01': Kilogramos colocados en el recurso SEPA01

'KG_15AHUM09': Kilogramos colocados en el recurso AHUM09
'KG_15EMBU01'. Kilogramos colocados en el recurso EMBU09
'KG_15EMPM06': Kilogramos colocados en el recurso EMPM06
'KG_15TUNE01': Kilogramos colocados en el recurso TUNE01
'KG_15MEZC07': Kilogramos colocados en el recurso MEZC07
'KG_15MOLI01': Kilogramos colocados en el recurso MOLI01
'KG_15MOLI06': Kilogramos colocados en el recurso MOLI06
'KG_15EMPA11': Kilogramos colocados en el recurso EMPA11
'KG_15EMPM01': Kilogramos colocados en el recurso EMPM01
'KG_15MEZC05': Kilogramos colocados en el recurso MEZC05
'KG_15MOLI03': Kilogramos colocados en el recurso MOLI03
'KG_15HORN01': Kilogramos colocados en el recurso HORN01
'KG_15PORC03': Kilogramos colocados en el recurso PORC03
'KG_15TUNE03': Kilogramos colocados en el recurso TUNE03
'KG_15EMPA06': Kilogramos colocados en el recurso EMPA06
'KG_15EMPA02': Kilogramos colocados en el recurso EMPA02
'KG_15MOLI02': Kilogramos colocados en el recurso MOLI02
'KG_15SELL02': Kilogramos colocados en el recurso SELL02
'KG_15MEZC04': Kilogramos colocados en el recurso MEZC04
'KG_15MOLI08': Kilogramos colocados en el recurso MOLI08
'KG_15TUNE02': Kilogramos colocados en el recurso TUNE02
'KG_15PORC01': Kilogramos colocados en el recurso PORC01
'KG_15TUNE04': Kilogramos colocados en el recurso TUNE04
'KG_15EMBU03': Kilogramos colocados en el recurso EMBU03
'KG_15EMPA08': Kilogramos colocados en el recurso EMPA08
'KG_15EMPM05': Kilogramos colocados en el recurso EMPM05
'KG_15COCI01': Kilogramos colocados en el recurso COCI01
'KG_15EMPA04': Kilogramos colocados en el recurso EMPA04
'KG_15TAJA02': Kilogramos colocados en el recurso TAJA02
'KG_15EMUL01': Kilogramos colocados en el recurso EMUL01
'KG_15SELL05': Kilogramos colocados en el recurso SELL05

'KG_15DOSI01': Kilogramos colocados en el recurso DOSI01
'KG_15SELL06': Kilogramos colocados en el recurso SELL06
'KG_15SELL03': Kilogramos colocados en el recurso SELL03
'Hrs_15EMPA09': Horas ocupadas el recurso EMPA09
'Hrs_15EMPA10': Horas ocupadas el recurso EMPA10
'Hrs_15PREM01': Horas ocupadas el recurso PREM01
'Hrs_15MEZC08': Horas ocupadas el recurso MEZC08
'Hrs_15MOLI07': Horas ocupadas el recurso MOLI07
'Hrs_15AHUM10': Horas ocupadas el recurso AHUM10
'Hrs_15FORM02': Horas ocupadas el recurso FORM02
'Hrs_15SEPA02': Horas ocupadas el recurso SEPA02
'Hrs_15MEZC06': Horas ocupadas el recurso MEZC06
'Hrs_15AHUM01': Horas ocupadas el recurso AHUM01
'Hrs_15FORM01': Horas ocupadas el recurso FORM01
'Hrs_15SEPA01': Horas ocupadas el recurso SEPA01
'Hrs_15AHUM09': Horas ocupadas el recurso AHUM09
'Hrs_15EMBU01': Horas ocupadas el recurso EMBU01
'Hrs_15EMPM06': Horas ocupadas el recurso EMPM06
'Hrs_15TUNE01': Horas ocupadas el recurso TUNE01
'Hrs_15MEZC07': Horas ocupadas el recurso MEZC07
'Hrs_15MOLI01': Horas ocupadas el recurso MOLI01
'Hrs_15MOLI06': Horas ocupadas el recurso MOLI06
'Hrs_15EMPA11': Horas ocupadas el recurso EMPA11
'Hrs_15EMPM01': Horas ocupadas el recurso EMPM01
'Hrs_15MEZC05': Horas ocupadas el recurso MEZC05
'Hrs_15MOLI03': Horas ocupadas el recurso MOLI03
'Hrs_15HORN01': Horas ocupadas el recurso HORN01
'Hrs_15PORC03': Horas ocupadas el recurso PORC03
'Hrs_15TUNE03': Horas ocupadas el recurso TUNE03
'Hrs_15EMPA06': Horas ocupadas el recurso EMPA06
'Hrs_15EMPA02': Horas ocupadas el recurso EMPA02

'Hrs_15MOLI02': Horas ocupadas el recurso MOLI02
'Hrs_15SELL02': Horas ocupadas el recurso SELL02
'Hrs_15MEZC04': Horas ocupadas el recurso MEZC04
'Hrs_15MOLI08': Horas ocupadas el recurso MOLI08
'Hrs_15TUNE02': Horas ocupadas el recurso TUNE02
'Hrs_15PORC01': Horas ocupadas el recurso PORC01
'Hrs_15TUNE04': Horas ocupadas el recurso TUNE04
'Hrs_15EMBU03': Horas ocupadas el recurso EMBU03
'Hrs_15EMPA08': Horas ocupadas el recurso EMPA08
'Hrs_15EMPM05': Horas ocupadas el recurso EMPM05
'Hrs_15COCI01': Horas ocupadas el recurso COCI01
'Hrs_15EMPA04': Horas ocupadas el recurso EMPA04
'Hrs_15TAJA02': Horas ocupadas el recurso TAJA02
'Hrs_15EMUL01': Horas ocupadas el recurso EMUL01
'Hrs_15SELL05': Horas ocupadas el recurso SELL05
'Hrs_15DOSI01': Horas ocupadas el recurso DOSI01
'Hrs_15SELL06': Horas ocupadas el recurso SELL06
'Hrs_15SELL03': Horas ocupadas el recurso SELL03
'energia': Consumo de energía eléctrica en kWh.

La variable respuesta o a predecir es ‘energía’

El modo de acceso a los datos se encuentra en el repositorio, con el nombre “data_csv.csv”.

Datasets

Los datasets de entrenamiento y prueba se crearon con la función “train_test_split” con un tamaño del dataset de prueba de un 30%.

Del dataset de entrenamiento se creó el dataset de validación, el cual se utiliza en el GridSearch CV con la técnica de validación cruzada RepeatedKFold con 10 particiones y se repite 2 veces.

Descriptiva

A continuación, se observa la capa de los datos resumida como carga de datos en un dataframe de pandas.

	Fecha	Semana	DiaMes	Mes	DiasSem	KG_15EMPA09	KG_15EMPA10	KG_15PREM01	KG_15MEZC08	KG_15MOLI07	...
0	12/02/2018	7	12	2	1	22201	23849	23	48618	54361	...
1	13/02/2018	7	13	2	2	24285	26285	24	53509	77637	...
2	14/02/2018	7	14	2	3	20939	16431	14	42169	43799	...
3	15/02/2018	7	15	2	4	27570	29001	30	58535	83841	...
4	16/02/2018	7	16	2	5	24085	23171	21	48828	56316	...
...	...	...	...	...	...	...	...	...	...	...	...
1367	10/11/2021	45	10	11	3	35814	35284	21	69560	86573	...
1368	11/11/2021	45	11	11	4	38214	40601	27	72945	99950	...
1369	12/11/2021	45	12	11	5	34014	35996	12	69003	87102	...
1370	13/11/2021	45	13	11	6	30813	28107	24	53371	82392	...
1371	14/11/2021	45	14	11	7	2337	3382	6	6123	9539	...

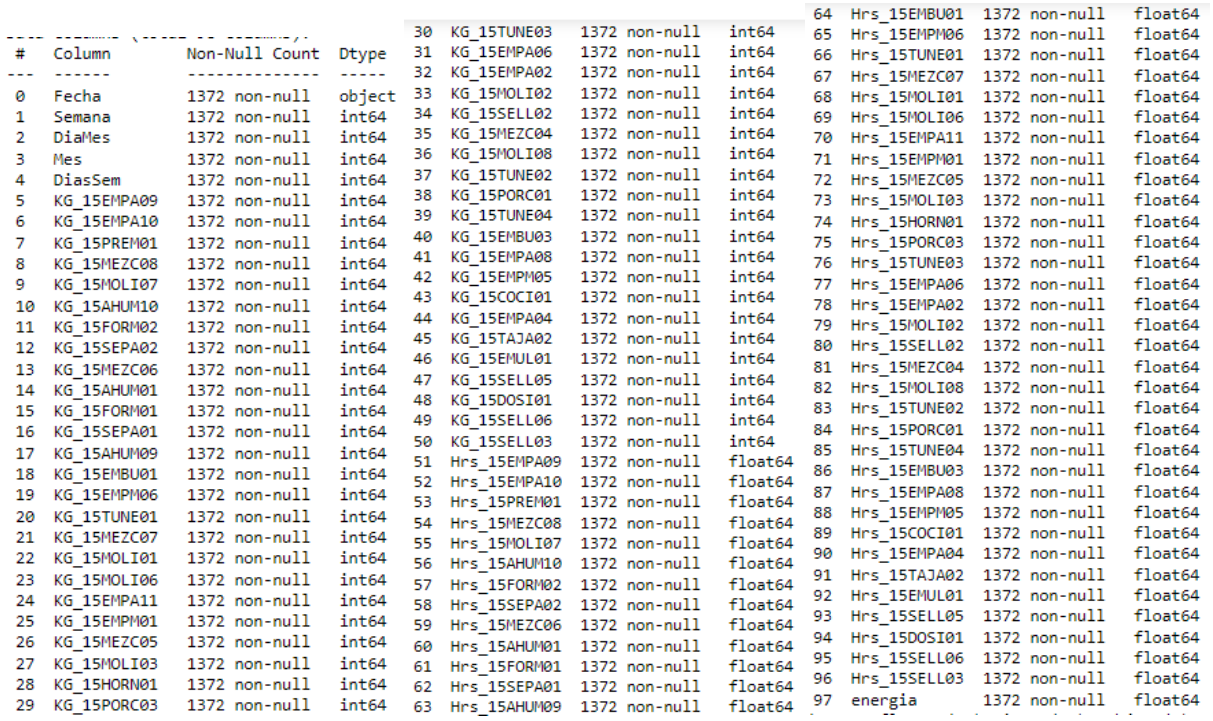
1372 rows × 98 columns

Ilustración 1: Capa de datos originales resumida - Parte 1 (Elaboración propia)

...	Hrs_15EMPM05	Hrs_15COCI01	Hrs_15EMPA04	Hrs_15TAJA02	Hrs_15EMUL01	Hrs_15SELL05	Hrs_15DOSI01	Hrs_15SELL06	Hrs_15SELL03	energia
...	2.53	2.35	8.16	8.16	0.26	0.00	11.80	2.95	0.0	35797.68
...	3.93	3.66	5.83	5.83	0.18	0.60	20.65	0.00	2.0	36442.08
...	3.98	3.70	8.52	8.52	0.27	3.04	10.86	2.96	0.0	31412.52
...	3.95	3.68	9.09	9.09	0.29	3.14	20.96	3.00	0.0	39806.88
...	4.00	3.72	6.58	6.58	0.21	3.30	11.46	2.95	0.0	34684.20
...	...	...	...	...	...	...	...	...	...	...
...	0.00	0.00	6.51	6.51	16.38	16.01	0.00	5.79	0.0	46592.00
...	0.00	0.00	6.28	6.33	26.94	16.37	3.86	2.91	0.0	46792.00
...	12.85	11.96	6.09	6.09	23.49	0.00	0.00	2.99	0.0	47580.00
...	0.00	0.00	6.47	6.47	16.87	15.72	0.00	6.16	0.0	19119.00
...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0	14387.00

Ilustración 2: Capa de datos originales resumida - Parte 2 (Elaboración propia)

En las siguientes imágenes, se identifica información respecto a la cantidad de datos no nulos por variable o columna y el tipo de dato reconocido por pandas.



#	Column	Non-Null Count	Dtype
0	Fecha	1372 non-null	object
1	Semana	1372 non-null	int64
2	DiaMes	1372 non-null	int64
3	Mes	1372 non-null	int64
4	DiasSem	1372 non-null	int64
5	KG_15EMPA09	1372 non-null	int64
6	KG_15EMPA10	1372 non-null	int64
7	KG_15PREM01	1372 non-null	int64
8	KG_15MEZC08	1372 non-null	int64
9	KG_15MOLI07	1372 non-null	int64
10	KG_15AHUM10	1372 non-null	int64
11	KG_15FORM02	1372 non-null	int64
12	KG_15SEPA02	1372 non-null	int64
13	KG_15MEZC06	1372 non-null	int64
14	KG_15AHUM01	1372 non-null	int64
15	KG_15FORM01	1372 non-null	int64
16	KG_15SEPA01	1372 non-null	int64
17	KG_15AHUM09	1372 non-null	int64
18	KG_15EMBU01	1372 non-null	int64
19	KG_15EMPM06	1372 non-null	int64
20	KG_15TUNE01	1372 non-null	int64
21	KG_15MEZC07	1372 non-null	int64
22	KG_15MOLI01	1372 non-null	int64
23	KG_15MOLI06	1372 non-null	int64
24	KG_15EMPA11	1372 non-null	int64
25	KG_15EMPM01	1372 non-null	int64
26	KG_15MEZC05	1372 non-null	int64
27	KG_15MOLI03	1372 non-null	int64
28	KG_15HORN01	1372 non-null	int64
29	KG_15PORC03	1372 non-null	int64
30	KG_15TUNE03	1372 non-null	int64
31	KG_15EMPA06	1372 non-null	int64
32	KG_15EMPA02	1372 non-null	int64
33	KG_15MOLI02	1372 non-null	int64
34	KG_15SELL02	1372 non-null	int64
35	KG_15MEZC04	1372 non-null	int64
36	KG_15MOLI08	1372 non-null	int64
37	KG_15TUNE02	1372 non-null	int64
38	KG_15PORC01	1372 non-null	int64
39	KG_15TUNE04	1372 non-null	int64
40	KG_15EMBU03	1372 non-null	int64
41	KG_15EMPA08	1372 non-null	int64
42	KG_15EMPM05	1372 non-null	int64
43	KG_15COCI01	1372 non-null	int64
44	KG_15EMPA04	1372 non-null	int64
45	KG_15TAJA02	1372 non-null	int64
46	KG_15EMUL01	1372 non-null	int64
47	KG_15SELL05	1372 non-null	int64
48	KG_15DOSI01	1372 non-null	int64
49	KG_15SELL06	1372 non-null	int64
50	KG_15SELL03	1372 non-null	int64
51	Hrs_15EMPA09	1372 non-null	float64
52	Hrs_15EMPA10	1372 non-null	float64
53	Hrs_15PREM01	1372 non-null	float64
54	Hrs_15MEZC08	1372 non-null	float64
55	Hrs_15MOLI07	1372 non-null	float64
56	Hrs_15AHUM10	1372 non-null	float64
57	Hrs_15FORM02	1372 non-null	float64
58	Hrs_15SEPA02	1372 non-null	float64
59	Hrs_15MEZC06	1372 non-null	float64
60	Hrs_15AHUM01	1372 non-null	float64
61	Hrs_15FORM01	1372 non-null	float64
62	Hrs_15SEPA01	1372 non-null	float64
63	Hrs_15AHUM09	1372 non-null	float64
64	Hrs_15EMBU01	1372 non-null	float64
65	Hrs_15EMPM06	1372 non-null	float64
66	Hrs_15TUNE01	1372 non-null	float64
67	Hrs_15MEZC07	1372 non-null	float64
68	Hrs_15MOLI01	1372 non-null	float64
69	Hrs_15MOLI06	1372 non-null	float64
70	Hrs_15EMPA11	1372 non-null	float64
71	Hrs_15EMPM01	1372 non-null	float64
72	Hrs_15MEZC05	1372 non-null	float64
73	Hrs_15MOLI03	1372 non-null	float64
74	Hrs_15HORN01	1372 non-null	float64
75	Hrs_15PORC03	1372 non-null	float64
76	Hrs_15TUNE03	1372 non-null	float64
77	Hrs_15EMPA06	1372 non-null	float64
78	Hrs_15EMPA02	1372 non-null	float64
79	Hrs_15MOLI02	1372 non-null	float64
80	Hrs_15SELL02	1372 non-null	float64
81	Hrs_15MEZC04	1372 non-null	float64
82	Hrs_15MOLI08	1372 non-null	float64
83	Hrs_15TUNE02	1372 non-null	float64
84	Hrs_15PORC01	1372 non-null	float64
85	Hrs_15TUNE04	1372 non-null	float64
86	Hrs_15EMBU03	1372 non-null	float64
87	Hrs_15EMPA08	1372 non-null	float64
88	Hrs_15EMPM05	1372 non-null	float64
89	Hrs_15COCI01	1372 non-null	float64
90	Hrs_15EMPA04	1372 non-null	float64
91	Hrs_15TAJA02	1372 non-null	float64
92	Hrs_15EMUL01	1372 non-null	float64
93	Hrs_15SELL05	1372 non-null	float64
94	Hrs_15DOSI01	1372 non-null	float64
95	Hrs_15SELL06	1372 non-null	float64
96	Hrs_15SELL03	1372 non-null	float64
97	energia	1372 non-null	float64

Ilustración 3: Información por variable (Elaboración propia)

Se puede afirmar de lo anterior que no se encuentran datos nulos en ningún campo del dataset. Por otro lado se observa que algunas variables como “Semana”, “DiaMes”, ”Mes”, “DiaSem” no son de tipo “int” (entero), sino que son de tipo “category” o categórico, lo cual se deberá tomar en cuenta en la etapa de preparación de datos.

En las siguientes imágenes, se observa una descripción general de cada variable o campo del dataset a trabajar.

	Semana	DiaMes	Mes	DiaSem	KG_15EMPA09	KG_15EMPA10	KG_15PREM01	KG_15MEZC08	KG_15MOLI07	KG_15AHUM10
count	1372.000000	1372.000000	1372.000000	1372.000000	1372.000000	1372.000000	1372.000000	1372.000000	1372.000000	1372.000000
mean	26.535714	15.716472	6.508017	4.000000	23114.567784	23457.238338	18.024781	45460.644315	57455.849854	43407.661808
std	14.431489	8.790980	3.302731	2.000729	11927.103727	11966.736649	9.893052	23341.702385	29513.264844	22320.486575
min	1.000000	1.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	14.000000	8.000000	4.000000	2.000000	20942.000000	21463.250000	14.000000	40928.250000	48425.500000	38920.750000
50%	26.500000	16.000000	7.000000	4.000000	26460.500000	27364.500000	20.000000	52986.000000	67458.500000	50464.500000
75%	39.000000	23.000000	9.000000	6.000000	31003.750000	31380.500000	25.000000	60980.750000	78227.750000	58315.500000
max	53.000000	31.000000	12.000000	7.000000	49131.000000	49027.000000	45.000000	97234.000000	105192.000000	93106.000000

Ilustración 4: Descripción general de cada variable (Elaboración propia)

Hrs_15EMPM05	Hrs_15COCI01	Hrs_15EMPA04	Hrs_15TAJA02	Hrs_15EMUL01	Hrs_15SELL05	Hrs_15DOSI01	Hrs_15SELL06	Hrs_15SELL03	energia
1372.000000	1372.000000	1372.000000	1372.000000	1372.000000	1372.000000	1372.000000	1372.000000	1372.000000	1372.000000
1.729213	1.609752	5.473149	5.474891	7.922558	5.291764	1.586385	3.219759	0.266443	34565.356735
3.181216	2.961440	3.834628	3.834934	8.253332	4.801806	3.430366	2.848040	0.704311	10154.919005
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	4431.240000
0.000000	0.000000	0.855000	0.855000	0.190000	0.000000	0.000000	0.000000	0.000000	31635.450000
0.000000	0.000000	6.120000	6.130000	0.405000	6.155000	0.000000	2.970000	0.000000	35773.560000
2.770000	2.572500	7.890000	7.890000	16.345000	9.330000	3.652500	5.550000	0.000000	42567.480000
16.660000	15.510000	17.490000	17.490000	31.840000	32.480000	33.240000	19.160000	3.990000	78768.720000

Ilustración 5: Descripción general de cada variable - Parte 2 (Elaboración propia)

Por la gran cantidad de variables o campos, es complejo entender de forma adecuada la información en formato tabla, además de conllevar a requerir una gran cantidad de capacidad computacional, por lo cual, a continuación, se analizará una submuestra del conjunto de datos, seleccionando únicamente las variables numéricas para realizar una matriz de correlación de Pearson, con el fin de explorar un poco mejor las relaciones.

Ilustración 6: Matriz de correlación de variables numéricas (Elaboración propia)

Se puede afirmar de la matriz de correlación anterior, que se observa presencia de colinealidad positiva entre las variables predictoras en algunos casos, posteriormente se analizará la importancia de características como resultado adicional de algunas tipologías de modelos que se emplearan y permiten este analisis, con respecto a la variable respuesta también se observan relaciones positivas, la mayoría entre moderadas-fuertes, esto indica que las variables consideradas tienen un adecuado potencial predictor respecto a la solución que se pretende a generar.

Ahora, se explorara la presencia de outliers o atípicos por cada variable, para ello se escalará con la función MinMaxScaler y luego se graficará un boxplot con los parámetros predeterminadas, el resultado se ilustra a continuación:

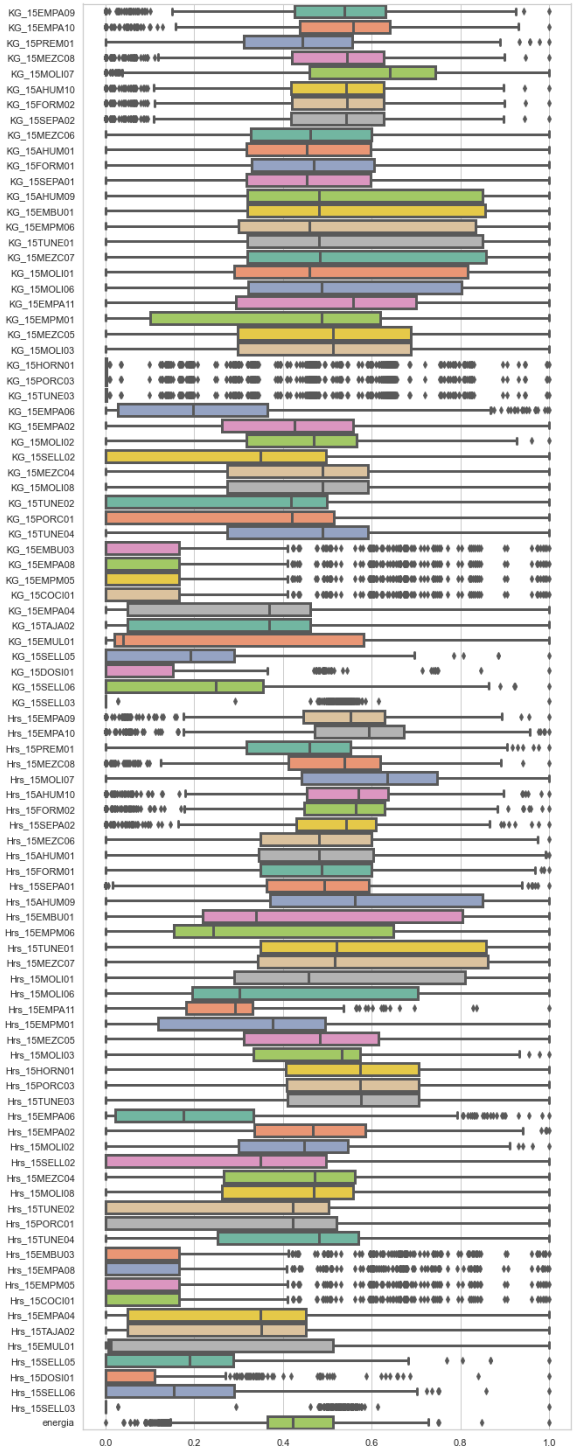


Ilustración 7: Boxplot de las variables (Elaboración propia)

De la anterior ilustración, se puede afirmar que en la mayoría de los casos si se presentan outliers, estos, sin embargo, son muy cercanos a la figura del boxplot (“caja”) para cada variable, menos para el horno 01, la porcionadora 03, el túnel 03 y la selladora 03, que presentan demasiados outliers, esto puede ser debido a una inconsistencia en los datos ya que las diferencias entre un dato y otro son muy grandes tanto en Kg, como en horas.

Otros recursos a revisar serían la embutidora 03, la empacadora automática 08, la empacadora manual 05, el cocinador 01 y la dosificadora 01, los cuales tienen outliers importantes, aproximadamente del 60% de los valores superiores.

PROCESO DE ANALITICA

Pipeline Principal

El pipeline principal para el desarrollo del proceso de analítica para este proyecto se basa en la metodología CRISP-DM (Schröer et al., 2021) framework altamente utilizado en la industria, en el siguiente grafico se ilustra el flujo de trabajo que se describirá a continuación:

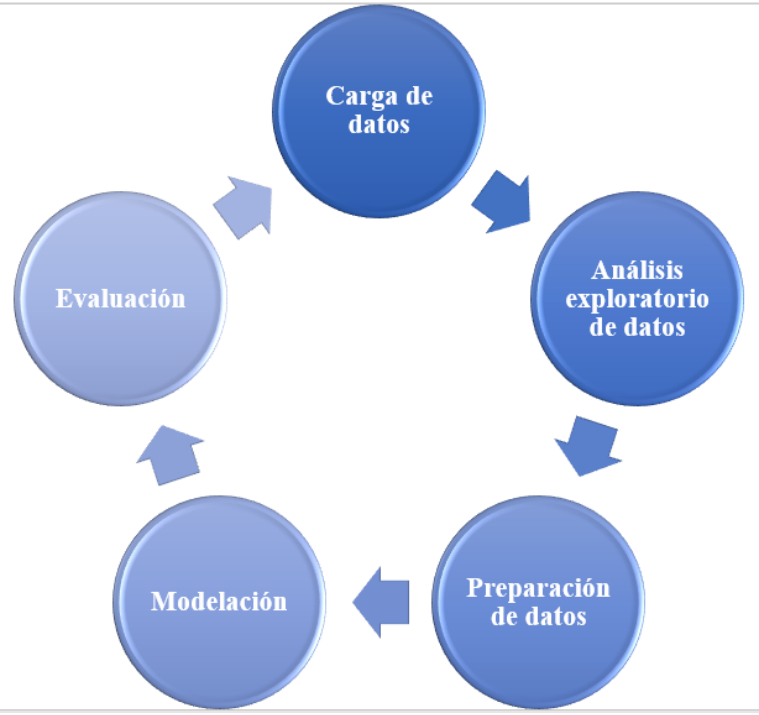


Ilustración 8: Diseño metodológico propuesto (Elaboraciòn propia)

El flujo de trabajo general del proyecto se compone de 5 etapas, que se describen a continuación:

1. Carga de datos: En esta etapa, cargamos los datos desde el archivo CSV a la herramienta seleccionada para realizar el proyecto analítico, la cual será Jupyter Notebook, se cargará en una estructura de datos que permita su preparación, modelación y evaluación de forma ágil y flexible.
2. Análisis exploratorio de datos: En esta etapa, se desarrollarán análisis y gráficas descriptivas para explorar la información, como distribuciones, relaciones, outliers o información de interés de la información.

3. Preparación de datos: Con los insights de la etapa anterior, se transforman los datos, de forma adecuada para la correcta modelación.

4. Modelación: Se aplican los métodos y técnicas adecuadas de modelación con algoritmos de machine learning para el ejercicio predictivo.

5. Evaluación: Muy ligada a la etapa anterior, se evalúa el desempeño de los modelos para la selección del mejor de acuerdo a varias variables a considerar.

6.

Preprocesamiento

La primera transformación del preprocesamiento de los datos será la conversión de algunas variables tipo “int” en “category” ya que esta es la real naturaleza de las variables, se ilustra a continuación el proceso y resultado.

```
data["Semana"]= data["Semana"].astype('category')
data["DiaMes"]= data["DiaMes"].astype('category')
data["Mes"]= data["Mes"].astype('category')
data["DiasSem"]= data["DiasSem"].astype('category')
```

```
data
```

	Fecha	Semana	DiaMes	Mes	DiasSem	KG_15EMPA09	KG_15EMPA10
0	12/02/2018	7	12	2	1	22201	23849
1	13/02/2018	7	13	2	2	24285	26285
2	14/02/2018	7	14	2	3	20939	16431
3	15/02/2018	7	15	2	4	27570	29001
4	16/02/2018	7	16	2	5	24085	23171
...	...	...	...	...	...	...	...
1367	10/11/2021	45	10	11	3	35814	35284
1368	11/11/2021	45	11	11	4	38214	40601
1369	12/11/2021	45	12	11	5	34014	35996
1370	13/11/2021	45	13	11	6	30813	28107
1371	14/11/2021	45	14	11	7	2337	3382

1372 rows × 98 columns

Ilustración 9: Preparación de datos - transformación de tipo de dato de variables (Elaboración propia)

La segunda transformación será volver dummies algunas variables ya categóricas como lo son “Semana”, “DiaMes”, “Mes” y “DiaSem” con la opción drop (eliminar) la primera variable por ser redundante, esta transformación permitirá desarrollar algunos modelos muy interesantes que requieren que todas las variables en X o predictoras sean cuantitativas, el proceso y resultado se ilustra a continuación:

```
data_con_var_fechas = pd.get_dummies(data, columns=['Semana', 'DiaMes','Mes','DiasSem'], drop_first=True)
```

	Fecha	KG_15EMPA09	KG_15EMPA10	KG_15PREM01
0	12/02/2018	22201	23849	23
1	13/02/2018	24285	26285	24
2	14/02/2018	20939	16431	14
3	15/02/2018	27570	29001	30
4	16/02/2018	24085	23171	21
...	...	...	...	...
1367	10/11/2021	35814	35284	21
1368	11/11/2021	38214	40601	27
1369	12/11/2021	34014	35996	12
1370	13/11/2021	30813	28107	24
1371	14/11/2021	2337	3382	6

1372 rows × 193 columns

Ilustración 10: Preparación de datos - transformación dummies (Elaboración propia)

Ahora, se “dropeara” o eliminará la variable o campo fecha por no ser relevante a-priori para el ejercicio. Se ilustra a continuación el proceso y resultado:

```
1 data_con_var_fechas_2 = data_con_var_fechas.drop("Fecha", axis=1)
```

	KG_15EMPA09	KG_15EMPA10	KG_15PREM01	KG_15MEZC08	KG_15MOLI07	KG_15AHUM10	KG_15FORM02	KG_15SEPA02	KG_15MEZC06	KG_15AHUM
0	22201	23849	23	48618	54361	46131	48618	46131	24837	237
1	24285	26285	24	53509	77637	51248	53619	51248	26594	250
2	20939	16431	14	42169	43799	40047	42169	40047	20785	198
3	27570	29001	30	58535	83841	56028	58652	56028	27671	260
4	24085	23171	21	48828	56316	46376	48829	46376	26747	255
...	...	...	...	...	...	...	...	...	...	
1367	35814	35284	21	69560	86573	66471	69642	66471	37825	362
1368	38214	40601	27	72945	99950	69734	73139	69734	37718	365
1369	34014	35996	12	69003	87102	66009	69102	66009	33797	327
1370	30813	28107	24	53371	82392	50950	53464	50950	42023	406
1371	2337	3382	6	6123	9539	5749	6123	5749	3416	31

1372 rows × 191 columns

Ilustración 11: Preparación de datos - eliminación de variables (Elaboración propia)

Modelos

La lista de modelos que se desarrollaron en el proyecto, de acuerdo con la naturaleza del ejercicio. son los siguientes:

1. Regresión lineal múltiple por mínimos cuadrados
2. ElasticNet
3. k-nearest neighbors (KNN) regressor
4. Decision Tree Regressor

Las configuraciones de cada modelo se realizaron por optimización de los parámetros más relevantes de cada modelo de acuerdo con la documentación. Para lo anterior se utilizó el método GridSearchCV con técnica de validación cruzada (CV) Repeated K Fold.

Métricas

La métrica de desempeño ML se calcula como la explicación de la varianza, la cual mide la proporción a la que un modelo matemático explica la variación de un conjunto de datos dado. Matemáticamente se explica a continuación:

$$explained\_variance(y, \hat{y}) = 1 - \frac{Var\{y - \hat{y}\}}{Var\{y\}}$$

Siendo:

$\hat{y}$: resultado objetivo estimado

y: resultado objetivo correcto

Var: Varianza

Se utiliza la función de la librería **sklearn.metrics**.explained_variance_score

La métrica de desempeño para el negocio, se mide de acuerdo con la desviación con respecto al real, es decir, la diferencia porcentual del valor real y el valor proyectado, según el modelo actual y el propuesto.

METODOLOGÍA

Baseline

Una vez obtenido el dataset final, luego de la preparación de los datos, se realizó el proceso de modelación, seleccionando el modelo con mayor interpretabilidad de la lista a evaluar y ese fue la regresión lineal múltiple por mínimos cuadrados.

Los resultados no fueron los esperados, la salida de coeficientes era confusa y el resultado de la métrica dio totalmente fuera de rango. Lo anterior, posiblemente debido a mucho "peso" o importancia a variables irrelevantes.

Validación

El proceso de particiones inició con la función train_test_split, en la cual se configuró como test_size = 0.3, luego de ello, se escaló los conjuntos "X" con la función MinMaxScaler, después se aplica la técnica de validación cruzada RepeatedKFold con 10 particiones y 2 repeticiones, técnica ampliamente utilizada para medir correctamente los resultados de un modelo y para la

detección de sobre-ajuste o sub-estimación (Refaeilzadeh et al., 2016). Lo anteriormente descrito, se ilustra a continuación:

Modelación

Realizamos la partición train-test de nuestro conjunto de datos con un tamaño del test del 30%

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(data_con_var_fechas_sin_y_2, data_con_var_fechas_2['energia'], \
                                                    test_size=0.3, random_state=0)
```

A continuación realizaremos la estadarización de la escala de las variables ya que las unidades de las variables son diferentes algunos en Kilogramos y otras en Horas

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
sc = scaler.fit(data_con_var_fechas_sin_y_2)
X_train_n = sc.transform(X_train)
X_test_n = sc.transform(X_test)
```

Como tecnica de validación cruzada, se utilizará "RepeatedKFold"

```
from sklearn.model_selection import RepeatedKFold
rkf = RepeatedKFold(n_splits=10, n_repeats=2, random_state=1)
```

Ilustración 12: Preparación de datos - partición de data set, escalamiento y técnica de validación cruzada (Elaboración propia)

Todo lo anterior se toma en cuenta para la función GridSearchCV que se utiliza posteriormente.

Iteraciones y evolución

Posteriormente de analizar el proceso y los resultados de la primera iteración, se propuso la utilización de modelos un poco más complejos, para que, de algún modo buscar y perseguir la maximización del ajuste del entrenamiento del modelo y de manera subsecuente la validación, sin quedar en la subestimación o el sobre-entrenamiento.

ElasticNet

Propuesto por Zou y Hastie (2005), es un modelo de regresión lineal que normaliza el vector de coeficientes con las normas L1 y L2. Esto permite generar un modelo en el que solo algunos de los coeficientes sean no nulos. El proceso se ilustra a continuación:

```
from sklearn.linear_model import ElasticNet
```

```
mod2 = ElasticNet(random_state=0)

parameters = {'alpha':[0,0.01,0.05,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1], \
              'l1_ratio':[0,0.01,0.05,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1]}

mod2_1 = GridSearchCV(estimator=mod2, param_grid = parameters, cv=rkf, scoring='explained_variance',return_train_score=True,
mod2_1.fit(X_train_n, y_train)
```

```
Fitting 20 folds for each of 169 candidates, totalling 3380 fits
```

Ilustración 13: Parametrización Modelo ElasticNet (Elaboraciòn propia)

Como se puede observar, en el fragmento de código anterior, los parámetros a iterar son “alpha” y “l1_ratio” en un rango de 0 a 1, con diferentes valores, estos parámetros permiten ajustar el tamaño de la regularización o normalización de los coeficientes del modelo lineal, también se observa la utilización del GridSearchCV y rkf (Repeated K Fold) de acuerdo con los lineamientos de las mejoras prácticas en desarrollo de soluciones analíticas.

k-nearest neighbors (KNN) regressor

A continuación, se observa el proceso de la implementación del modelo k-nearest neighbors:

```
from sklearn.neighbors import KNeighborsRegressor
```

```
start_time = time.time()

mod3 = KNeighborsRegressor(n_jobs=3, algorithm='auto')

parameters = {'n_neighbors':[3,5,7,9,11,13,15,17,19,21,23,25], 'weights':['uniform','distance'], 'p':[1,2], \
              'metric':['euclidean','manhattan','chebyshev','minkowski']}

mod3_1 = GridSearchCV(estimator=mod3, param_grid = parameters, cv=rkf, scoring='explained_variance', \
                      return_train_score=True, n_jobs=3,verbose=4)

mod3_1.fit(X_train_n, y_train)

elapsed_time = time.time() - start_time
print(elapsed_time)
```

```
Fitting 20 folds for each of 192 candidates, totalling 3840 fits
150.96168327331543
```

Ilustración 14: Parametrización Modelo K nearest neighbors (Elaboraciòn propia)

Los parámetros para evaluar serán el “n_neighbors” que es la cantidad de vecinos más cercanos, “weights” que hace referencia a los pesos de los datos, “p” valor de la métrica “Minkowski” y el parámetro “metric” para la selección del tipo de métrica de distancia.

Decision Tree Regressor

A continuación, se observa el proceso de la implementación del modelo Decision Tree Regressor.

```
from sklearn.tree import DecisionTreeRegressor
```

```
start_time = time.time()

mod4 = DecisionTreeRegressor(random_state=0)

parameters = {'min_samples_split':[9,10,11,12,13],
              'criterion':['squared_error','friedman_mse', 'absolute_error', 'poisson'],
             'splitter':['best','random'],
             'max_depth':[4,5,6,7,8,9],
             'min_samples_leaf':[2,3,4,5,6,7],
             'max_features':['auto','sqrt','log2']}

mod4_1 = GridSearchCV(estimator=mod4, param_grid = parameters, cv=rkf, scoring='explained_variance',return_train_score=True,
mod4_1.fit(X_train_n, y_train)

elapsed_time = time.time() - start_time
print(elapsed_time)
```

```
Fitting 20 folds for each of 13608 candidates, totalling 272160 fits
1356.277352809906
```

Ilustración 15: Parametrización Modelo Desicion Tree Regressor (Elaboración propia)

Los parámetros que se iteraron fueron ‘min_samples_split’ que es el número mínimo de muestras requeridas para dividir un nodo interno, ‘criterion’ que es el parámetro de la función para medir la calidad de una división, ‘splitter’ que es la estrategia utilizada para elegir la división en cada nodo, ‘max_depth’ que es la máxima profundidad del árbol, ‘min_samples_leaf’ que es el número mínimo de muestras requeridas para estar en un nodo de hoja y ‘max_features’ que es el número de características a tener en cuenta al buscar la mejor división. Parametrización realizada con base a lo propuesto por Mantovani et al. (2016).

Herramientas

Para el desarrollo del proyecto se utilizó Jupyter Notebook, como herramienta de editor de código.

RESULTADOS

Métricas

En esta sección, se describirán los resultados obtenidos de la ejecución de la metodología antes mencionada.

Baseline: Cómo se mencionó anteriormente los resultados obtenidos del modelo inicialmente evaluado, el cual fue una regresión lineal múltiple por mínimos cuadrados, no obtuvo un resultado adecuado, a continuación, se ilustra el resultado obtenido al predecir el conjunto de datos "X_test_n".

```
1  mod1_1.best_score_
```

```
-3172.2178139236144
```

Ilustración 16: Resultados de validación del modelo de regresión lineal por mínimos cuadrados (Elaboración propia)

Cómo se observa el anterior resultado es completamente diferente a lo esperado, ya que se esperaba un resultado entre 0 y 1. Se puede afirmar, la posibilidad de una subestimación. Por lo cual se toma la decisión de iterar con otros modelos para verificar si persiste o no este tipo de resultado.

1era iteración: Ahora se utilizó el modelo ElasticNet, un modelo de regresión lineal con regularizaciones o normalización, es decir, con parámetros adicionales que tratan de controlar el tamaño de los coeficientes para lograr un mejor ajuste, luego de ejecutar con este modelo las técnicas GridSearchCV y RepeatedKFold, se obtuvo la siguiente configuración óptima de parámetros:

```
1  mod2_1.best_params_
```

```
{'alpha': 0.4, 'l1_ratio': 0.9}
```

a. Con esta configuración se obtuvo los siguientes resultados en las métricas:
    i. La explicación de la varianza para el conjunto de validación fue de:

```
1  mod2_1.best_score_
```

```
0.8446060973202709
```

    ii. La explicación de la varianza para el conjunto de entrenamiento fue de:

```
results_mod2_1= pd.DataFrame(mod2_1.cv_results_)
results_mod2_1.where(results_mod2_1.params == mod2_1.best_params_).dropna()['mean_train_score'].T
```

```
89    0.863888
Name: mean_train_score, dtype: float64
```

iii. La explicación de la varianza para el conjunto de test fue de:

```
Yest = mod2_1.predict(X_test_n)

print(f"explained_variance_score = {explained_variance_score(y_test,Yest)}")
```

```
explained_variance_score = 0.877342554599692
```

En conclusión, para esta iteración con el modelo ElasticNet, se puede afirmar que se obtiene un mejor ajuste, representado en unas mejores métricas, por lo cual la inclusión de las regularizaciones, se expresa en resultados satisfactorios en la explicación de la varianza.

2da iteración: Para contrastar el resultado anterior obtenido, se ejecuta el modelo propuesto k nearest neighbors, luego de la evaluación de parámetros con GridSearchCV se obtienen la mejor configuración que es la siguiente:

```
mod3_1.best_params_
```

```
{'metric': 'manhattan', 'n_neighbors': 11, 'p': 1, 'weights': 'distance'}
```

a. Con esta configuración se obtuvo los siguientes resultados en las métricas:

i. La explicación de la varianza para el conjunto de validación fue de:

```
mod3_1.best_score_
```

```
0.8869175328580564
```

ii. La explicación de la varianza para el conjunto de entrenamiento fue de:

```
results_mod3_1= pd.DataFrame(mod3_1.cv_results_)
results_mod3_1.where(results_mod3_1.params == mod3_1.best_params_).dropna()['mean_train_score'].T
```

```
45    1.0
Name: mean_train_score, dtype: float64
```

iii. La explicación de la varianza para el conjunto de test fue de:

```
Yest = mod3_1.predict(X_test_n)

print(f"explained_variance_score = {explained_variance_score(y_test,Yest)}")
```

```
explained_variance_score = 0.8919916836253556
```

De los resultados anteriores de las métricas con el modelo k nearest neighbors se obtuvo valores muy similares al modelo anterior, ElasticNet, hay un aumento en la explicación de la

varianza en el conjunto de validación alrededor del 4% y en el conjunto de test un poco menos del 2%.

En la siguiente ilustración, se observa las características con mayor importancia, de acuerdo con el modelo de ElasticNet, se resalta el coeficiente los kilogramos y horas del emulsificador como una variable de gran relevancia.

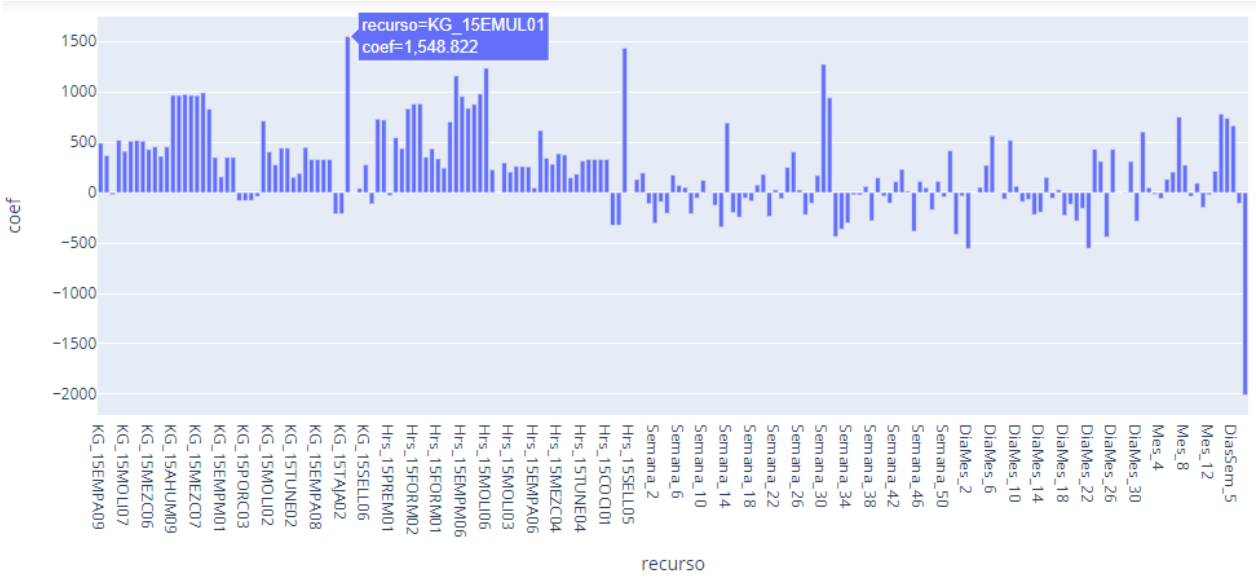


Ilustración 17: Importancia de características por el modelo ElasticNet (Elaboración propia)

3era iteración: Por último, se evaluó el modelo Decision Tree Regressor, luego de la evaluación de parámetros con GridSearchCV se obtienen la mejor configuración que es la siguiente:

```
1 mod4_1.best_params_
```

```
{'criterion': 'friedman_mse',
 'max_depth': 7,
 'max_features': 'auto',
 'min_samples_leaf': 5,
 'min_samples_split': 11,
 'splitter': 'random'}
```

a. Con esta configuración se obtuvo los siguientes resultados en las métricas:
   i. La explicación de la varianza para el conjunto de validación fue de:

```
1 mod4_1.best_score_
```

```
0.8876196505402433
```

   ii. La explicación de la varianza para el conjunto de entrenamiento fue de:

```
results_mod4_1= pd.DataFrame(mod4_1.cv_results_)
results_mod4_1.where(results_mod4_1.params == mod4_1.best_params_).dropna()['mean_train_score'].T
```

```
1655    0.919367
Name: mean_train_score, dtype: float64
```

iii. La explicación de la varianza para el conjunto de test fue de:

```
Yest = mod4_1.predict(X_test_n)

print(f"explained_variance_score = {explained_variance_score(y_test,Yest)}")
```

```
explained_variance_score = 0.8941635588883651
```

De los resultados anteriores de las métricas utilizando el modelo Decision Tree Regressor y comparándolo con los valores de las anteriores iteraciones, se puede afirmar que son muy similares a los obtenidos con el modelo k nearest neighbors y dado los buenos valores, se decide no continuar iterando a modelos más complejos ya que el ajuste parece inicialmente convencer con adecuada asertividad al grupo de interesados de la solución analítica.

Evaluación Cualitativa

Se puede afirmar de acuerdo con los resultados de las métricas de las iteraciones y modelos anteriores parece existir un poco de overfitting en el modelo K nearest neighbors por obtener un resultado en la métrica de entrenamiento de 1, en los modelos de ElasticNet y Decision Tree Regressor parece no existir casos de overfitting o underfitting.

La utilidad de los resultados, son de gran utilidad para compararlos entre ellos y de acuerdo a la complejidad y a la explicación de la proyección, se tome la decisión de seleccionar un modelo para la solución de la necesidad de la organización.

Se prevé una satisfacción satisfactoria por parte de los stakeholders y usuarios de la herramienta y los resultados, porque abarca en gran medida los requerimientos tanto en la evaluación del modelo desde la perspectiva del Machine Learning como también en la métrica de asertividad porcentual del negocio.

Consideraciones de producción

Las consideraciones de producción se mencionan a continuación:

El monitoreo real del desempeño del modelo, debe realizarse de acuerdo con un análisis comparativo mensual, entre el consumo real de energía y el proyectado para ese mismo periodo.

Se debe habilitar los mecanismos para que los analistas de capacidades que realizan la planeación táctica mensualmente actualicen constantemente el archivo con los datos a predecir.

CONCLUSIONES

En el presente proyecto, se aborda una necesidad importante para diversas áreas de la organización, como calidad, mantenimiento y ambiental. En trabajo en conjunto con ellos, se logra realizar un buen trabajo metodológico aplicando la analítica.

De acuerdo con la información suministrada por las áreas, se logra construir un conjunto de datos de valor para el objetivo y alcance del proyecto, obteniendo así un adecuado desempeño en la evaluación de los modelos predictivos.

De acuerdo con los resultados obtenidos de las métricas en cada uno de los dataset, se afirma que el modelo a seleccionar es el Decision Tree Regressor por sus buenos resultados en explicación de la varianza y por el correcto seguimiento metodológico a casos de uso de este tipo.

El diseño metodológico propuesto permitió realizar un proyecto analítico end-to-end de forma exitosa que satisface el requerimiento del negocio para la proyección de consumo y la generación de escenarios estratégicos.

El modelo final desplegado permitirá a los analistas de ambiental y capacidades como usuarios interactuar con la herramienta y modificar las variables predictoras de colocación en kilogramos y horas para evaluar alternativas tácticas.

Se concluye que existen oportunidades para continuar el desarrollo para los otros servicios públicos de la empresa, como agua y gas.

BIBLIOGRAFÍA

Rosenthal, G. & Rosenthal, J. (2011). Statistics and Data Interpretation for Social Work. Springer Publishing Company.

Schröer, C., Kruse, F., & Gómez, J. M. (2021). A Systematic Literature Review on Applying CRISP-DM Process Model. Procedia Computer Science, 181, 526–534. https://doi.org/10.1016/j.procs.2021.01.199

Refaeilzadeh, P., Tang, L., & Liu, H. (2016). Cross-Validation. Encyclopedia of Database Systems, 1–7. https://doi.org/10.1007/978-1-4899-7993-3_565-2

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2), 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x

Mantovani, R. G., Horvath, T., Cerri, R., Vanschoren, J., & de Carvalho, A. C. (2016). Hyper-Parameter Tuning of a Decision Tree Induction Algorithm. 2016 5th Brazilian Conference on Intelligent Systems (BRACIS). https://doi.org/10.1109/bracis.2016.018