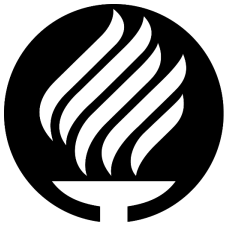


Tecnológico de Monterrey

CAMPUS MONTERREY

INTELIGENCIA ARTIFICIAL AVANZADA PARA LA CIENCIA DE DATOS II

TC3007C

ACTIVIDAD 7 - FEATURE SELECTION

Prof. Sebastián Ulises Adán Saldívarez

Daniela Jiménez Téllez - A01654798

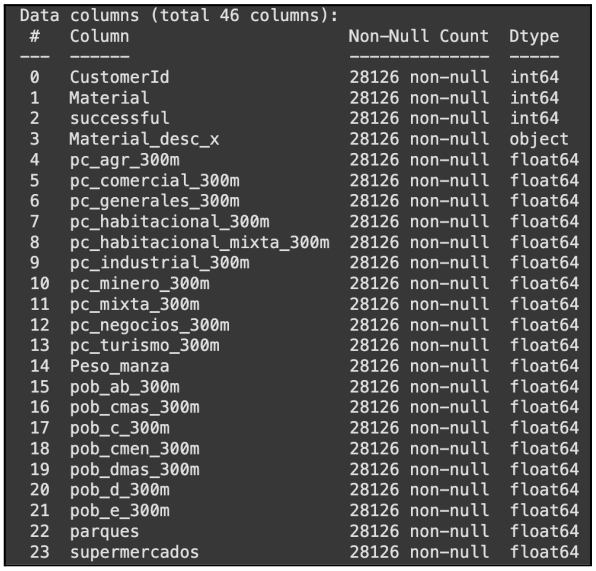
Lautaro Gabriel Coteja - A01571214

Andrés Villareal González - A00833915

Héctor Hibran Tapia Fernández - A01661114

I. Análisis y Justificación de Features Seleccionados (LAUTARO)

Para el proyecto, los datos se han dividido en 3 conjuntos, el final que fue el set original ya procesado *productos_exitosos.csv*, *df_entrenamiento.csv* y *df_prueba.csv*. A continuación se justificará el por que de las columnas seleccionadas para entrenamiento y prueba, considerando el objetivo del cliente: *predecir si un producto será exitoso basándose en sus características.*



```
Data columns (total 46 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   CustomerId                   28126 non-null  int64
 1   Material                     28126 non-null  int64
 2   successful                   28126 non-null  int64
 3   Material_desc_x              28126 non-null  object
 4   pc_agr_300m                  28126 non-null  float64
 5   pc_comercial_300m            28126 non-null  float64
 6   pc_generales_300m            28126 non-null  float64
 7   pc_habitacional_300m         28126 non-null  float64
 8   pc_habitacional_mixta_300m   28126 non-null  float64
 9   pc_industrial_300m           28126 non-null  float64
 10  pc_minero_300m               28126 non-null  float64
 11  pc_mixta_300m                28126 non-null  float64
 12  pc_negocios_300m             28126 non-null  float64
 13  pc_turismo_300m              28126 non-null  float64
 14  Peso_manza                   28126 non-null  float64
 15  pob_ab_300m                  28126 non-null  float64
 16  pob_cmas_300m                28126 non-null  float64
 17  pob_c_300m                   28126 non-null  float64
 18  pob_cmen_300m                28126 non-null  float64
 19  pob_dmas_300m                28126 non-null  float64
 20  pob_d_300m                   28126 non-null  float64
 21  pob_e_300m                   28126 non-null  float64
 22  parques                      28126 non-null  float64
 23  supermercados                28126 non-null  float64
```

```
 24  hospitales                   28126 non-null  float64
 25  preescolares                 28126 non-null  float64
 26  primarias                    28126 non-null  float64
 27  secundarias                  28126 non-null  float64
 28  preparatorias                28126 non-null  float64
 29  universidades                28126 non-null  float64
 30  gimnasios                    28126 non-null  float64
 31  gasto_promedio_300m          28126 non-null  float64
 32  gasto_total_300m             28126 non-null  float64
 33  ingreso_minimo_300m          28126 non-null  float64
 34  ingreso_promedio_300m        28126 non-null  float64
 35  ingreso_maximo_300m          28126 non-null  float64
 36  ingreso_remesas_300m         28126 non-null  float64
 37  ingreso_rentas_300m          28126 non-null  float64
 38  accesibilidad                28126 non-null  float64
 39  industry_customer_size       28126 non-null  int64
 40  sub_canal_comercial          28126 non-null  object
 41  Productos_Por_Empaque        28126 non-null  int64
 42  ProductType                  28126 non-null  object
 43  calmonth_x                   28126 non-null  object
 44  conteo_instalaciones         28126 non-null  float64
 45  categoria_instalaciones      28126 non-null  object
dtypes: float64(36), int64(5), object(5)
```

Columnas Seleccionadas y Justificación

a. Identificación del Cliente y Producto
   i. **CustomerId:** Identifica al cliente, permitiendo asociar características demográficas y de consumo.
   ii. **Material:** Identifica al producto, esencial para relacionar características específicas del producto con su éxito.
   iii. **Succesful:** Variable objetivo que indica si un producto fue exitoso.

b. Descripción del Producto
   i. **Material_desc_x:** Describe el producto (marca, sabor, presentación), clave para capturar atributos descriptivos que influyen en la preferencia del cliente.
   ii. **ProductType:** Proporciona la categoría del producto, importante para identificar tendencias en categorías específicas.

c. Características del Entorno
   i. **Pc_comercial_300m, pc_habitacional_300m, pc_generales_300m, pob, parques, supermercado, universidades, hospitales, primarias, secundarias, preescolares, preparatorias, gimnasios:** Proporcionan datos sobre la densidad de distintos tipos de establecimientos en un radio cercano, lo que puede afectar la demanda.
   ii. **Accesibilidad:** Indica que tan accesible es la ubicación, un factor que puede impactar el éxito de un producto.

d. Datos Socioeconómicos y Demográficos
   i. **Ingreso_remesas_300m, ingreso_rentas_300m:** Reflejan el nivel económico de la zona, influenciando el poder adquisitivo de los clientes.
   ii. **Industry_customer_size:** Tamaño del cliente industrial, una métrica para categorizar el perfil del consumidor.

e. Canales de distribución
    i. **Sub_canal_comercial:** Informa sobre el canal de venta, ayudando a identificar los más efectivos para ciertos tipos de productos.
f. Temporalidad
    i. **Calmonth_x:** Permite evaluar la influencia de la temporalidad en el éxito del producto.

Principales razones para la Selección:

El Socio Formador desea entender el éxito potencial de un producto con base en su descripción del entorno.

Las columnas seleccionadas reflejan:

- Atributos del Producto: Capturan las preferencias y tendencias.
- Entorno Socioeconómico: Ayudan a evaluar como las características de la ubicación afectan la demanda.
- Patrones de consumo: Identifican qué canales y períodos son más efectivos.

II. Métodos y Técnicas Utilizadas

Para la selección de features en el reto, aún no se ha aplicado pero, se ha decidido utilizar una combinación de métodos intrínsecos, específicamente Decision trees, en conjunto con Lasso Regularization, debido a su capacidad para identificar automáticamente características más relevantes al modelar datos con una gran capacidad de variables.

Método Intrínseco: Decision Trees

Los árboles de decisión son algoritmos de aprendizaje supervisado que no solo son útiles para clasificación y regresión, sino que también pueden ser utilizados como una técnica de selección de features. El modelo construye jerarquías de decisiones basadas en los valores de las características, asignando mayor importancia a aquellas que contribuyen más a la reducción de la impureza.

Proceso:

1. Entrenamiento Inicial: Se entrena un árbol de decisión utilizando todas las features disponibles.
2. Importancia de las Features: Se mide la importancia relativa de cada feature en el árbol, basado en la cantidad de información que aporta en las particiones.
3. Eliminación de Features Irrelevantes: Se descartan aquellas características con baja importancia para simplificar el modelo y mejorar su eficiencia.

Método Intrínseco Complementario: Lasso Regularization

La regularización Lasso agrega una penalización a la magnitud de los coeficientes de las variables en un modelo lineal, lo que fuerza a que algunos coeficientes se reduzcan exactamente a cero. Esto permite que Lasso actúe como un filtro, seleccionando automáticamente las features más relevantes.

Proceso:

1. Entrenamiento con Penalización: Se ajusta un modelo de regresión lineal con penalización L1.
2. Selección automática de Features: Las características con coeficientes reducidos a cero son descartadas, conservando sólo aquellas que tienen mayor relevancia predictiva.

Ventajas del Enfoque Combinado

- Interpretabilidad: Los árboles de decisión proporcionan una visualización más clara de las relaciones entre variables.
- Automatización: Lasso automática la eliminación de features redundantes, reduciendo la dimensionalidad.
- Robustez: La combinación de métodos ofrece una selección más robusta al considerar tanto las relaciones lineales como no lineales entre las variables.

Este enfoque garantiza un modelo más eficiente y menos propenso al sobreajuste, mejorando la capacidad predictiva y la generalización.

III. Criterios Adicionales

El proyecto tiene como objetivo identificar a los clientes ideales para comprar productos de lanzamiento, basándose en su comportamiento pasado y en el contexto que los rodea. En este caso, para poder llevar a cabo el reto se eligieron variables que reflejan el tipo de cliente con el que se está trabajando (por ejemplo, abarrotes, hogar con venta, estanquillos, etcétera), los establecimientos cercanos, como lo son gimnasios, escuelas, y hospitales, y el porcentaje de áreas residenciales, industriales, de negocios y turísticas en su entorno. Así, se puede estimar la demanda potencial en cada zona.

También se incluyeron características financieras, como los ingresos y gastos de cada cliente, junto con el nivel socioeconómico de la población cercana, para tener una idea del tipo de ventas que podrían generar. En cuanto al tipo de producto, se analizaron sus características principales, dividiéndolo en diferentes categorías y usando variables dummy para integrarlo fácilmente al modelo.

Toda esta selección de variables se hizo basada en un entendimiento que se adquirió del análisis exploratorio. Igualmente, se consideraron diferentes enfoques y este nos pareció el más apropiado, ya que tenía factores relevantes tanto del cliente, como del producto. Asimismo, dado a que las bases de datos proporcionadas originalmente eran muy extensas, se dejaron de lado variables que no brindaban tanta información, como la actividad móvil por hora o día, si una mujer u hombre era votante o no, velocidades promedio, entre otros, para simplificar el modelo y enfocarse en lo que realmente puede hacer la diferencia en el lanzamiento del producto.

IV. Resultados de Iteración

A continuación presentaremos los resultados de dos Redes Neuronales Secuenciales, *ambos modelos tienen las mismas características*, lo único que cambia es el set de features que se seleccionaron para alimentar a dichos modelos.

Para el primer modelo el set de features del cuál se alimentó la red, fue el set ya descrito arriba, donde solamente se droppearon las columnas de identificación ('successful', 'calmonth_x', 'Material_desc_x', 'CustomerId', 'Material', 'Productos_Por_Empaque'), lo que nos deja con las siguientes características:

1. 'pc_agr_300m'
2. 'pc_comercial_300m'
3. 'pc_generales_300m'
4. 'pc_habitacional_300m'
5. 'pc_habitacional_mixta_300m'
6. 'pc_industrial_300m'
7. 'pc_minero_300m'
8. 'pc_mixta_300m'
9. 'pc_negocios_300m'
10. 'pc_turismo_300m'
11. 'gasto_promedio_300m'
12. 'gasto_total_300m'
13. 'ingreso_minimo_300m'
14. 'ingreso_promedio_300m'
15. 'ingreso_maximo_300m'
16. 'ingreso_remesas_300m'
17. 'ingreso_rentas_300m'
18. sub_canal_comercial'
19. 'ProductType'
20. 'conteo_instalaciones'
21. 'categoria_instalaciones'
22. 'Peso_manza'
23. 'pob_ab_300m'
24. 'pob_cmas_300m'
25. 'pob_c_300m'
26. 'pob_cmen_300m'
27. 'pob_dmas_300m'
28. 'pob_d_300m'
29. 'pob_e_300m'
30. 'parques'
31. 'supermercados'
32. 'hospitales'
33. 'preescolares'
34. 'primarias'
35. 'secundarias'
36. 'preparatorias'
37. 'universidades'
38. 'gimnasios'
39. 'accesibilidad'
40. 'industry_customer_size'

Pasando de 45 a 40.

Para el segundo modelo, siguiendo la aplicación del modelo de Decision Trees para obtener la importancia de las features y poder quitar las que menos importancia tienen.

```
df = df_combinado.copy()
df_preprocessed = df.drop(columns=['CustomerId', 'Material', 'Material_desc_x','ProductType',
'calmonth_x', 'sub_canal_comercial', 'categoria_instalaciones'])
#df_preprocessed = pd.get_dummies(df_preprocessed, columns=[], drop_first=True)
X = df_preprocessed.drop(columns=['successful'])
y = df_preprocessed['successful']

from sklearn.tree import DecisionTreeClassifier
tree_clf = DecisionTreeClassifier(random_state=0)
tree_clf.fit(X, y)

feature_importances_transformed = pd.DataFrame({'feature': X.columns,'importance':
tree_clf.feature_importances_}).sort_values(by='importance', ascending=False)
```

Quitamos un total de 3 más características:

1. 'pc_negocios_300m'
2. 'pc_minero_300m'
3. 'pc_agr_300m'

Lo cual nos deja con 37 features para el Modelo 2.

Descripción de la Red Neuronal Secuencial

Nuestro modelo es un modelo de clasificación binaria diseñado para predecir si una condición (columna `successful`) se cumplirá o no, basándose en varias características de entrada.

La arquitectura de la red consta de una capa de entrada, dos capas ocultas con 64, 32, y 16 neuronas respectivamente, y una capa de salida con una neurona que utiliza la función de activación sigmoide para producir una probabilidad entre 0 y 1.

Se aplican técnicas de normalización por lotes (Batch Normalization) y regularización mediante Dropout en cada capa, ayudando a estabilizar el aprendizaje y prevenir el sobreajuste.

Entrenada con datos de clientes, productos y otros factores categóricos, la red busca reconocer patrones complejos en estos datos para hacer predicciones precisas sobre el éxito o fracaso de las condiciones definidas en el conjunto de datos.

Resultados de los Modelos

Red Neuronal sin cambios

Metric	Value
Predicciones Correctas	416 / 500
Model Accuracy on Test Data	0.80

Classification Report

Class	Precision	Recall	F1-Score	Support
Not Successful	0.82	0.95	0.88	5078
Successful	0.67	0.31	0.42	1559
Accuracy			0.80	6637
Macro Avg	0.74	0.63	0.65	6637
Weighted Avg	0.78	0.80	0.77	6637

Confusion Matrix

	Predicted Not Successful	Predicted Successful
Actual Not Successful	4834	244
Actual Successful	1073	486

Red Neuronal con features seleccionados con Árbol de Decisión

Metric	Value
Predicciones Correctas	415 / 500
Model Accuracy on Test Data	0.80

Classification Report

Class	Precision	Recall	F1-Score	Support
Not Successful	0.82	0.95	0.88	5078
Successful	0.67	0.33	0.44	1559
Accuracy			0.80	6637
Macro Avg	0.75	0.64	0.66	6637
Weighted Avg	0.79	0.80	0.78	6637

Confusion Matrix

	Predicted Not Successful	Predicted Successful
Actual Not Successful	4826	252
Actual Successful	1044	515

Como puede notarse en las predicciones de los modelos, realmente no hay una diferencia, aunque cuando se usa árboles de decisión para la selección de características se muestra un pequeño pero positivo impacto en la predicción de "Successful". Aquí parece haber permitido al modelo captar más patrones relevantes en dicha clase, mejorando ligeramente el recall y el F1-score sin comprometer el rendimiento general o el de la clase mayoritaria.

En general el impacto es relativamente leve, lo que nos dice que el modelo original ya estaba bien optimizado o lo que consideramos en este caso es que la selección de características adicionales **podría explorar otros métodos para mejorar más drásticamente el rendimiento en la clase minoritaria.**

Link del Notebook:
https://colab.research.google.com/drive/1xtxekhQTsLS2ZUFu-yo8M8Zha0sVeNSx?usp=sharing