UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

MODELO DE CLASIFICACIÓN DE COMPRAS PÚBLICAS EN ESTÁNDAR UNCCS

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN TECNOLOGÍAS DE LA INFORMACIÓN

HUGO PATRICIO GALLARDO RODRÍGUEZ

PROFESOR GUÍA:
BENJAMÍN BUSTOS CÁRDENAS

MIEMBROS DE LA COMISIÓN:
ANDRÉS ABELIUK KIMELMAN
IVÁN SIPIRÁN MENDOZA
RODRIGO ALFARO ARANCIBIA

Este trabajo ha sido parcialmente financiado por Becas ANID

SANTIAGO DE CHILE
2023

Resumen

La transparencia en las compras públicas en Chile permite a la ciudadanía conocer cómo se gastan los recursos públicos, cuál es su nivel de eficiencia y permitir su rendición de cuentas hacia la sociedad. Para fomentar lo anterior, el Observatorio del Gasto Fiscal (OGF) realiza distintos análisis sobre los datos públicos provenientes de la Dirección de Compras y Contratación Pública del estado (DCCP). Sin embargo, cuentan con diversos problemas de calidad, reduciendo la capacidad de un correcto estudio sobre éstos. En específico, el trabajo de esta tesis aborda el problema de la concordancia que existe entre la asignación del código de producto, y su descripción por parte de los compradores en los datos provenientes de la DCCP.

A través de un análisis inicial de 1.200 productos de Tecnologías de la Información provenientes de municipalidades, se desprende que un 64 % de éstos no tienen una correcta concordancia entre el etiquetado y su descripción de texto. Teniendo en cuenta la gran cantidad de productos transados mensualmente, resulta necesario el generar una solución automática para la clasificación de éstos.

El marco teórico nos presenta como solución la utilización de modelos de clasificación, tanto multiclase como binario. Para poder ser aplicados correctamente, se generó un proceso que aplica técnicas de preprocesamiento de texto. Luego se buscó el mejor modelo entre Máquina de Soporte Vectorial (SVM), Árboles Aleatorios (RF), Clasificador Bayesiano y K-vecinos más cercanos.

Inicialmente se aplicó un modelo SVM multiclase para identificar la clase de un producto basado en su descripción. Sin embargo, no fue capaz de generalizar correctamente al llegando solo a un 49 % de promedio F1 macro. Agregando un post procesamiento incluyendo una clase desconocida, tampoco se llegó a un resultado satisfactorio.

Dados los problemas del modelo anterior, se aplicó un modelo RF binario identificando la concordancia entre la etiqueta y descripción del producto. Este presentó un promedio F1 macro de 85 % dentro de los datos de evaluación, concluyendo que este es un modelo viable para utilizar en este problema.

El modelo binario llevado al servicio web gratuito de HFS tuvo un resultado satisfactorio, permitiendo correctamente a usuarios externos la utilización del modelo sobre nuevos conjuntos de datos. En específico para OGF, el prototipo les permitirá generar modelos de clasificación para futuros análisis relacionados a las CCPP, enfocado en distintos sectores.

A todos aquellos que han sido una parte integral de mi camino académico y personal.

Agradecimientos

A mi familia, quienes me han apoyado y fomentado a crecer a lo largo de todas mis etapas académicas, incluso a la distancia.

A mi esposa Noemí, que gracias a su apoyo y amor infinito, siempre me animó a superar los distintos desafíos encontrados en este camino.

A mi profesor guía Guillermo Bustos, por su orientación experta y constante apoyo a lo largo de esta tesis. Siempre estuvo dispuesto a brindar su tiempo y experiencia para ayudarme en lograr avances en las distintas etapas.

A mi amigo Alfonso Tobar, que fue un pilar importante en el desarrollo de esta tesis. Su apoyo en dificultades técnicas, y orientación experta fueron indispensables para lograr los objetivos propuestos.

A José Mora del Observatorio del Gasto Fiscal, y Esteban Olivares de ChileCompra por su ayuda e interés en la motivación de esta tesis.

Al programa de Becas ANID por el financiamiento de este programa de Magíster, y al Ministerio de Desarrollo Social por su ayuda en la obtención de este financiamiento.

Al programa de Magíster TI de la Facultad de Ciencias Físicas y Matemáticas, y sus profesores, que a lo largo de los distintos ramos me brindaron una comprensión sólida de los fundamentos necesarios para llevar a cabo esta tesis.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

La siguiente tesis se enmarca en una colaboración con la Organización no Gubernamental "Observatorio del Gasto Fiscal", en adelante denominado OGF. Esta institución tiene dentro de sus lineamientos, el recopilar, analizar, y desarrollar diversos productos asociados a las compras públicas de Chile (CCPP), que les permitan a la sociedad civil y ciudadanía en general, conocer cómo se gastan los recursos públicos, cuál es su nivel de eficiencia y su grado de transparencia en la rendición de cuentas hacia la sociedad.

Gran parte de la información que analiza OGF proviene de datos de compras públicas suministrada directamente desde las bases de datos transaccionales de la Dirección de Compras y Contrataciones Públicas (DCCP). Si bien esta plataforma entrega la completitud de las licitaciones y órdenes de compra de las transacciones realizadas por las instituciones del estado, a nivel de diseño, es aún muy dependiente de la información manual que entregan sus compradores y proveedores. Lo anterior genera problemas en la calidad de los datos de entrada de los productos y servicios adquiridos, impactando directamente a todo análisis que se realice asociados a las CCPP, y, por tanto, a la transparencia de estos procesos, y su rendición de cuentas hacia la sociedad.

En específico, uno de los mayores problemas de calidad de datos se encuentra en el incorrecto etiquetado de los productos o servicios a transar. Actualmente, el etiquetado de éstos es obligatorio dentro de la plataforma de la DCCP, y se debe realizar mediante la codificación UNCCS (Sistema de Codificación Común de las Naciones Unidas), propuesta por la Organización de las Naciones Unidas (ONU). Esto considera que tanto el comprador como vendedor al inicio de la transacción deban asignar a todo elemento transado el código correspondiente a su categoría definida en el listado de productos y servicios UNCCS.

Para enfrentar parcialmente lo anterior, OGF tiene como plan el de exponer a la comunidad a través de una visualización en su portal, los problemas de etiquetado productos y servicios en el estándar UNCCS, teniendo como principal foco los adquiridos por municipalidades, que a juicio experto de OGF, son las instituciones que presentan los mayores problemas en el etiquetado de datos. En términos de magnitud, las municipalidades junto a las corporaciones municipales corresponden a 359 instituciones, cuyos gastos en los productos y servicios anteriormente mencionados durante enero a noviembre del año 2021, ascienden a $128,745,463,192.

Dada que la variedad de tipos de productos que se transan es demasiado amplia para ser abarcada en su totalidad con bajos recursos, se acordó con OGF que el segmento que se estudiará de productos será acotado al de Tecnologías de la Información, y si la solución obtenida es factible, extenderla en un futuro a otros segmentos. Lo anterior es escogido por tres motivos: es un segmento cuyo etiquetado es posible evaluar correctamente por el tesista, existen modalidades de compra en que los datos históricos están correctamente etiquetados, y que estos representan un gran porcentaje de los gastos incurridos por las municipalidades.

1.1. Evaluación inicial del problema

Dado el contexto anterior, para poder exhibir en un portal los casos de mala calidad de datos por mal etiquetado de productos, es necesario discernir correctamente cuál sería la etiqueta correcta para cada producto analizado. Sin embargo, si bien tenemos el juicio experto de OGF sobre la mala calidad de éstos, al momento no contamos con una línea base que nos indique aproximadamente la precisión del etiquetado de los compradores que se buscaría mejorar, ni cuánto sería el costo en horas hombres para el etiquetado si fuese realizado de manera manual.

Con el fin de comprobar el juicio experto de OGF y tener una línea base, se realiza el ejercicio de revisar manualmente el etiquetado de una muestra con aproximadamente 1.200 productos provenientes de licitaciones ingresadas manualmente (Categorizadas en el sistema interno como SE) de municipalidades en 2 meses distintos, clasificados dentro de la codificación UNCSS: "Tecnologías de la información, telecomunicaciones y radiodifusión". Para determinar si un producto está correctamente etiquetado, se evalúa que el código asignado por el comprador tenga concordancia con el texto de la descripción del producto tanto por el comprador como el proveedor. Los datos utilizados fueron obtenidos desde el portal de datos abiertos de la DCCP, que permite la descarga de sus transacciones a nivel mensual. Para extraer muestras que no tengan alguna relación entre ellas, tanto por su estacionalidad o por los compradores participantes, se eligen 1.200 productos aleatorios de los meses de marzo 2020 y junio 2021.

Tras realizar el ejercicio de etiquetado manual de los productos con las características anteriormente mencionadas, se llega a que, del total, solo 772 se consideraron como correctamente etiquetados, es decir, el 64 %. Además, tras medir el tiempo en la inspección manual en base al número de productos, el tiempo aproximado que toma evaluar 50 transacciones es de una hora. Si consideramos que se tienen datos de transacciones de compras públicas históricas aproximadamente desde el año 2007, y que, además, los productos y servicios de municipalidades relacionadas a tecnologías de la información son de alrededor de 2.500 al mes, esto llevado a horas hombres serían aproximadamente más de 50 horas por mes. Este subsegmento es solo una pequeña fracción del total de productos transados mensualmente, que pueden llegar a ser de 140 mil.

Basados en los resultados de este ejercicio, es posible notar que existe un margen considerable para la mejora en el etiquetado en las actuales transacciones, y que, además, la inspección manual resulta costosa en términos de horas hombres. Dado lo anterior, se ve la necesidad de desarrollar una solución de etiquetado automático, que entregue resultados

en un menor tiempo, y que además sea escalable a un aumento de datos en caso de que se desee ampliar el segmento en una segunda versión. A esto se agrega, que para motivar su utilización, esta debe ser lo suficientemente sencilla en su uso por gente no especializada en el área de programación.

A priori se propone que la solución para este problema es el de utilizar técnicas automáticas de clasificación de etiquetas, sin embargo, no se cuenta con datos previos para poder comparar y mejorar la precisión de un proceso previamente desarrollado. Por lo tanto, se opta por la exploración y comparación de un set de técnicas que resulten factibles para reducir los tiempos de inspección sobre productos etiquetados. El resultado de lo anterior permitiría generar una línea base para afrontar este problema.

Para generar esta línea base, en esta tesis se propone probar dos alternativas de modelos de clasificación sobre estos datos. El primer modelo es un modelo multiclase, que dada la descripción de un producto predice cuál es su etiqueta correcta. Esto permite compararla con la etiqueta declarada y ver si hay una discrepancia. El segundo modelo es un modelo binario, que dada la descripción de un producto y su etiqueta declarada predice si dicha etiqueta es correcta o no. Se evaluarán ambos tipos de modelo, para estudiar cuál de ellos es el que obtiene mejor precisión en la identificación de etiquetados incorrectos.

1.2. Objetivos

El objetivo general es el desarrollo de un servicio web con un sistema de clasificación automática de productos y servicios de compras públicas en formato UNCCS, para productos relacionados a tecnologías de la información provenientes de municipios.

Para lograr el objetivo general, se plantean los siguientes objetivos específicos:

1. Creación de un set de datos con etiquetas correctas, que funcione como un ground truth para evaluar distintos modelos de clasificación.
2. Selección y desarrollo de modelo multiclase de clasificación de productos.
3. Selección y desarrollo de modelo binario de clasificación de productos para identificar casos erróneos.
4. Evaluación de los modelos de clasificación desarrollados, y selección del modelo con mejor rendimiento.
5. Desarrollo de un servicio web automático que envíe los datos del sistema a un sistema interno de OGF.

Capítulo 2

Marco teórico

2.1. Clasificación supervisada

El problema de clasificación [1] en el contexto de esta tesis es el de aprender la estructura de un conjunto de datos, siendo una de sus características que ya estén clasificados en grupos, que comúnmente se les llama clases o etiquetas. La mayor parte de los métodos de clasificación se basan en la construcción de un modelo o clasificador en la fase de entrenamiento, a través de la aplicación de un algoritmo de clasificación. Luego, este modelo se usa para asignar clases predefinidas en instancias de evaluación o test.

Por lo tanto, para que este tipo de modelos funcione, se requiere principalmente de 2 sets de datos correctamente etiquetados: uno que sirva como entrenamiento para el aprendizaje del modelo denominado set de entrenamiento, y un set de datos que no se incluya en el modelo para evaluar la precisión de este. Dado lo anterior, podemos decir que este tipo de técnicas es "supervisada", al requerir de un set de datos de ejemplo para poder funcionar correctamente.

Los modelos de clasificación supervisada comúnmente consisten en 2 fases:

1. Entrenamiento: Se construye un modelo basado en los datos de entrenamiento.
2. Testing: Se aplica el modelo en los datos de prueba. Comúnmente, las salidas de esta fase pueden ser la predicción de la clase, o un puntaje numérico determinando la cercanía a una clase en particular.

Entre los modelos de clasificación comúnmente usados se encuentran las máquinas de soporte vectorial, árboles de decisión, K-vecinos más cercanos, y clasificador bayesiano que serán descritos a continuación en este marco teórico.

2.1.1. Máquinas de Soporte Vectorial

Las Máquinas de Soporte Vectorial (Support Vector Machines SVMs) es un algoritmo supervisado que se utiliza para problemas de clasificación y regresión [15, 20, 8]. Considera los datos de entrada como vectores n-dimensionales dentro de un conjunto mayor que se denominará "Espacio". Cada uno de estos datos estará representado en una posición de este espacio en la cual se le asignará una posible clase. El algoritmo SVM construye un modelo que permite predecir a qué clase pertenece un punto nuevo con clase desconocida.

La característica fundamental del modelo SVM es que en el entrenamiento busca separar las diferentes clases en el espacio con la mayor distancia posible, para que los datos de prueba puedan ser correctamente clasificados dependiendo de la función de proximidad con respecto a cada conjunto. De esta manera, los elementos que estén clasificados en una categoría en particular buscarán estar en un extremo del espacio, y los de otra categoría en otro, tal como ejemplifica la Figura 2.1.

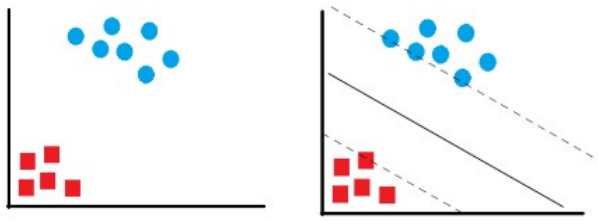


Figura 2.1: Ejemplo gráfico de SVM (Parra [15]: Figura 6.8 de Capítulo 6.7).

Cabe destacar, que si el límite entre las dos clases es lineal, el clasificador SVM es un método natural para la clasificación en el caso de dos clases, como se muestra en la Figura 2.1. Sin embargo, en la práctica los límites pueden ser no lineales entre las clases. Para solucionar parcialmente este problema, el SVM permite la utilización de funciones de núcleo (kernel) para proyectar la información en un espacio de características de dimensión mayor.

Uno de los mayores inconvenientes en los modelos de SVM es su alto costo computacional [11]. En general, un alto número de atributos pueden conllevar a altos tiempos de procesamiento. Este inconveniente se ve contrarrestado por el hecho de que, una vez formado un modelo SVM, pequeños ajustes en los datos de entrenamiento no alterarán significativamente los coeficientes del modelo siempre que los vectores de soporte permanezcan constantes. Además, los SVM se han convertido en uno de los algoritmos de aprendizaje automático más adaptables debido, en parte, a su resistencia al sobreajuste, y su alta flexibilidad en diferentes tipos de aplicaciones.

2.1.2. Algoritmo K-vecinos más cercanos

El algoritmo de K-vecinos más cercanos (KNN) [15, 5] es un algoritmo de clasificación supervisada no paramétrico que permite estimar la probabilidad a posteriori que un elemento x pertenezca a una clase Cj, a partir de los datos de entrenamiento.

A modo de ejemplo, en la Figura 2.2 se muestran doce elementos clasificados en dos clases distintas, azul y rojo. A este set de datos, agregamos el punto x que está cercano a ambas clases, y que deseamos estimar a qué clase pertenece. Para esto, en este caso particular se revisan los tres vecinos más cercanos ($k = 3$), siendo en este caso, dos vecinos rojos y uno azul, por lo tanto, este nuevo elemento sería clasificado como rojo. Sin embargo, si hubiésemos utilizado un $k = 1$, hubiese sido clasificado como azul, puesto que el elemento azul es el más cercano. Cabe destacar, que el cálculo de la distancia entre un punto y sus vecinos generalmente es a través de la distancia euclidiana, sin embargo, es posible aplicar otras.

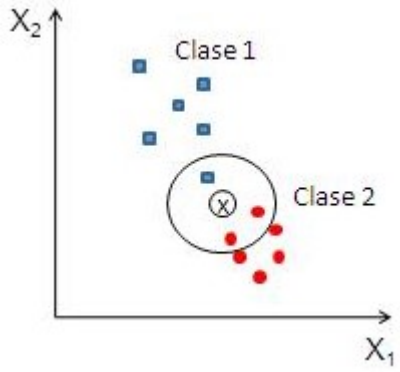


Figura 2.2: Ejemplo gráfico de KNN (Parra [15]: Figura 6.6 de capítulo 6.4).

Intuitivamente, utilizando este algoritmo se podría suponer que utilizando todos los atributos el vecino más cercano siempre nos dará la mejor precisión, sin embargo, se debe tener en cuenta la posibilidad que exista una mayoría de atributos irrelevantes que tengan peso dentro de la clasificación por sobre los relevantes. Esto podría solucionarse estimando la mejor elección de los vecinos más cercanos (k), seleccionando uno que permita atenuar el efecto de los atributos irrelevantes. El cálculo de un buen valor de k se puede realizar a través de un proceso de optimización.

Una de las mejores características es su capacidad para aplicarse a distintos tipos de datos, siempre que se disponga de una función de distancia para calcular las separaciones entre objetos [1]. Con frecuencia, el problema de clasificación es la principal consideración a la hora de crear funciones de distancia.

Por otro lado, la eficacia de la técnica en el proceso de clasificación es un problema importante con el uso de este tipo de clasificadores. Esto se debe a la posibilidad de que el tiempo de ejecución para recuperar los K-vecinos más cercanos esté linealmente correlacionado con el tamaño del set de datos. Este problema genera como consecuencia, que esta técnica pueda presentar problemas de escalabilidad en un gran aumento del volumen de datos y su dimensionalidad.

2.1.3. Árboles de decisión

El algoritmo de árboles de decisión (Decision Tree, DT) [15, 9, 11] es un algoritmo de clasificación supervisada no paramétrico, que se basa en la partición sucesiva de los datos. Son particularmente útiles al encontrarse una gran cantidad de datos, teniendo como gran ventaja la interpretabilidad de las estimaciones del modelo desde un punto de vista analítico.

En específico, los DT se basan en un proceso de división secuencial, iterativa y descendente que, a partir de una variable dependiente, forma subgrupos mediante la combinatoria de variables dependientes restantes. Estos se componen de tres elementos:

- Nodos: Variables dependientes que entran al modelo. Cabe destacar que se denomina nodo raíz al de mayor relevancia dentro del proceso
- Ramas: Indica los posibles valores de entrada de las variables dependientes
- Hojas: Potenciales valores finales, o de salida del modelo

Por lo general, los DT se comportan de manera consistente a los datos de entrada, sin embargo, se debe tener en cuenta que, si los datos presentan problemas en su lógica, el modelo aprenderá de estas lógicas afectando en la precisión de su estimación. A este efecto sobre el modelo se le llama sobreajuste. Una forma de reducir la posibilidad de llegar al sobreajuste es el de utilizar las técnicas de "podado"(Pruning), tales como limitar el crecimiento del árbol, o la cantidad de nodos a considerar.

Otra manera de reducir el sobreajuste es la utilización del método de árboles aleatorios (Random Forest, RF) [2], el cual ejecuta de forma paralela múltiples árboles de decisión cambiando levemente el subconjunto de datos. Luego, se ensamblan los resultados de la clasificación de cada uno de estos y se elige la clase que tenga la mayoría.

En la Figura 2.3 se ilustra a modo de diagrama de flujo la secuencia que tomaría la utilización de RF. Se cuenta con los datos de entrenamiento, dentro de los cuales se generan distintas muestras aleatorias distintas $(S-1, S-2, \ldots, S-n)$. En cada una de estas muestras se aplica un árbol de decisión, los cuales cada uno entregará una clase. Una vez se tienen todas las clases de los árboles se elige el que cuente con la mayoría de estos.

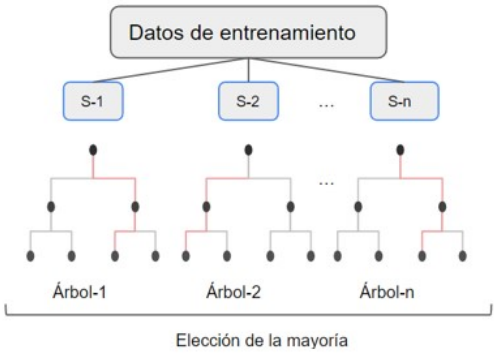


Figura 2.3: Ejemplo gráfico de RF.

Una de las mayores ventajas de los árboles de decisión, es que no es necesaria la creación de variables ficticias (Dummy) en la presencia de predictores categóricos, por lo tanto requiere menor esfuerzo en la preparación de los datos. Además, el rendimiento del árbol no se ve afectado por interacciones no lineales entre parámetros.

2.1.4. Clasificador Bayesiano

El clasificador bayesiano [15, 22] es un algoritmo de clasificación supervisada que, a diferencia de los anteriormente vistos, tiene sus raíces en la teoría estadística y de probabilidad. En particular, está basado en el teorema de Bayes, el que intuitivamente nos permite calcular hasta cierto punto cuánto cambia una variable dada la probabilidad condicionada de esta con otra. A este teorema también se le conoce como de probabilidad condicionada, cuya fórmula teniendo X e Y como variables independientes entre sí se muestra en la Ecuación 2.1 .

$$P(Y|X) = \frac{P(Y) * P(X|Y)}{P(X)} \tag{2.1}$$

Para el caso de un modelo de clasificación bayesiano [11, 18], si consideramos X como el set de atributos de un objeto y Y como la clase del objeto, la fórmula describe lo siguiente:

- $P(X)$: Probabilidad a priori que la evidencia es verdadera.
- $P(Y)$: La probabilidad (A priori) de la clase, que es la probabilidad de ocurrencia del evento antes que nuevos datos sean introducidos. Esta puede ser calculada mediante el número de ocurrencias de una clase sobre el total de datos de entrenamiento.
- $P(Y/X)$: Es la probabilidad condicionada de la clase Y dados los datos históricos de los atributos de X. Esta es la probabilidad a posterior que deseamos calcular una vez se introduzcan datos nuevos.
- $P(X/Y)$: Es la probabilidad condicionada de la existencia de atributos X dada una clase Y. Esta probabilidad puede ser calculada a través de los datos de entrenamiento.

Es uno de los algoritmos de clasificación más utilizados por lo simple y rápido que resulta implementar en cualquier lenguaje de programación. Por lo general, se usa como cota inferior al comparar con otros modelos de clasificación más complejos y con mayores tiempos de ejecución. A diferencia del SVM o DT, el clasificador bayesiano no se ve mayormente afectado por los valores nulos en los sets de datos, omitiéndolos de la clase condicional del total de valores.

Este algoritmo posee ciertas limitaciones que se indicarán a continuación:

- Fallos con un set de entrenamiento incompletos: En caso de que en el set de entrenamiento no existan atributos que después sí estén representados en el set de prueba, el clasificador le asignará una probabilidad de cero, seleccionando una predicción aleatoria.

- Atributos continuos: El algoritmo solo permite variables discretas, no continuas. Para solucionar esto, se debe discretizar si es posible las variables continuas.
- Independencia de los atributos: El algoritmo tiene como prerrequisito fundamental que todos los atributos sean independientes entre sí. Esto se puede solucionar a través de un preprocesamiento de los datos, identificando y removiendo los atributos que tengan una fuerte correlación.

2.1.5. Tabla comparativa de modelos

A continuación, en la Tabla 2.1 se muestra una comparación de las ventajas y desventajas de cada modelo de clasificación indicado en este capítulo.

	Ventajas	**Desventajas**
Máquinas de Soporte Vectorial	• Pequeños cambios en los datos no requieren un alto costo en el remodelamiento debido a su robustez. • "Resistencia al sobreajuste", debido a que el límite de las clases dentro de un dataset puede ser descrito usualmente con solo algunos vectores de soporte.	• Alto costo computacional.
K-vecinos más cercanos	• Modelo altamente robusto cuando el set de datos para test presenta todos sus valores en los atributos. • Fácil implementación y extensión a modelos multiclases. • Puede ser ocupado en cualquier tipo de datos, siempre y cuando la función de distancia pueda ser aplicada para poder cuantificar la distancia entre los objectos.	• Poco eficiente cuando el volumen o dimensionalidad de los datos es excesivamente grande, debido a que el tiempo que necesita en el procesamiento es lineal al tamaño del set de datos.
Árboles de decisión	• Fácil interpretabilidad del proceso de clasificación • Requiere menor esfuerzo en el preprocesamiento de los datos en caso de variables categóricas. • No requiere asumir linealidad en los datos.	• Un inadecuado proceso de podado (pruning) puede llevar a sobreajuste en los resultados.
Clasificador Bayesiano	• Sirve como benchmarking al usarse como cota inferior para comparar con otros modelos. • Algoritmo robusto en caso de outliers o valores faltantes en el set de datos. • Implementación sencilla en comparación a otros modelos más complejos	• Baja precisión si el supuesto de independencia es violado o es impreciso. • Limitación del algoritmo al trabajar solo con atributos discretos.

Tabla 2.1: Tabla comparativa de modelos revisados en el marco teórico.

2.2. Métricas de evaluación para modelos de clasificación

Una vez se tiene entrenado un modelo de clasificación con los datos de entrenamiento, es necesario evaluar su efectividad mediante los datos de prueba [1, 6]. Para esto, existen distintas métricas que nos permiten evaluar su desempeño relativo. Esto nos permite comparar de igual manera distintos modelos para elegir el que tenga mejor desempeño.

Matriz de confusión

Considera la comparación de las clases reales del set de datos de prueba con las clases estimadas del modelo sobre este mismo set de datos [11]. Si usamos como ejemplo un modelo con clases binarias (cero y uno), se ordenaría con la siguiente lógica:

		Clase observada	**Clase observada**
		1	0
Clase estimada	1	Verdadero Positivo	Falso Positivo
Clase estimada	0	Falso Negativo	Verdadero Negativo

Tabla 2.2: Ejemplo de matriz de confusión para modelo binario.

- Verdadero Positivo (VP): Si la clase estimada es 1, y la clase observada es 1.
- Falso Positivo (FP): Si la clase estimada es 1, y la clase observada es 0.
- Falso Negativo (FN): Si la clase estimada es 0, y la clase observada es 1.
- Verdadero Negativo (VN): Si la clase estimada es 0, y la clase observada es 0.

Métricas derivadas de la matriz de confusión

A partir de la matriz de confusión de la Tabla 2.2 se pueden obtener métricas para la evaluación numérica de un modelo [11]:

- Sensibilidad: También conocida como "Exhaustividad", es el ratio que indica si el modelo selecciona los casos que deben ser seleccionados. Se calcula usando la Ecuación 2.2:

$$VP/(VP + FN) \tag{2.2}$$

- Especificidad: Ratio que indica si el modelo rechaza los casos que deben ser rechazados. Se calcula usando la Ecuación 2.3:

$$VN/(VN + FP) \tag{2.3}$$

- Precisión: Proporción de los casos positivos verdaderos estimados sobre todo lo que el modelo identificó como positivo. Se calcula usando la Ecuación 2.4:

$$VP/(VP + FP) \tag{2.4}$$

- Exactitud (Accuracy): Proporción de todos los casos estimados correctamente estimados, por sobre todos los casos. Se calcula usando la Ecuación 2.5:

$$(VP + VF)/(VP + VF + FP + FN) \tag{2.5}$$

Cabe destacar que comúnmente se utiliza la exactitud como métrica inicial para medir los modelos de clasificación, sin embargo, en casos con datos desbalanceados puede llevar a conclusiones equivocadas [1]. En tales casos, la métrica F1-score resulta más apropiada, dado que mide de igual manera la precisión y la exhaustividad:

- F1-score: Agrupa en una sola métrica la precisión y exhaustividad. Se calcula usando la Ecuación 2.6:

$$(2 * Precision * Exhaustividad)/(Precision + Exhaustividad) \tag{2.6}$$

Para los casos de modelos multiclase, las métricas anteriores se pueden agregar de dos formas [21]:

- Promedio macro (Macro Average): Promedio simple de la métrica a elección en cada una de las clases. Esta métrica se utiliza cuando cada una de las clases tiene exactamente la misma prioridad dentro del modelo. Se calcula usando la Ecuación 2.7, considerando C como el valor de la métrica a elección de la clase i, y n como el número de clases.

$$\frac{1}{n}\sum_{\mathrm{i}=1}^{n} C_{\mathrm{i}} \tag{2.7}$$

- Promedio ponderado (Weighted Average): Promedio ponderado de cada clase por el soporte de cada una de éstas. Es decir, las clases con mayor porcentaje de datos sobre el total tienen un mayor peso dentro del cálculo de la métrica. Lo anterior, se utiliza cuando la métrica no es importante en las clases con menos casos. Se calcula usando la Ecuación 2.8, considerando C como el valor de la métrica a elección, W como el peso de la clase con respecto al total de datos, y n como el número de clases.

$$\sum_{\mathrm{i}=1}^{n} W_{\mathrm{i}} * C_{\mathrm{i}} \tag{2.8}$$

La vectorización basada en frecuencia de palabras consiste en transformar un texto en un vector, no una palabra en vector.

2.3. Preprocesamiento de texto

A diferencia de los datos cuantitativos o cualitativos (nominales u ordinales), el trabajar con documentos de "texto libre" supone un mayor reto para ser utilizados en modelos, dada su potencial dispersión y multidimensionalidad [1, 7]. Dado lo anterior, preprocesamientos y algoritmos especializados deben ser considerados para ser aplicados en el texto libre para servir correctamente como insumos para un modelo de clasificación. Algunos de éstos se indican a continuación:

- Tokenización: Consiste en convertir cada palabra o término del documento en un atributo independiente dentro de una lista. Por ejemplo, la frase "Esto es un ejemplo de texto." quedaría de la siguiente forma considerando palabras entre comillas y separadas por coma ["Esto", "es", "un", "ejemplo", "de","texto", "."].
- Remover stop words: Consiste en eliminar del texto palabras que son comunes dentro del idioma, y que no aportan mayor valor semántico. Ejemplo de esto son conectores, artículos, pronombres, etc.
- Remover puntuación y modificar caracteres especiales: Consiste en eliminar del texto caracteres especiales que causen ruido, y que no tengan mayor valor semántico. Ejemplo de esto son puntos, comas, paréntesis, vocales con tilde, dígitos, etc.
- Stemming: Consiste en llevar cada palabra hasta su raíz o "mínimo esencial", independiente si ese cambio llegue a una palabra que no exista en el diccionario.
- Vectorización: Consiste en transformar un determinado texto en un vector numérico para ser procesado por el modelo. Para esto, existen distintas metodologías, siendo las más básicas basadas en la frecuencia de las palabras dentro de los documentos.

2.4. Extracción de características para texto

Para que los datos de textos libres puedan ser utilizados por un modelo matemático, estos deben ser transformados en espacios estructurados de características [12, 24]. Para lograr lo anterior, existen técnicas que serán indicadas a continuación.

Count Vectorizer

Es una forma de construcción de un vector de características, sobre el cual cada palabra es considerada una dimensión distinta. Sobre éstas, el valor de cada documento será la asignación de cada palabra con 1 o 0 en caso de existir en éste [23, 19].

Se asume con este método que las palabras que cuenten con un alto valor de esta métrica son consideradas importantes. La Ecuación 2.9 define esta lógica, donde $n_{\mathrm{i}j}$ representa el número de ocurrencias de una palabra t_{i}, en un documento d_j, y $\sum_k n_{k,j}$ representa el total de casos de palabras en un documento d_j.

$$CV_{\mathrm{i},j} = \frac{n_{\mathrm{i},j}}{\sum_k n_{k,j}} \tag{2.9}$$

Un problema que puede ocurrir con la utilización de este método es que asigna un mayor peso a las palabras con una mayor frecuencia, lo que puede no ser cierto dependiendo del contexto de los documentos. Para mitigar este efecto, se puede utilizar el método descrito a continuación.

Term Frequency - Inverse Document Frequency (TF-IDF)

Este método se utiliza para evaluar la relación de una palabra en un conjunto de documentos. Esta métrica se compone de la relación de dos elementos, que son TF e IDF [10, 4]. Describiendo los elementos por separado:

TF: Esta métrica calcula el número de ocurrencias que tiene cada palabra en un archivo. Donde $n_{\mathrm{i},j}$ representa el número total de ocurrencias de la palabra t_{i} en el documento d_j, y $\sum_k n_{k,j}$ representa el total de número de ocurrencias de palabras en el documento d_j. Lo anterior está representado por la Ecuación 2.10

$$TF_{\mathrm{i},j} = \frac{n_{\mathrm{i},j}}{\sum_k n_{k,j}} \tag{2.10}$$

DF: Esta métrica mide cuantas veces cada palabra aparece en la colección de documentos, y se calcula a través de la división del total de documentos por el número de documentos en los que aparece la palabra t_j. Donde $|D|$ representa el número total de documentos, y $|\mathrm{d}_j \in D : t_j \in \mathrm{d}_j|$ representa el número de documentos en los que la palabra t_j aparece. Lo anterior está representado por la Ecuación 2.11

$$DF_{\mathrm{i},j} = \frac{|\mathrm{d}_j \in D : t_j \in \mathrm{d}_j|}{|D|} \tag{2.11}$$

IDF: El DF cuenta cuántas veces una palabra en específico se encuentra dentro de todos los documentos. El encontrar una palabra con un valor alto de esta métrica, se asume que no es una palabra importante porque aparece constantemente en todos los documentos. Por lo tanto, el inverso de esta métrica (IDF), indica la importancia de una palabra en un conjunto de documentos. Lo anterior está representado por la Ecuación 2.12

$$IDF_{\mathrm{i},j} = \log \frac{|D|}{|\mathrm{d}_j \in D : t_j \in \mathrm{d}_j|} \tag{2.12}$$

Al utilizar ambos elementos de la forma $TFIDF = TF * IDF$, el contar con un valor elevado nos indica que una palabra posee una frecuencia alta en un documento determinado, pero que la frecuencia de ésta en todos los documentos es baja.

Word2Vec

Es un método que aprende una representación vectorial para cada palabra en el vocabulario entregado, que a diferencia de los modelos anteriores basados en frecuencias, este se basa en un modelo de redes neuronales [13, 17].

Este método puede ser utilizado a través de dos arquitecturas distintas: Continuous Bag-Of-Words (CBOW) y Skip–Gram (SG). La arquitectura CBOW busca predecir la palabra actual usando el contexto del entorno a través de la minimización de una función de pérdida. Por otro lado, la arquitectura Skip-Gram busca predecir la palabra actual con el contexto del entorno dada la palabra objetivo, en lugar de predecir la propia palabra objetivo.

La elección de CBOW o SG dependerá de cada caso, pero en términos generales SG tiene un mejor funcionamiento con sets de datos pequeños, representando de mejor manera las palabras menos frecuentes. CBOW usualmente se entrena más rápidamente que SG, y genera una mejor representación de las palabras más frecuentes [14].

2.5. Encoders para variables categóricas

La utilización de técnicas de encoder, permiten modificar las variables categóricas de un conjunto de datos a una forma numérica, para que puedan ser correctamente utilizadas por un modelo matemático. En esta sección se revisan dos técnicas distintas:

One Hot Encoding (OHE)

Esta técnica consiste en añadir tantas columnas como categorías se encuentren en el vector de variables categóricas elegidas, para luego ser completados de manera binaria dependiendo del valor de la categoría para cada fila.

De manera general, si consideramos un vector v de tamaño n y m categorías, se agregan $m-1$ nuevas columnas. Para completar estas nuevas columnas, se asigna de manera binaria 1 en caso de que el elemento v_{i} contiene la categoría m_{i}, en caso contrario se asigna cero [3]. En la Tabla 2.3 se muestra un ejemplo de cómo se transforma un vector de 3 valores con tres categorías distintas.

Original		One Hot Encoding			
Dato	Categoría	Dato	cat_a	cat_b	cat_c
1	a	1	1	0	0
2	b	2	0	1	0
3	c	3	0	0	1

Tabla 2.3: Ejemplo transformación de variable categóric a con OHE.

Ordinal Encoder

Esta técnica asigna valores numéricos consecutivos a las categorías de un vector a medida que aparecen. De manera general, considerando un vector v de tamaño n y m categorías, y $M = [m_1, \ldots, m_m]$, a todas las posibles categorías distintas, se codifica M como $M\prime = [0, \ldots, m\ -1]$ [16]. En la Tabla 2.4 se ejemplifica la utilización de esta técnica.

Original		Original	
Dato	Categoría	Dato	Categoría
1	a	1	0
2	b	2	1
3	c	3	2

Tabla 2.4: Ejemplo transformación de variable categórica con Ordinal Encoding.

Capítulo 3

Metodología

Para cumplir con los objetivos planteados, se propone desarrollar un modelo supervisado de clasificación, que permita identificar la correctitud de la etiqueta que asigna el comprador a un determinado producto o servicio. Para facilitar la utilización de éste por los usuarios, se pretende su implementación en un servicio web, que al ingresar una tabla de CCPP agregue una columna con la predicción del código de producto. Esto permitiría que los usuarios no tengan que inspeccionar visualmente cada una de las compras públicas para determinar su correctitud, sino que se realizaría de manera automática, sin grandes tiempos de espera y sin necesidad de requerir expertos en el rubro.

La solución descrita se puede diseccionar en los siguientes puntos ilustrados en la Figura 3.1.

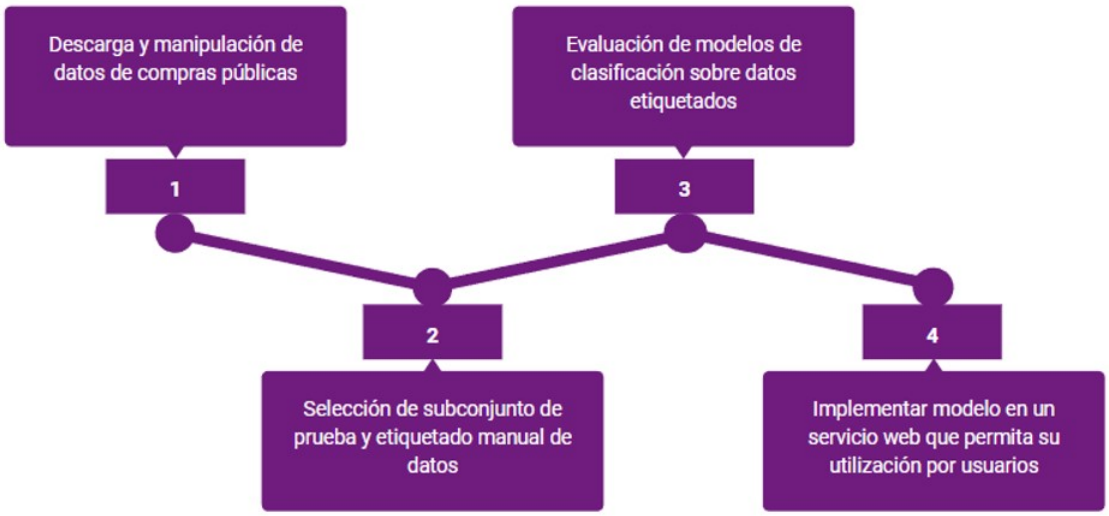


Figura 3.1: Diagrama de flujo de metodología

Cabe destacar que es posible reproducir cada una de estas etapas mediante los códigos que se encuentran en el siguiente enlace de GitHub: `https://github.com/HugoGallardoR/Tesis_CCPP`

En lo que sigue de este capítulo se describirán de manera general cada una de estas etapas,

siendo profundizadas con sus resultados y criterios utilizados en los Capítulos 4, 5 y 6.

3.1. Descarga y manipulación de datos de compras públicas

Los datos históricos que se utilizan para entrenar y evaluar los modelos de clasificación provienen de archivos en formato CSV, con datos tabulares de órdenes de compra. Estos datos se descargan y descomprimen manualmente desde el portal de datos abiertos de la DCCP.

Dado que estos datos son de gran magnitud, para procesarlos correctamente de acuerdo con los recursos disponibles y permitir su fácil reproducibilidad, se realiza un proceso que almacena sólo los datos relevantes para el modelo en una base de datos local. Además, para reducir el tamaño del set de datos, se realiza un proceso de manipulación de datos en Python que reduce la cantidad de columnas presentes en las planillas de órdenes de compra, y acota los datos sólo a las categorías de TI.

Como resultado de esta etapa, se genera un set de datos único que consolide los datos de descripciones únicas de productos correspondientes a TI, en los periodos que se necesiten para el estudio.

3.2. Selección de subconjunto de prueba y etiquetado de datos

A partir del set de datos generado en la etapa anterior, se genera un subconjunto de éste para ser etiquetado manualmente. Esto resulta necesario dada la gran cantidad de tipos de productos que se encuentran dentro del rubro de TI, siendo aproximadamente 500. Los criterios para delimitar el tamaño del nuevo subconjunto dependen de los recursos disponibles para realizar el etiquetado en un tiempo determinado, y del tipo de modelo que se desee generar. Para facilitar esta tarea, se aplican técnicas de análisis de datos usando tablas dinámicas y Python.

La tarea de etiquetado se realiza mediante la inspección visual, identificando si el texto libre de la descripción del producto tiene concordancia con el código de producto asignado por el comprador. Estos datos se anexan como columna al subconjunto del set de datos.

Una vez etiquetados manualmente los datos, se aplican técnicas de preprocesamiento de texto a las descripciones de texto libre para mejorar la precisión de los modelos a evaluar.

El resultado de esta etapa es un set de datos único con la consolidación de datos etiquetados de los productos seleccionados. Lo anterior se usará como insumo directo para el entrenamiento de los modelos de clasificación.

3.3. Evaluación de modelos de clasificación sobre datos etiquetados

Teniendo como entrada el set de datos generado en la etapa anterior, se procede a desarrollar y evaluar el mejor modelo de clasificación. Los modelos pueden trabajar con tres tipos de inputs del set de datos: texto libre de la descripción del producto, código de producto y etiqueta manual de concordancia texto-código.

Una vez desarrollado el modelo, dado que es un objeto creado a partir de librerías en Python, se decide su almacenamiento como un archivo "Pickle". Este formato es nativo de Python, que permite la serialización en bytes de sus objetos para luego ser utilizados fácilmente por otras aplicaciones de este mismo lenguaje.

A grandes rasgos, el desarrollo del modelo se desarrolla en Python mediante las siguientes etapas:

1. Importar y manipular datos de entrada de acuerdo con las necesidades del proceso del modelo.
2. Separar datos de entrada en datos de entrenamiento y testing.
3. Definir los encoders a probar para el texto libre.
4. Definir los encoders a probar para el código de producto.
5. Definir los modelos de clasificación a probar.
6. Definir la métrica de evaluación a utilizar.
7. Realizar un Grid Search que encuentre la mejor métrica de evaluación, revisando la combinación entre los encoders y modelos definidos en los puntos tres, cuatro y cinco.
8. Una vez encontrado el mejor modelo con sus combinaciones, se aplica otro Grid Search para encontrar los mejores hiperparámetros del modelo.
9. Exportar el modelo junto a sus procesos de encoders en un archivo Pickle.
10. Evaluar el modelo con datos de producción de un periodo distinto a los utilizados en el modelo.
11. Evaluación de casos forzados del modelo.

En caso de que el desempeño del modelo en datos de producción cumpla con las expectativas, se procede a utilizar el archivo Pickle exportado en la siguiente sección.

3.4. Implementación de servicio web para usuarios

Ya con el modelo de clasificación entrenado y exportado en un archivo Pickle, se debe buscar un SW que idealmente cumpla con las siguientes características:

- Estar siempre disponible para el usuario, sin depender de la disponibilidad del desarrollador.
- No suponer costo monetario para el usuario ni para el desarrollador.
- Debe ser sencillo en su utilización. Es decir, el usuario no debe codificar ni generar cambios estructurales a la aplicación.

Una vez encontrado el SW que cumpla con lo anterior, se debe desarrollar una aplicación compatible con éste, que permita la utilización del modelo en un nuevo conjunto de datos.

Capítulo 4

Tratamiento de datos

En esta sección se profundiza en las etapas 1 y 2 de la metodología mencionadas en las Secciones 3.1 y 3.2 respectivamente. Indicando los criterios utilizados y los resultados obtenidos en cada una de éstas.

4.1. Descarga y almacenamiento de órdenes de compra

A continuación, se presenta de manera esquemática en la Figura 4.1 los pasos que se tomaron en la etapa 1:

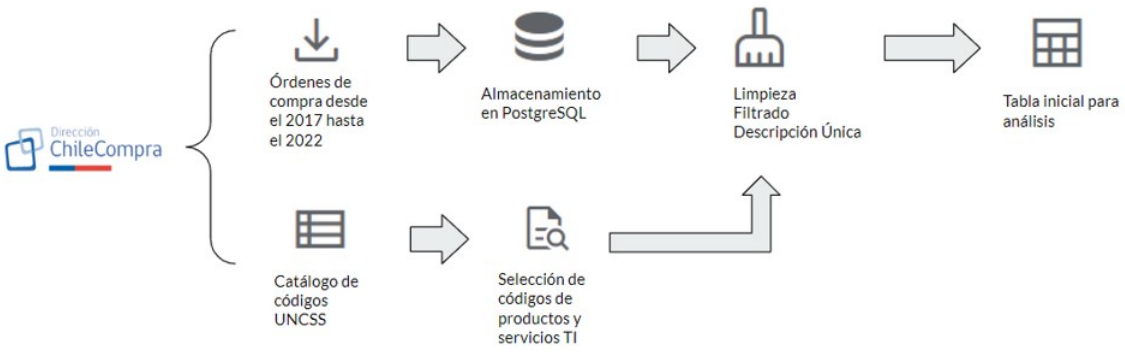


Figura 4.1: Ilustración de flujo etapa 1.

Los datos iniciales utilizados provienen de archivos CSV que pueden ser descargados del portal de datos abiertos de la DCCP: `https://datos-abiertos.chilecompra.cl/descargas`. Todas las descripciones de las columnas del archivo pueden obtenerse de esa misma página. En específico, los datos descargados en la sección "Descarga masiva por URL" fueron las órdenes de compra de todos los meses de los años 2017 a 2022. Luego, se descargan los datos del portal, y después se descomprimen y almacenan en una carpeta determinada todos los archivos CSV que provienen de estos datos.

Luego de guardar todos los archivos necesarios, es posible apreciar que éstos tienen un gran tamaño que podrían generar problemas al intentar manipularlos o generar modelos

Figura 4.2: Sección de descarga de órdenes de compra en portal DCCP.

directamente. En específico, todos los datos descargados y descomprimidos pesan aproximadamente 45 gigabytes. Dado esto, y que los datos son tabulares, se decide almacenarlos en una base de datos local PostgreSQL.

La subida de archivos a la base de datos local se realiza mediante un proceso desarrollado en Python. El código se encuentra dentro de la carpeta de GitHub "Extraccion_arch", resultando en la generación de una tabla por cada archivo mensual descargado (71 tablas), con exactamente las mismas columnas y tipos de datos.

Se debe tener en cuenta que los datos del archivo "2017_07.csv"generan problemas al tener un formato incorrecto dentro de sus columnas, por lo que quedan fuera del proceso de carga.

4.2. Descarga y almacenamiento de datos complementarios

De forma paralela a la descarga de órdenes de compra, desde el sitio de complementos de la DCCP se descargan y almacenan los siguientes archivos:

Catálogo de códigos UNCSS: Es un archivo XLSX con un listado de todos los productos y servicios con su respectivo código de producto. El contenido de este archivo se almacena dentro de la base de datos en la tabla "rubros_onu".

Dado que el problema está acotado sólo a productos y servicios del área TI, sobre este archivo además se realiza una selección de los que corresponden a este rubro. Para esto, se utilizan 2 criterios:

- Todos los productos que tengan como rubro nivel 1: "Tecnologías de la información, telecomunicaciones y radiodifusión".

- Selección de servicios relacionados del rubro nivel 1: “Servicios basados en ingeniería, ciencias sociales y tecnología de la información”.

En la Tabla 4.1 se muestra un ejemplo de los datos del catálogo.

CodigoProductoONU	NombreProducto	RubroN1	RubroN2	RubroN3
43191501	Teléfonos celulares	Tecnolog	Dispositiv	Dispositivos de comunicación personal
43191502	buscapersonas o beeper	Tecnolog	Dispositiv	Dispositivos de comunicación personal
43191503	Teléfonos públicos de previo pago	Tecnolog	Dispositiv	Dispositivos de comunicación personal

Tabla 4.1: Catálogo de productos UNCSS proveniente del portal DCCP.

Los nuevos datos filtrados solo con productos TI, se guardan en la tabla “items_ti”

Instituciones Compradoras: Es un archivo en formato CSV, con el listado y clasificación de todas las entidades compradoras que han participado en el sistema de la DCCP. El contenido de este archivo se almacena dentro de la base de datos en la tabla “lista_compradores”.

Dado que para el análisis inicial fue necesario filtrar los datos de las municipalidades, se creó una tabla adicional llamada “municipalidades” a partir del archivo de instituciones compradoras.

4.3. Manipulación de datos de entrada

El objetivo en esta etapa es la de consolidar los datos de órdenes de compras obtenidas en la Sección 4.1, en dos tablas que puedan ser utilizadas correctamente como datos de entrada para los modelos.

De manera intuitiva, dado que los datos iniciales vienen a nivel de producto, es decir, una fila de la tabla es un producto, se pensaría que de esta misma manera deberían introducirse los datos de entrenamiento. Sin embargo, las descripciones en texto libre de estos productos en muchos casos se repiten múltiples veces, lo que generaría un sesgo en el modelo en caso de ser utilizados sin filtrar. Para evitar esto, se utilizaron criterios para filtrar los datos de entrada.

Como criterio inicial, se consideran solo las descripciones únicas por cada tipo de producto. Esto se consigue a través de una unión de todas las tablas generadas mensuales. Para lograr lo anterior, se hace necesario realizar una separación entre productos de Convenio Marco (CM) y las otras modalidades de compras, ya que ambos subconjuntos requieren de tratamientos distintos en el filtrado.

Además de lo anterior, desde esta sección y lo que sigue de este Capítulo, sólo se considerarán los datos del año 2017 a 2021, dado que los datos del 2022 se utilizarán después como testing adicional de los modelos.

Convenio Marco:

Para lograr lo anterior, se utiliza la consulta dentro de la base de datos de PostgreSQL del archivo "extraer_cm.sql". Esta genera una tabla única con la unión de todos los meses considerados en el estudio, para posteriormente ser exportado manualmente en un archivo CSV. A continuación, se indican los criterios utilizados para mejorar la calidad de los datos de entrada:

- El producto debe pertenecer a una orden de compra de CM.
- El código de producto por cada producto no debe ser nulo.
- El producto debe poseer al menos una descripción no vacía, ya sea del comprador o proveedor.
- Se remueve de las descripciones de texto libre del producto el código numérico de CM. Esto es una particularidad que solo se aplica a las compras de CM.

El archivo con los datos resultantes que será utilizado para la siguiente fase es data_agg_cm_raw.csv. Esta tabla contiene en total 25 distintos tipos de productos, y 3.025 descripciones únicas de productos.

Otras modalidades de compra:

De la misma manera que en el proceso de CM, se utiliza la consulta dentro de la base de datos de PostgreSQL del archivo extraer_not_cm.sql. Esta genera una tabla única con la unión de todos los meses considerados en el estudio, para posteriormente ser exportado manualmente en un archivo CSV. A continuación, se indican los criterios utilizados para mejorar la calidad de los datos de entrada:

- El código de producto por cada producto no debe ser nulo.
- El producto debe poseer al menos una descripción no vacía, ya sea del comprador o proveedor.
- El producto no debe pertenecer a una orden de compra de CM.

El archivo con los datos resultantes que será utilizado para la siguiente fase es "data_not_cm_raw.csv". Esta tabla contiene en total 500 distintos tipos de productos, y 162.427 descripciones únicas.

4.4. Selección de subconjunto de prueba y etiquetado manual de datos

A continuación, en la Figura 4.3 se presenta de manera esquemática los pasos que se tomaron para desarrollar la etapa 2 indicada en la Sección 3.2:

Figura 4.3: Ilustración de flujo etapa 2.

Tal como se pudo ver en la motivación de la tesis, existen problemas de calidad en la concordancia del texto libre con el código de producto, por lo que no se puede asumir que los datos de entrada son correctos. Dado esto, resulta necesario para este ejercicio el etiquetar manualmente un subconjunto de datos, para asegurarse que el modelo está recibiendo datos de calidad, y no tenga problemas a futuro en su predicción.

Para determinar cuáles son los tipos de productos a abarcar en el subconjunto, a través de tablas dinámicas se realizan agrupaciones de la cantidad de datos que posee cada tipo de producto, y se aplican los siguientes criterios:

- Solo productos que estén disponibles en Convenios Marco, ya que las descripciones y etiquetas de éstos han sido realizados correctamente por la DCCP. Esto permite ahorrar tiempo en el proceso de etiquetado.
- Se consideran en total solo 15 tipos de productos: 8 Productos y 7 Servicios.
- Los productos deben tener a lo menos 100 descripciones únicas.

Una vez delimitado el set de tipos de producto, se procede a realizar el etiquetado manual. Este consiste en abrir el archivo de descripciones únicas como planilla Excel, y revisar por cada fila si existe concordancia entre el la descripción única y el tipo de producto. En caso correcto, en la columna "Correcto" se agrega un 1, en caso contrario un 0. Además de lo anterior, en los casos incorrectos se agrega un comentario del por qué se consideró incorrecto.

Descripción	Código producto	Nombre producto	Revisado	Correcto	Comentarios
24 PLANES PARA PARA ESCUELA RURAL PUERTO NUEVO	81112101	Proveedores de servicio de Internet (ISP)	1	0	Mala descripción
SERVICIO DE TELEFONIA ABRIL 2019	43191504	Teléfonos fijos	1	0	Servicio de telefonía
SERVICIO REPARACIÓN SISTEMA CONDUCTO DE ASPIRADO DE TURBO PARA BUS CHEVROLET MODELO NQR916EURO V AÑO 2019 PLACA UNICA LCGV-16	43211609	Conectores o concentradores de bus serie universal	1	0	No TI
TELEFONO NITSUKO AT-70 VISOR 32 MEMORIAS - Solicitud 37 Oficina de Transparencia -	43191504	Teléfonos fijos	1	1	
Servicio de diseño y desarrollo del portal web para Revista FEN	81112103	Servicios de diseño de sitio Web	1	1	
Licencias Sistema Operativo Windows 10 PRO VLC	43231513	Paquetes de software para oficinas	1	1	

Tabla 4.2: Tabla ejemplo de datos manualmente etiquetados.

El número final de datos etiquetados por tipo de producto se basó en dos criterios:

- El total de descripciones únicas debe ser mínimo de 100.
- Debe existir un balance entre las etiquetas positivas y negativas: Un tipo de etiqueta no puede ser más del doble que la otra.

En la Tabla 4.3 se muestra un resumen del etiquetado manual por cada tipo de producto. Es posible ver que en total se etiquetaron manualmente alrededor de cinco mil descripciones únicas. Estos datos provienen de los archivos "etiquetado_aggCM.xlsx" y "etiquetado_notCM.xlsx".

Producto o Servicio	Tipo	Total etiquetas correctas	Total Etiquetas erróneas	Total Etiquetado
Análisis de sistemas	Servicio	81	60	141
Centro de servicios de datos	Servicio	329	225	554
Computadores de escritorio	Producto	115	62	177
Conectores o concentradores de bus serie universal	Producto	103	165	268
Diseño de base de datos	Servicio	197	129	326
Implantación de aplicaciones	Servicio	51	63	114
Impresoras de matriz de puntos	Producto	118	431	549
Mantenimiento y soporte de hardware	Servicio	134	146	280
Notebook	Producto	169	124	293
Paquetes de software para oficinas	Producto	276	195	471
Proveedores de servicio de Internet (ISP)	Servicio	424	306	730
Servicios de diseño de sitio Web	Servicio	132	102	234
Software de cadena de suministro y logística de Planificación	Producto	43	66	109
Teléfonos fijos	Producto	216	196	412
Teléfonos para usos especiales	Producto	159	123	282
		2.547	**2.393**	**4.940**

Tabla 4.3: Tabla ejemplo de datos manualmente etiquetados.

La impresora matriz de puntos es una excepción al segundo criterio por haber sido la primera prueba de etiquetado. Sin embargo, no se modificó dado que tras revisiones posteriores esto no afectó en la precisión de este tipo de producto.

El análisis de estos datos se encuentra en el archivo "analisis_datos_etiquetados.xslx"

4.5. Preprocesamiento de descripciones de texto libre

Tras una inspección visual el texto libre de las descripciones de productos, es posible notar que éste cuenta con una amplia variabilidad en su texto y mala calidad. Esto tiene el potencial de afectar negativamente el desempeño de los modelos de clasificación, por lo que es necesario la aplicación de técnicas de preprocesamiento de texto.

Como primer criterio, se realizó la concatenación de las descripciones de comprador y proveedor, dado que en múltiples casos se encontró que solo uno de éstos tiene una calidad aceptable para ser utilizado.

A continuación, se explicarán las técnicas de preprocesamiento de texto aplicadas sobre las descripciones de productos.

1. Tokenización: Se separan las palabras del texto para evitar problemas de separación de espacios.
2. Remover tildes: Permite reducir la variabilidad de las palabras en el modelo.
3. Remover puntuaciones: Se aplica ya que no aportan valor al modelo en este caso.
4. Remover palabras con baja cantidad de letras: Se remueven las palabras menores a 3 letras para disminuir la variabilidad de las palabras en el texto.
5. Remover stop words: Se aplica ya que no aportan valor al modelo en este caso.
6. Transformar todo a minúsculas: Permite reducir la variabilidad de las palabras en el modelo.
7. Eliminar caracteres no alfanuméricos: Permite eliminar caracteres especiales y emojis que no aportan valor al modelo.

En la Tabla 4.4 se muestra un ejemplo del resultado:

Descripción Comprador	Descripción Proveedor	Texto Limpio
Valor Mensual correspondiente al año 2022 Octubre, Noviembre y Diciembre, en los sigientes Hogares Indígenas: Hogar Pelontuwe Hogar Purram Hogar Pewenche Hogar Centro Cultural Mapuche Hogar Newen Kimun Hogar Lawen Mapu	Se adjunta valor total neto según las bases meses de contrato 27	valor mensual correspondiente 2022 octubre noviembre diciembre sigientes hogares indigenas hogar pelontuwe hogar purram hogar pewenche hogar centro cultural mapuche hogar newen kimun hogar lawen mapu adjunta valor neto bases meses contrato
SERVICIOS DE ENLACE DE DATOS E INTERNET, SEDES SANTIAGO Y VALPARAÍSO, PARA LA BIBLIOTECA DEL CONGRESO NACIONAL	Enlaces datos e Internet Valparaíso y Santiago. Valor unitario mes neto	servicios enlace datos internet sedes santiago valparaiso biblioteca congreso nacional enlaces datos internet valparaiso santiago valor unitario neto
IMPRESORA EQUIVALENTE A MODELO Multifuncional L5290, COLOR, WIFI		impresora equivalente modelo multifuncional l5290 color wifi

Tabla 4.4: Tabla ejemplo de limpieza de texto.

El resultado del preprocesamiento se encuentra en la carpeta “preprocesamiento” en el archivo “inputs_total_csv.csv”.

Capítulo 5

Modelo de clasificación multiclase bajo el estándar UNCSS

En este Capítulo se detalla el proceso realizado correspondiente a la etapa tres de la metodología descrita en el Capítulo 3. Esta etapa consta del desarrollo y evaluación de un modelo de clasificación multiclase, que permita predecir el código de un producto a partir del texto libre de la descripción de este.

El desarrollo del modelo con sus resultados se encuentra en la carpeta "Modelos". Este se desarrolla completamente en código Python.

5.1. Preprocesamiento de datos de entrada

Los datos de entrada que se utilizarán para este modelo son los datos preprocesados resultantes de la Sección 4.5. Sobre éstos se realiza un primer filtro, utilizando solo los casos de concordancias correctas entre código de producto y descripción de producto. Lo anterior, dado que en este caso la etiqueta de correcto o incorrecto no se incluye como parámetro, por lo que los casos incorrectos afectarían a la precisión del modelo.

Con esto, el número de descripciones únicas de los 15 tipos de producto a revisar se reduce a un total de 2.553, considerando 4.888 palabras distintas. La proporción de descripciones de los tipos de producto se ilustra en el gráfico de la Figura 5.1.

Además de lo anterior, los datos para ser correctamente evaluados por el modelo deben ser separados en subconjuntos de entrenamiento y test. Los criterios utilizados para esta separación son:

- Determinar porcentaje de separación: Se utiliza por defecto lo recomendado en la literatura, que es separar el subconjunto en 70 % de entrenamiento y 30 % de testing. Este porcentaje puede ser modificado a futuro en caso de que la evaluación de los modelos sea muy distinta a lo mostrado por la validación cruzada.

- Separación estratificada: Se utiliza esta técnica para que la separación porcentual sea por tipo de producto. Esto permite la representación de todas las clases dentro del modelo.

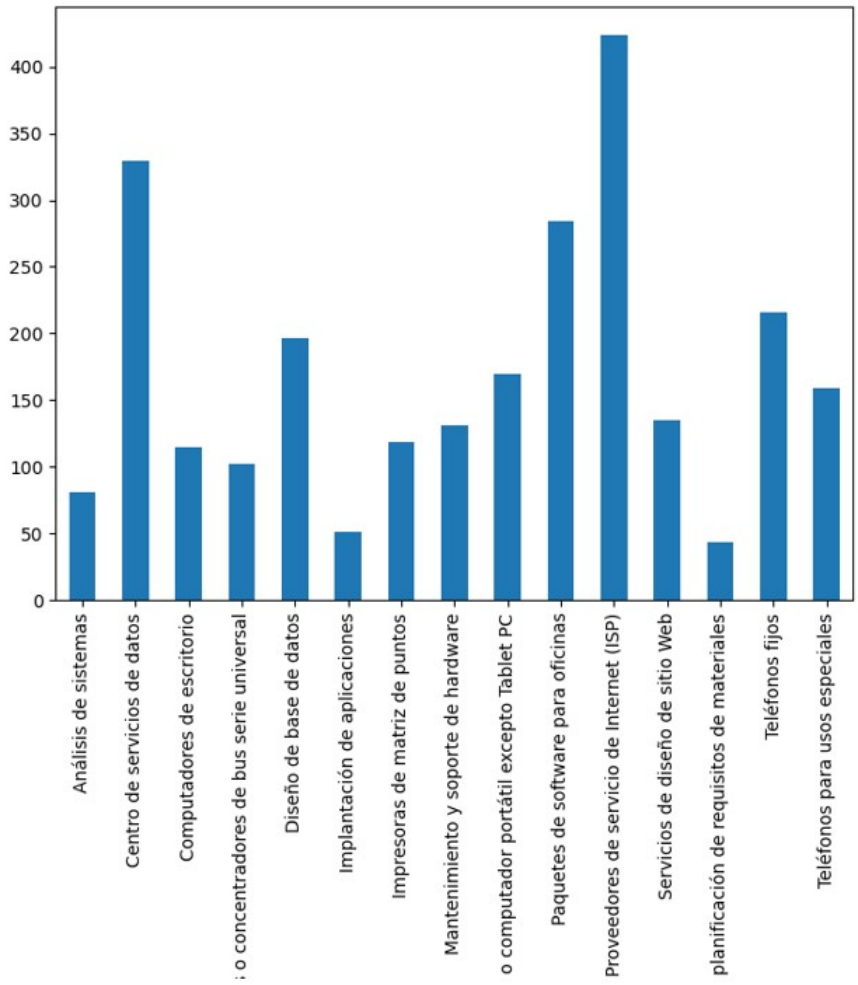


Figura 5.1: Gráfico de descripciones únicas por tipo de producto.

En la Tabla 5.1 se muestra un ejemplo de como son los datos de entrada al modelo. En estos, se utiliza la descripción del producto en la columna "texto" para predecir el código de producto en la columna "cod_prod".

cod_prod	texto
81111808	analista funcional senior valor hora habil
43211609	conector rj45 seri contactor kvar 1032854 conector rj45 seri contactor kvar
43211503	notebook cotizacion 24497
43191507	arriendo telefono satelital
43212102	impresora matriz punto okidata okidata

Tabla 5.1: Tabla ejemplo de datos de entrada al modelo.

5.2. Búsqueda de mejor modelo de clasificación

Con los subconjuntos de datos obtenidos en la sección anterior, se procede a "buscar" el modelo con mejor rendimiento. Para esto, se crea un flujo con dos procesos:

1. Aplicar una técnica de encoding sobre los datos de texto libre para poder ser utilizados por el modelo.
2. Aplicar el modelo sobre las descripciones transformadas por el encoder, y sus códigos de producto.

Dado que a priori no se sabe cuál de todas estas técnicas tendrá mejor rendimiento, se realiza un Grid Search que busca la mejor combinación. En la Figura 5.2 se indican cuáles son las técnicas a evaluar.

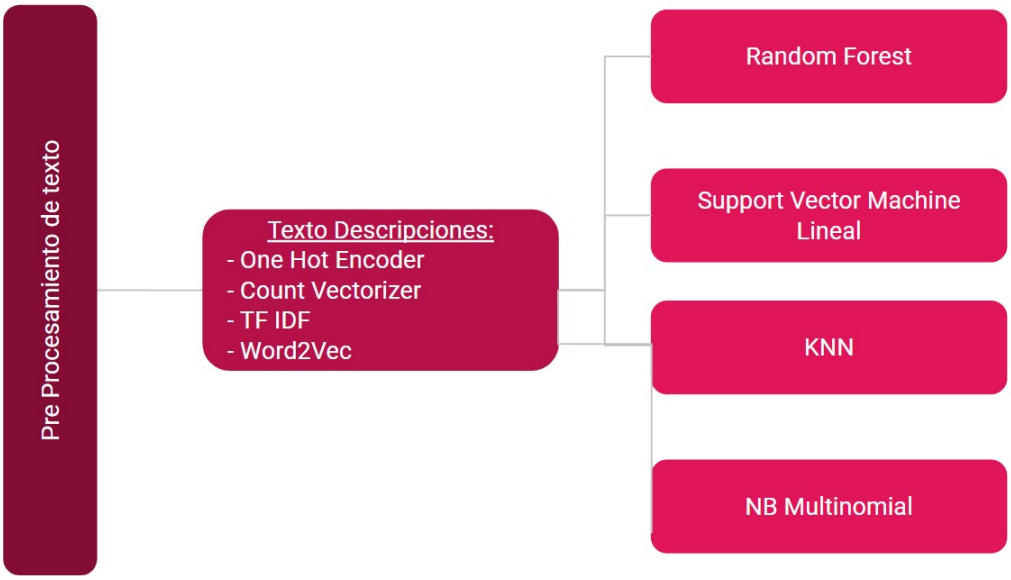


Figura 5.2: Flujo de técnicas a utilizar para desarrollar el mejor modelo multiclase.

Los criterios utilizados para configurar el Grid Search son:

- Cada técnica de encoder se combina con cada uno de los modelos de clasificación.
- La métrica de rendimiento de cada combinación es el F1-score macro average.
- Por cada combinación, se evalúa con una validación cruzada usando 10 K-fold. Se decide utilizar esta técnica dado que toma un tiempo razonable de tiempo de ejecución, y al compararlo con un caso aislado usando "leave-one-out", la diferencia no fue de más de 1 %.

La combinación con mejor F1-score macro average fue utilizar como encoder TF-IDF, y como modelo de clasificación SVM Lineal.

En la Tabla 5.2 se muestra un ejemplo de los datos de salida del modelo. La columna "cod_prod" se considera como la etiqueta real, y "predicción" como la etiqueta generada por el modelo. En este caso, solo las tres primeras filas fueron correctamente etiquetadas por el modelo.

cod_prod	texto	predicción
81111808	analista funcional senior valor hora habil	81111808
43211609	conector rj45 seri contactor kvar 1032854 conector rj45 seri contactor kvar	43211609
43211503	notebook cotizacion 24497	43211503
43191507	arriendo telefono satelital	43231513
43212102	impresora matriz punto okidata okidata	43211503

Tabla 5.2: Tabla ejemplo de datos de salida del modelo multiclase.

5.3. Mejora de hiperparámetros de mejor modelo encontrado

Una vez encontrada la mejor combinación, se procede a intentar mejorar el desempeño del modelo. Para esto, se vuelve a generar un Grid Search, pero buscando la mejor combinatoria entre los hiperparámetros de TF IDF y SVM lineal. En el caso de TF IDF, se itera con distintos rangos de n-gramas, desde unigramas a trigramas. En el caso de SVM lineal, se itera con el parámetro de regularización (C) entre 1 y 1.000.

	precision	recall	f1-score	support
Análisis de sistemas	0.93	0.98	0.96	65
Software de cadena de suministro y logística de planificación de requisitos de materiales	1.00	0.96	0.98	48
Impresoras de matriz de puntos	0.96	0.98	0.97	51
Conectores o concentradores de bus serie universal	0.97	0.91	0.94	34
Diseño de base de datos	1.00	0.94	0.97	31
Servicios de diseño de sitio Web	0.97	0.91	0.94	35
Proveedores de servicio de Internet (ISP)	0.73	0.62	0.67	13
Centro de servicios de datos	0.93	0.96	0.95	85
Teléfonos para usos especiales	0.71	0.67	0.69	15
Paquetes de software para oficinas	0.95	0.98	0.97	59
Implantación de aplicaciones	1.00	0.92	0.96	24
Mantenimiento y soporte de hardware	0.89	0.87	0.88	39
Computadores de escritorio	0.93	0.97	0.95	99
Notebook, laptop o computador portátil excepto Tablet PC	0.98	0.97	0.98	127
Teléfonos fijos	0.93	0.98	0.95	41
accuracy			0.95	766
macro avg	0.93	0.91	0.92	766
weighted avg	0.95	0.95	0.95	766

Figura 5.3: Métricas de desempeño de modelo multiclase sobre datos de prueba.

Como es posible ver en las métricas de desempeño de la Figura 5.3, el modelo se comporta con un alto F1-score en la gran mayoría de las clases, con excepción de las que tienen un soporte muy bajo. Revisando el F1-score macro average es posible ver que un 92 % puede considerarse alto teniendo en cuenta la baja cantidad de datos utilizados. Dado este buen comportamiento, el modelo se considera apto para ser evaluado con compras del año 2022, las cuales son completamente desconocidas por el modelo.

5.4. Evaluación de modelo en datos de 2022

Esta sección consta de dos etapas, la primera es la de importar y manipular los datos del año 2022, y la segunda es la aplicación del modelo sobre estos datos para determinar su rendimiento.

5.4.1. Importar datos de compras del año 2022

En la carpeta "Evaluacion_Modelos" se desarrolla un proceso en Python que importa los datos descargados del año 2022 a la base de datos, y exporta en un archivo CSV un subconjunto de este considerando solo los 15 productos seleccionados. Cabe destacar que a diferencia del proceso de los datos de entrenamiento descrita en el Capítulo /refcap4, no se realiza el filtro de quitar las descripciones duplicadas de productos.

En la Figura 5.4, se indica la cantidad de datos por producto. El producto con mayor cantidad de datos son los Notebooks con alrededor de 6.000 compras, y el menor "análisis de sistemas" con 33.

5.4.2. Aplicación de modelo sobre compras del año 2022

El modelo generado en la Sección 5.3 se exporta en formato Pickle, y se aplica sobre el set de datos del 2022. Tras esto, los códigos de producto predichos por el modelo se agregan al set de datos de entrada, para poder posteriormente ser comparados con los códigos de productos ingresados por los usuarios.

Para evaluar el resultado, se evalúa la concordancia del código de producto original con el código de producto predicho. En caso de ser iguales se considera correcto, en caso contrario incorrecto. Sobre esta lógica, se calcula el F1-score por cada uno de los tipos de productos evaluados, y el F1-score macro average sobre todo el conjunto. El resultado de esta evaluación se muestra en la Imagen 5.5.

En esta podemos dar cuenta que el modelo al ser utilizado sobre nuevas descripciones de productos no fue capaz de generalizar con el mismo desempeño mostrado con los datos de entrenamiento, disminuyendo de un 92 % a 49 % en su F1-score macro average.

Tras una inspección visual de los que el modelo clasificó incorrectamente, fue posible identificar que en casos en los que a través de la descripción no es posible determinar a un tipo de producto en concreto, se fuerza una etiqueta entre las 15 existentes. Lo anterior, sigue la lógica del modelo, que asigna automáticamente el tipo de producto con mejor probabilidad, aunque esta sea baja. Dado esto, en la siguiente sección se realiza una variación en el modelo que permita una mejor identificación de estos casos.

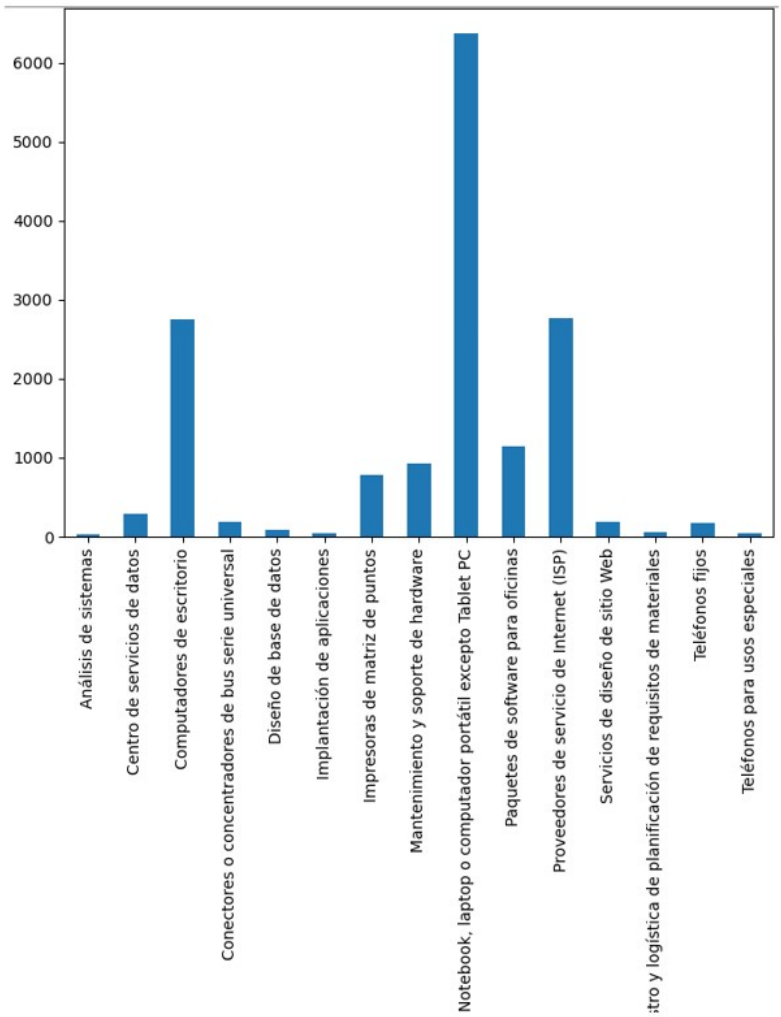


Figura 5.4: Gráfico de datos de compras por tipo de producto en el 2022.

	precision	recall	f1-score	support
Impresoras de matriz de puntos	0.37	0.72	0.49	172
Conectores o concentradores de bus serie universal	0.04	0.17	0.06	36
Notebook, laptop o computador portátil excepto Tablet PC	0.94	0.72	0.81	6372
Paquetes de software para oficinas	0.64	0.66	0.65	2755
Computadores de escritorio	0.35	0.86	0.49	188
Servicios de diseño de sitio Web	0.86	0.80	0.83	779
Proveedores de servicio de Internet (ISP)	0.07	0.37	0.12	49
Mantenimiento y soporte de hardware	0.58	0.59	0.58	1145
Diseño de base de datos	0.10	0.44	0.16	43
Centro de servicios de datos	0.15	0.13	0.14	78
Implantación de aplicaciones	0.28	0.36	0.32	33
Software de cadena de suministro y logística de planificación de requisitos de materiales	0.55	0.69	0.61	920
Teléfonos para usos especiales	0.45	0.60	0.51	286
Teléfonos fijos	0.94	0.97	0.95	2763
Análisis de sistemas	0.52	0.75	0.61	183
accuracy			0.74	15802
macro avg	0.45	0.59	0.49	15802
weighted avg	0.79	0.74	0.76	15802

Figura 5.5: Métricas de desempeño de modelo sobre set de datos de 2022.

5.5. Post procesamiento de modelo para identificar casos forzados

Para poder identificar los casos en los que el modelo está potencialmente forzando un código de producto, se realiza un post procesamiento sobre las predicciones obtenidas. En concreto, se busca encontrar los casos en los que el modelo identifica una baja probabilidad de precisión para todos los tipos de producto. Estos casos serían renombrados como "desconocidos", permitiendo al usuario descartar dichas predicciones.

Para esto se realizan dos acciones:

- Dado que el modelo Linear SVM implementado en Scikit-learn no permite la obtención de las probabilidades de confianza por cada uno de los 15 tipos de producto, fue necesario aplicar sobre éste una capa de calibración sobre el modelo (CalibratedClassifierCV). Cabe destacar que se revisaron nuevamente las métricas de desempeño con este cambio, sin embargo, no se encontraron diferencias en éstas.
- Sobre los datos predichos, se desarrolla un proceso que revise la confianza en la predicción sobre cada uno de los 15 tipos de producto. En caso de que ninguno cumpla con una probabilidad de confianza mínima determinada por el usuario, se determinará como "desconocido"

Para evaluar el impacto de este post procesamiento, se realiza el gráfico de la Figura 5.6 que nos permite visualizar cómo aumenta el porcentaje de etiquetas desconocidas sobre el total de datos, cuando se aumenta el porcentaje mínimo de precisión requerida para considerar a un código de producto como correcto.

Es posible ver en el gráfico de la Figura 5.6, que manteniendo la precisión mínima requerida hasta aproximadamente un 70 %, se pierden alrededor de un máximo de 11 % de las predicciones totales del modelo. Este mismo ejercicio se realizó sobre los datos de evaluación del año 2022, como se muestra en el siguiente gráfico de la Figura 5.7.

A diferencia del caso anterior que contaba con un modelo con una predicción de 92 % sobre los datos de prueba, es posible distinguir que delimitando un mínimo de probabilidad de confianza de un 70 %, perdemos aproximadamente un 37 % de todas las predicciones.

5.6. Conclusiones del desarrollo del modelo multiclase

Tras los resultados obtenidos en este ejercicio se puede concluir lo siguiente:

- Al desarrollar el modelo con los datos de entrenamiento etiquetados correctamente, se puede llegar a un porcentaje relativamente alto en los datos de prueba con un linear SVM y encoder TF IDF para el texto. Sin embargo, la cantidad de datos ingresados no permitió que el modelo pudiese generalizar correctamente con los datos de evaluación

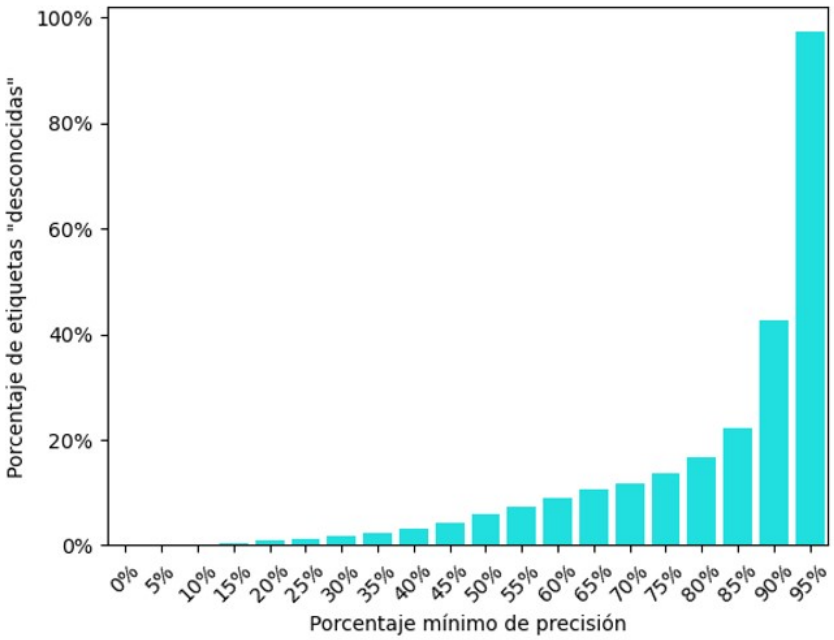


Figura 5.6: Gráfico de porcentaje de casos desconocidos por porcentaje de precisión en datos de entrenamiento.

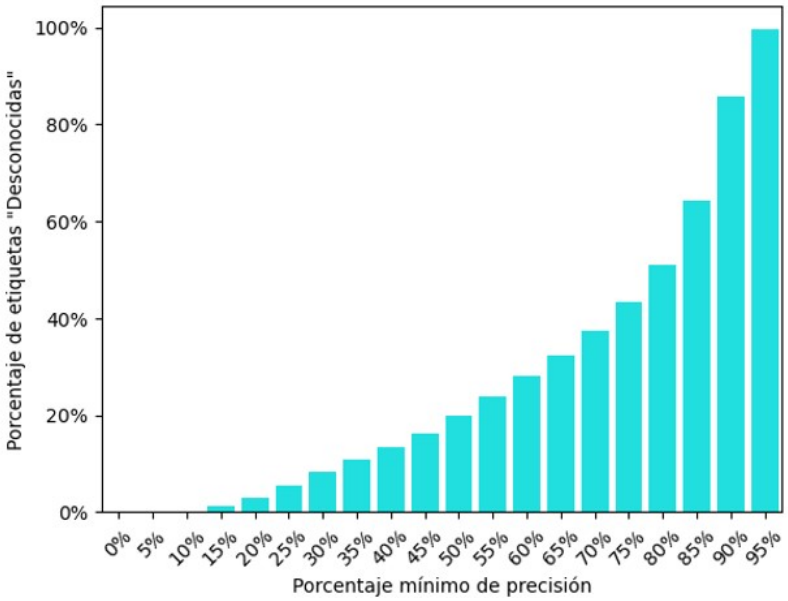


Figura 5.7: Gráfico de porcentaje de casos desconocidos por porcentaje de precisión en datos de evaluación.

del año 2022, los cuales en muchos casos poseen malas descripciones y palabras no vistas por el modelo desarrollado. Esto lleva a un bajo F1-macro average de un 49 %, que no es considerado aceptable para resolver el problema.

- Este tipo de modelos multiclase para poder desempeñarse correctamente requiere además de las potenciales clases erróneas que pueda tomar. Tomando como ejemplo el caso visto en el problema, se deberían agregar también productos de “Ducha teléfono”. Esto

requiere de aún más recursos para etiquetar manualmente productos.

- Para reducir el impacto de los códigos de productos forzados por el modelo, se realiza un post procesamiento que puede usar como entrada el porcentaje de confianza mínimo que determine el usuario. Sin embargo, si el modelo inicialmente no posee un buen rendimiento, el número de códigos de productos predicho puede llegar a reducirse en un 50 %.
- Dado que el modelo tuvo un buen comportamiento con las descripciones correctas que puede ver el modelo, se recomienda ser utilizado en un subconjunto acotado de productos, realizando una limpieza más exhaustiva de descripciones de mala calidad.

Dados estos resultados de este enfoque, se decide abordar en el Capítulo 6 un enfoque distinto para resolver este problema.

Capítulo 6

Modelo de clasificación binario bajo el estándar UNCSS

Dados los problemas que se indican en el Capítulo 5, se decide desarrollar un modelo con un enfoque distinto. Ya no se busca que éste genere una predicción con el código de producto basado en una descripción, sino que el modelo debe identificar si la concordancia entre el tipo de producto y su descripción es correcta. Dada esta característica, el modelo pasa a ser tratado como binario y no multiclase. Éste cumple de igual manera con el objetivo del problema, que es la fiscalización de los códigos de productos ingresados por el usuario.

6.1. Preprocesamiento de datos de entrada

Los datos de entrada que se utilizarán para este modelo son los datos preprocesados resultantes de la Sección 4.5. A diferencia del modelo anterior, en este caso si son utilizados los datos de etiquetas incorrectas. La cantidad de datos totales a ingresar es alrededor de 5.000, considerando 7.602 palabras distintas.

Además de lo anterior, los datos para ser correctamente evaluados por el modelo deben ser separados en subconjuntos de entrenamiento y test. Los criterios utilizados para esta separación son:

- Determinar porcentaje de separación: Se utiliza por defecto lo recomendado en la literatura, que es separar el subconjunto en 70 % de entrenamiento y 30 % de testing. Este porcentaje puede ser modificado a futuro en caso de que la evaluación de los modelos sea muy distinta a lo mostrado por la validación cruzada.
- Separación estratificada: Se utiliza esta técnica para que la separación porcentual sea por tipo de producto. Esto permite la representación de todas las clases dentro del modelo.

La proporción de productos con sus etiquetas correctas e incorrectas se pueden visualizar en la Figura 6.1.

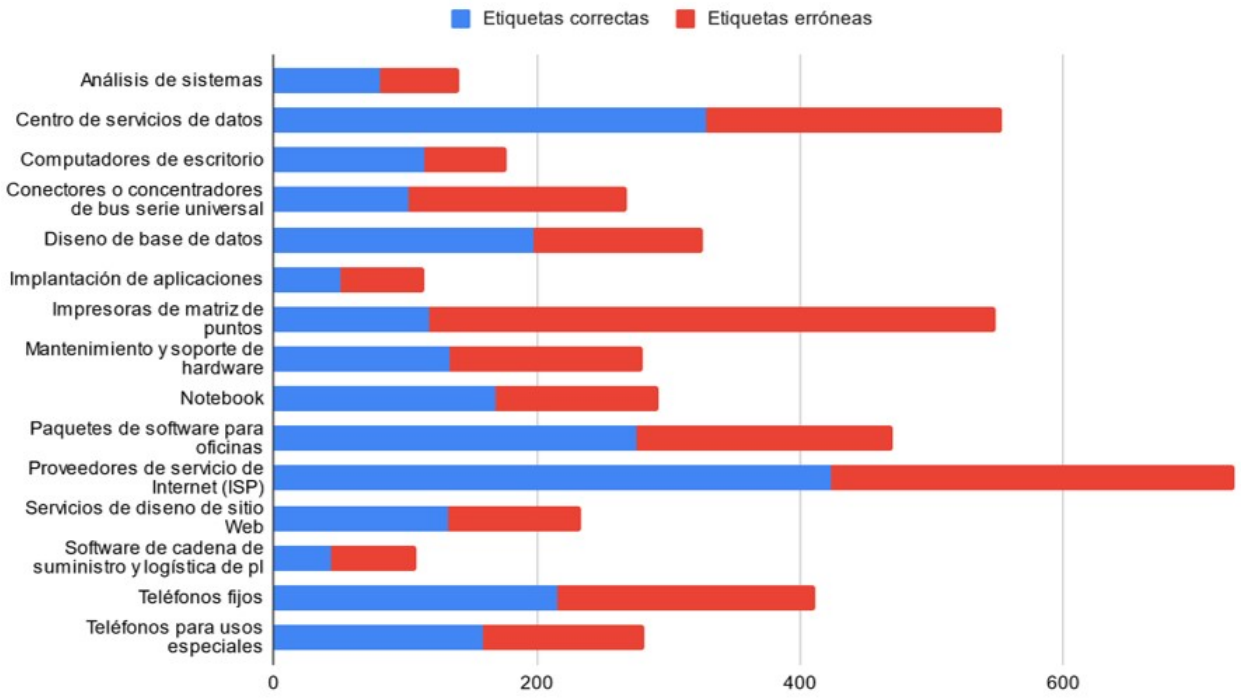


Figura 6.1: Gráfico de descripciones únicas por tipo de producto con casos correctos e incorrectos.

En la Tabla 6.1 se muestra un ejemplo de los datos de entrada al modelo. En estos, se utiliza la descripción del producto en la columna "texto", y el código de producto en la columna "cod_prod" para predecir la columna "correcto".

cod_prod	texto	correcto
81112101	enlace datos internet conexion internet 100mbps nac10mbps internac region valor mensual	1
43212102	Copias Blanco y negro Copias Blanco y negro	0
43191504	telefono inalambrico equivalente panasonic tgb110lcb	1
81111508	instalacion soportes cables bases adjuntas	0

Tabla 6.1: Tabla ejemplo de datos de entrada al modelo bineario.

6.2. Búsqueda de mejor modelo de clasificación

Con los subconjuntos de datos obtenidos en la sección anterior, se procede a "buscar" el modelo con mejor rendimiento. Para esto, se crea un flujo con tres procesos:

1. Aplicar una técnica de encoding sobre los datos de texto libre resultando en una matriz numérica.

2. Aplicar una técnica de encoding sobre los códigos de productos resultando en una matriz numérica.
3. Concatenar los resultados de ambas transformaciones anteriores en una única matriz para ser utilizados como parámetros del modelo.
4. Aplicar el modelo sobre la matriz resultante como entradas, para predecir el resultado binario de la variable "correcto".

Dado que a priori no se sabe cuál de todas estas técnicas tendrá mejor rendimiento, se realiza un Grid Search que busca la mejor combinación. En el flujo de la Figura 6.2 se indican cuáles son las técnicas evaluadas.

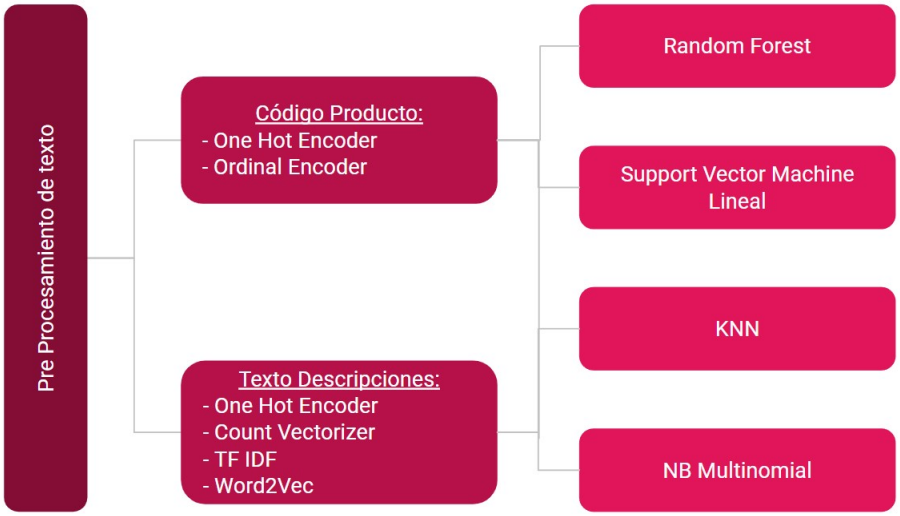


Figura 6.2: Flujo de técnicas a utilizar para desarrollar el mejor modelo multiclase.

Los criterios utilizados para configurar el Grid Search son:

- Las distintas técnicas de encoders aplicadas tanto al código de producto como a las descripciones de texto, se combinan con cada uno de los modelos de clasificación.
- La métrica de rendimiento de cada combinación es el F1-score macro average.
- Por cada combinación, se evalúa con una validación cruzada usando 10 K-fold. Se decide utilizar esta técnica dado que toma un tiempo razonable de tiempo de ejecución, y al compararlo con un caso aislado usando "leave-one-out", la diferencia no fue de más de 1 %.

La combinación con mejor F1-score fue utilizar como encoder de texto TF-IDF, como encoder de código de producto "One Hot Encoder" y como modelo de clasificación Random Forest.

En la Tabla 6.2 se muestra un ejemplo de los datos de salida del modelo. La columna "correcto" se considera como la etiqueta real, y "predicción" como la etiqueta generada por el modelo. En este caso, solo las dos primeras filas fueron correctamente etiquetadas por el modelo.

cod_prod	texto	correcto	predicción
81112101	enlace datos internet conexion internet 100mbps nac10mbps internac region valor mensual	1	1
43212102	Copias Blanco y negro Copias Blanco y negro	0	0
43191504	telefono inalambrico equivalente panasonic tgb110lcb	1	0
81111508	instalacion soportes cables bases adjuntas	0	1

Tabla 6.2: Tabla ejemplo de datos de salida del modelo.

6.3. Mejora de hiperparámetros de mejor modelo encontrado

Una vez encontrada la mejor combinación, se procede a intentar mejorar el desempeño del modelo. Para esto, se vuelve a generar un Grid Search, pero buscando la mejor combinatoria entre los hiperparámetros de TF-IDF y Random Forest. En el caso de TF IDF, se itera con distintos rangos de n-gramas, desde unigramas a trigramas. En el caso de Random Forest, con un número de árboles entre 300 y 1.000.

	precision	recall	f1-score	support
0	0.88	0.90	0.89	669
1	0.91	0.90	0.90	779
accuracy			0.90	1448
macro avg	0.90	0.90	0.90	1448
weighted avg	0.90	0.90	0.90	1448

Figura 6.3: Métricas de desempeño de modelo binario sobre datos de prueba.

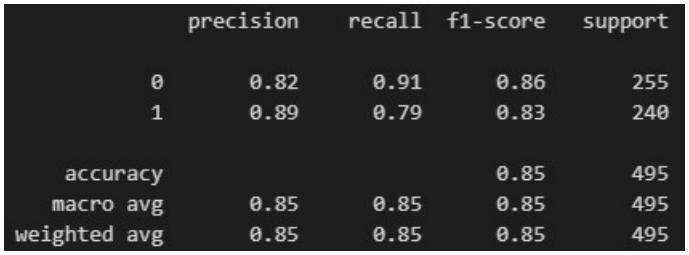
Es posible ver en la Figura 6.3, que el modelo se comporta con un alto puntaje F1-score macro avarage en ambas clases, promediando un 90 %. Esto se puede considerar alto teniendo en cuenta la baja cantidad de datos utilizados. Dado este buen comportamiento, el modelo se considera apto para ser evaluado con compras del año 2022, las cuales son completamente desconocidas por el modelo.

6.4. Evaluación del modelo en datos de del año 2022

A diferencia del modelo multiclase, no es posible la evaluación directa comparando con el código de producto, por lo que es necesario otra metodología de medición. En este caso, se opta por evaluar manualmente una muestra del set de datos con las predicciones del modelo. Considerando que el total de productos evaluados son cercanos a 16.000, es necesario acotar a un subconjunto menor. Para esto, se decide limitar a un número de muestras aleatorias por tipo de producto, usando como criterio usar la que presente el menor número de datos en el año 2022. Como se puede ver en la Figura 5.5 del capítulo 5.4.2, el menor número es 33. Con

este cambio, el subconjunto queda con aproximadamente 500 datos que deben ser evaluados.

Se puede apreciar de la Figura 6.4 que el modelo con un 85 % de puntaje F1-score macro average solo disminuyó un 5 % con respecto a lo evaluado en los datos de prueba. Por lo que se puede considerar que generaliza correctamente con datos nuevos.



	precision	recall	f1-score	support
0	0.82	0.91	0.86	255
1	0.89	0.79	0.83	240
accuracy			0.85	495
macro avg	0.85	0.85	0.85	495
weighted avg	0.85	0.85	0.85	495

Figura 6.4: Métricas de desempeño de modelo binario sobre set de datos de 2022.

Nombre Producto	**Tipo**	**F1-score**
Análisis de sistemas	Servicio	81 %
Centro de servicios de datos	Servicio	67 %
Computadores de escritorio	Producto	89 %
Conectores o concentradores de bus serie universal	Producto	76 %
Diseño de base de datos	Servicio	29 %
Implantación de aplicaciones	Servicio	76 %
Impresoras de matriz de puntos	Producto	100 %
Mantenimiento y soporte de hardware	Servicio	35 %
Notebook, laptop o computador portátil	Producto	93 %
Paquetes de software para oficinas	Producto	96 %
Proveedores de servicio de Internet (ISP)	Servicio	97 %
Servicios de diseño de sitio Web	Servicio	80 %
Software logística de planificación	Producto	93 %
Teléfonos fijos	Producto	97 %
Teléfonos para usos especiales	Producto	83 %

Tabla 6.3: F1-score por cada tipo de producto.

Además de lo anterior, analizando por separados los resultados por cada tipo de producto, se puede observar en la Tabla 6.3 que el modelo tiene un mejor rendimiento con los productos que con los servicios. El promedio de los F1-score de los productos es de 92 %, mientras que en los servicios es de un 76 %.

Dado el comportamiento de este modelo, se considera apto para ser utilizado en un servicio web. Los detalles de esta implementación se encuentran en el Capítulo 7.

6.5. Post procesamiento de modelo para identificar casos forzados

Para poder identificar los casos en los que el modelo está potencialmente forzando un código de producto, se realiza un post procesamiento sobre las predicciones obtenidas. En concreto, se busca encontrar los casos en los que el modelo identifica una baja probabilidad de precisión para todos los tipos de producto. Estos casos serían renombrados como "desconocidos", permitiendo al usuario descartar dichas predicciones.

Para esto, sobre los datos predichos se desarrolla un proceso que revise la confianza en la predicción sobre cada una de las clases. En caso de que ninguno cumpla con una probabilidad de confianza mínima determinada por el usuario, se determinará como “desconocido”.

Para evaluar el impacto de este post procesamiento, se realiza el gráfico de la Figura 6.5 que nos permite visualizar cómo aumenta el porcentaje de etiquetas desconocidas sobre el total de datos, cuando se aumenta el porcentaje mínimo de precisión requerida para considerar a un código de producto como correcto.

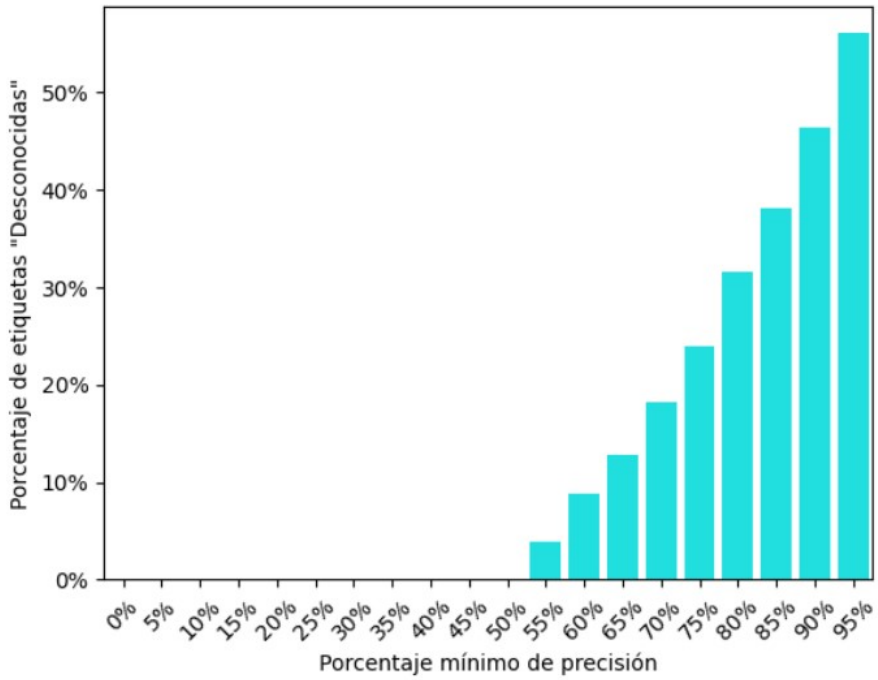


Figura 6.5: Gráfico de porcentaje de casos desconocidos por porcentaje de precisión en datos de entrenamiento.

Es posible ver en el gráfico de la Figura 6.5, que manteniendo la precisión mínima requerida hasta aproximadamente un 70 %, se pierden alrededor de un máximo de 18 % de las predicciones totales del modelo. Este mismo ejercicio se realizó sobre los datos de evaluación del año 2022, como se muestra en el siguiente gráfico de la Figura 6.6

A diferencia del caso anterior, es posible distinguir que delimitando un mínimo de probabilidad de confianza de un 70 %, perdemos aproximadamente un 40 % de todas las predicciones. Este gran cambio indica lo sensible que es ante nuevos datos.

6.6. Conclusiones del modelo binario

Las conclusiones del desarrollo del modelo binario se indican a continuación:

- El modelo con aproximadamente cinco mil datos resultó con una calidad aceptable para

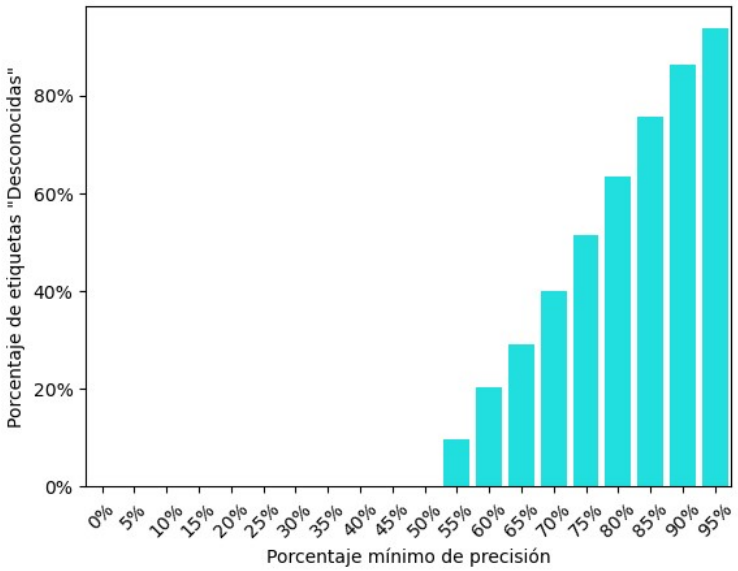


Figura 6.6: Gráfico de porcentaje de casos desconocidos por porcentaje de precisión en datos de entrenamiento.

solucionar el problema, al obtener un F1-score macro average de un 90 % en los datos de entrenamiento y 85 % en los de testing.

- El modelo binario requiere de menos datos que el modelo multiclase, ya que no es necesario entrenar clases adicionales a las que se quieren abarcar.
- La predicción del modelo en los servicios es de menor calidad que en los productos.
- La evaluación de este modelo resulta costosa, ya que no es automática y requiere de inspección manual de las predicciones.

Capítulo 7

Servicio web para la utilización del modelo por usuarios

El servicio web (SW) a utilizar tiene como objetivo el permitir a usuarios externos la utilización del modelo binario descrito en el Capítulo 6. Se espera que tenga las siguientes características:

- Estar siempre disponible para el usuario, sin depender de la disponibilidad del desarrollador.
- No suponer costo monetario para el usuario ni para el desarrollador.
- Debe ser sencillo en su utilización. Es decir, el usuario no debe codificar ni generar cambios a la aplicación.

Dado lo anterior, el SW utilizado que cumple con estas características es el de Hugging Face Spaces (HFS), el cual está diseñado para subir prototipos de aplicaciones enfocadas en modelos de aprendizaje de máquina.

7.1. Desarrollo de aplicación para HFS

La utilización y configuración del SW conlleva los siguientes pasos ilustrados en la Figura 7.1:

Figura 7.1: Ilustración de flujo de solución. Uso de Logo oficial de Hugging Face.

1. El modelo binario una vez entrenado, se exporta en un archivo pickle que permite su reutilización por otros códigos.
2. Se desarrolla una aplicación que permita tener como input un archivo zip provenientes de descargas masivas de CCPP, aplique el modelo entrenado sobre los datos y genere un archivo CSV de salida con los resultados. Además de lo anterior, se agrega una interfaz web generada con la librería "Gradio", que es compatible con HFS.
3. Se suben los datos de la aplicación, los requerimientos de librerías y el modelo a HFS. Una vez subidos, la aplicación se genera automáticamente por el servidor.

Tanto la aplicación como el código que la genera se encuentran en el siguiente link: `https://huggingface.co/spaces/HugoGallardoR/ws_tesis_ccpp`. Además, el código se encuentra respaldado en GitHub dentro de la carpeta "app".

7.2. Interacción del usuario con SW

A continuación, se muestra en la Figura 7.2 una imagen del SW para facilitar la comprensión de los pasos que debe seguir el usuario:

Figura 7.2: Ilustración de Servicio Web en Hugging Face Spaces.

El SW tiene la siguiente interacción con el usuario:

1. El usuario debe descargar desde el portal de datos abiertos de ChileCompra el archivo mensual en formato ZIP que desee aplicar el modelo.
2. El usuario debe subir el archivo dentro de la sección izquierda. Luego presionar el botón "Enviar".

3. Una vez el modelo se termine de ejecutar, aparecerá el ícono del archivo en la sección derecha. El usuario debe presionar el ícono para descargar el archivo en formato CSV, con los productos etiquetados por el modelo.

Consideraciones para el usuario:

- Dado que en general el tamaño de los documentos mensuales es grande, y se utiliza la versión gratuita de HFS, el tiempo de procesamiento puede llegar a 5 minutos por cada archivo.
- En caso de que la aplicación no se utilice en 72 horas se baja automáticamente por inactividad. Sin embargo, se reactiva nuevamente cuando un usuario ingresa. Esto supone como impacto, que se debe esperar un tiempo adicional a que la aplicación se vuelva a generar para ser usada.

7.3. Evaluación de tiempos de proceso

En esta sección se evaluarán los tiempos de respuesta del servicio web tras su utilización como se indica en la Sección 7.2. Para esto, se genera un archivo con todas las transacciones entre el año 2017 a 2022, de los 15 tipos de productos con los que el modelo fue entrenado. El archivo resultante cuenta con aproximadamente 100.000 transacciones, pesando 150 megabytes en formato CSV y 25 megabytes comprimido en formato ZIP.

La primera prueba realizada no fue exitosa, ya que la versión gratuita de HFS rechaza inputs pesados en caso de contar con pocos recursos. Dado esto, se procede a dividir el archivo en dos partes iguales con aproximadamente 50.000 transacciones seleccionadas de forma aleatoria. Esta segunda prueba resultó exitosa, tomando tiempos de proceso cercanos a 35 segundos en ambos. Se debe tener en cuenta que los tiempos de proceso pueden variar dependiendo de la capacidad del servicio de HFS.

Considerando lo anterior, el proceso demora aproximadamente 35 segundos en evaluar 50.000 transacciones. Si esto lo comparamos con el tiempo estimado de evaluación manual indicada en la Sección 1.1, que es de 50 por hora, podemos dar cuenta de una disminución en tiempo a menos de medio segundo por la misma cantidad de transacciones.

7.4. Evaluación del Observatorio del Gasto Fiscal

El proceso desarrollado en esta tesis se presentó al OGF para contar con su evaluación en términos generales. Dentro de lo revisado consideran lo siguiente:

- Desempeño general del modelo: Se considera que el modelo logra un desempeño aceptable a pesar de la baja cantidad de datos que posee. Se considera que la metodología es apropiada para resolver el problema con las circunstancias mencionadas anteriormente.

- Reproducibilidad: El proceso completo se encuentra documentado tanto en el documento de tesis como en un repositorio de GitHub. Esto permite la total reproducibilidad para ser utilizado en otros proyectos.
- Facilidad de uso: El servicio web desarrollado permite la rápida utilización del modelo para los usuarios. Esto agiliza el poder reentrenar el modelo mediante etiquetado manual.

Cabe destacar que al momento de presentado el desarrollo del prototipo, la motivación inicial de ser utilizado como un validador de CCPP de municipalidades ya no es prioridad para la ONG. Sin embargo, dado que el modelo está desarrollado de manera general para cualquier tipo de producto, se pretende utilizar como base para un proyecto futuro de análisis de CCPP en el área de la salud.

Capítulo 8

Conclusiones

A partir del análisis introductorio de los datos de CCPP, se desprende que éstos poseen una baja calidad en la descripción de texto libre y código de productos. Por lo que se recomienda para desarrollar modelos predictivos sobre estos datos el no utilizarlos en su forma original, sino tras un preprocesamiento de éstos.

El modelo multiclase desarrollado para predecir la etiqueta de un producto basado en su descripción, presentó un buen comportamiento en los datos de entrenamiento iniciales con un F1-score macro average de 92 %. Sin embargo, no fue capaz de generalizar correctamente al utilizar los datos de evaluación disminuyendo a un 49 % de esta misma métrica. La utilización de una clase desconocida en el post procesamiento del modelo multiclase si bien reduce la incertidumbre de los resultados del modelo, no permite cumplir con el objetivo planteado al reducir drásticamente el subconjunto de datos etiquetados.

El modelo binario desarrollado para identificar la concordancia entre la descripción y código de producto presentó un comportamiento estable entre los datos de entrenamiento y de testing, mostrando un F1-score macro average de 90 % y 85 % respectivamente. Dado este resultado, podemos concluir que este es un modelo viable para utilizar incluso con una baja cantidad de datos disponibles como el de este ejercicio. A partir de los resultados obtenidos sobre los datos de testing del modelo binario, fue posible ver que el comportamiento en la predicción de productos es mayor a la de los servicios, obteniendo un F1-score macro average de 92 % y 76 % respectivamente.

El modelo binario llevado al servicio web gratuito de HFS tuvo un resultado satisfactorio, permitiendo correctamente a cualquier usuario externo la utilización del modelo sobre nuevos conjuntos de datos. Además, permite evaluar aproximadamente 50.000 transacciones en 35 segundos, lo que es una sustantiva mejora comparada con la evaluación manual de 50 transacciones por hora. Este prototipo desarrollado permite a OGF generar y utilizar modelos de clasificación de mejor manera para futuros análisis relacionados con las CCPP.

8.1. Limitaciones y recomendaciones para trabajo futuro

La principal limitación para el desarrollo de modelos predictivos fue la cantidad de los datos de entrada, ya que dada su mala calidad, solo se pudieron ingresar los que el tesista pudo evaluar manualmente. Esto llevó a no solo a una baja en el desempeño de los modelos, sino que a descartar modelos más avanzados que requieren mayor cantidad de datos, tales como las redes neuronales.

Se recomienda para un trabajo futuro sobre esta misma temática, la reutilización del proceso generado en esta tesis, pero abarcando una mayor cantidad de datos, tipos de productos y técnicas de preprocesamiento. Además, en caso de que el número de usuarios aumente considerablemente, se recomienda utilizar la versión de pago de HFS u otra alternativa. Adicionalmente, el servicio web podría facilitar el ingreso de datos etiquetados por los usuarios al entrenamiento del modelo.

El tema que abarca esta tesis también fue expuesta a analistas de estudios de la DCCP. Ellos mostraron interés en la utilización de este modelo para la validación de transacciones, con el fin de mejorar la calidad de datos de entrada en el desarrollo de un modelo de riesgos a futuro. Si buen, no se profundizó en las conversaciones, se recomienda avanzar en este tipo de trabajos con ellos.

Bibliografía

[1] Charu C. Aggarwal. *Data Mining: The Textbook*. Springer Cham, New York, 1. ed edition, 2015.

[2] Yassine Al-Amrani, Mohamed Lazaar, and Kamal Eddine El Kadiri. Random forest and support vector machine based hybrid approach to sentiment analysis. *Procedia Computer Science*, 127, 03 2018.

[3] Adi Almajid. Multilayer perceptron optimization on imbalanced data using svm-smote and one-hot encoding for credit card default prediction. *Journal of Advances in Information Systems and Technology*, 3(2):67–74, Sep. 2022.

[4] Trstenjak Bruno, Mikac Sasa, and Donko Dzenana. Knn with tf-idf based framework for text categorization. *Procedia Engineering*, 69:1356–1364, 2014. 24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013.

[5] Zhuo Chen et al. The lao text classification method based on knn. *Elsevier*, 2020.

[6] Kou Gang, Yang Pei, Peng Yi, Xiao Feng, Chen Yang, and Alsaadi Fawaz E. Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods. *Applied Soft Computing*, 86:105836, 2020.

[7] Yaakov HaCohen-Kerner, Daniel Miller, and Yair Yigal. The influence of preprocessing on text classification using a bag-of-words representation. *PLOS ONE*, 15:e0232525, 05 2020.

[8] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.

[9] Bahzad Jijo and Adnan Mohsin Abdulazeez. Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, 2, 01 2021.

[10] Sang-Woon Kim and Joon-Min Gil. Research paper classification systems based on tf-idf and lda schemes. *Human-centric Computing and Information Sciences*, 2019.

[11] Vijay Kotu and Bala Deshpande. *Data Science: Concepts and Practice Data Science*. Morgan Kaufmann, New York, second edition edition, 2015.

[12] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. *Information*, 10(4), 2019.

[13] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 957–966, Lille, France, 07–09 Jul 2015. PMLR.

[14] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[15] F. Parra. Estadística y machine learning con r. España, 2019. Licencia Creative Commons Attribution-NonCommercial CC BY-NC. `https://creativecommons.org/licenses/by-nc/4.0/`.

[16] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830, nov 2011.

[17] M.T. Pilehvar and J. Camacho-Collados. *Embeddings in Natural Language Processing: Theory and Advances in Vector Representations of Meaning*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2020.

[18] Nitin Rajvanshi and Prof Chowdhary. Comparison of svm and naïve bayes text classification algorithms using weka. *International Journal of Engineering Research and Technology (IJERT)*, V6, 09 2017.

[19] Khan Sajid, Anwar Mehmoon, Qayyum Huma, Ali Farooq, and Nawaz Marriam. Fake news classification using machine learning: Count vectorizer and support vector machine. *Journal of Computing and Biomedical Informatics*, 4(01), Dec. 2022.

[20] Dr. B. Srinivasu Shugufta Fatima. Text document categorization using support vector machine. *International Research Journal of Engineering and Technology (IRJET)*, 2017.

[21] Kanae Takahashi, Kouji Yamamoto, Aya Kuchiba, and Tatsuki Koyama. Confidence interval for micro-averaged f1 and macro-averaged f1 scores. *Springer*, 2021.

[22] M. Thangaraj and M Sivakami. Text classification techniques: A literature review. *Interdisciplinary Journal of Information, Knowledge, and Management*, 01 2018.

[23] Turki Turki and Sanjiban Sekhar Roy. Novel hate speech detection using word cloud visualization and ensemble learning coupled with count vectorizer. *Applied Sciences*, 12(13), 2022.

[24] Ye Wang, Zhi Zhou, Shan Jin, Debin Liu, and Mi Lu. Comparisons and selections of features and classifiers for short text classification. *IOP Conference Series: Materials Science and Engineering*, 261:012018, 10 2017.