Inteligencia de negocios

Proyecto 1 – Etapa 1

Juan José Montenegro 202012725

Juan Pablo Junco 201912957

Cristian David Caro 202011710

[1]*Departamento de Ingeniería de Sistemas y Computación, Universidad de los Andes*

1 de abril de 2024

Entendimiento del negocio y enfoque analítico

- **Presente en la wiki**

Entendimiento y preparación de los datos.

- **Análisis de los caracteres que no tienen representación ASCII:**

  Se quieren eliminar caracteres que no tengan representación en ASCII, sin embargo, en el español existen caracteres que son importantes en el lenguaje y no tienen representación ASCII. Por ello, ciertos caracteres deben mantenerse por ejemplo, las vocales tildadas y la 'ñ'.
  En nuestro análisis se encontraron vocales con acentuaciones que no son propiamente del español y uso de emojis. Estos caracteres se tuvieron que eliminar ya que no representan ningún valor al modelo ni el fin que persigue la compañía que es analizar y predecir reseñas en español.
  Se usa la librería "re" para operaciones de expresiones regulares en Python. En este caso se usaron para realizar operaciones de búsqueda y reemplazo de patrones en cadenas de texto.
- **Análisis de la puntuación**

  A la hora de tokenizar cada carácter se separa y se toma como una palabra. Esto desde luego, no aporta en nada al modelo. Por esta razón la puntuación se eliminó. Se usa la librería "re" para reemplazar patrones en cadenas de texto y, pandas con el método applymap.

- **Analizar el tamaño de los diferentes reviews y la palabra más repetida**

  Esperábamos que al eliminar la palabra más repetida, pero en su mayoría son wtopwords, entonces se eliminarán posteriormente. Se usa pandas para aplicar applymap al dataframe y buscar con la función lambda la palabra más repetida.
- **Evaluar la completitud**

  Revisar que no hayan valores nulos
- **Evaluar la unicidad**

  Identificar cuantas filas repetidas hay de acuerdo con todas las columnas
- **Evaluar la consistencia**

  Se busca determinar que los reviews solo estén en español, de lo contrario, se eliminan dichos reviews. Se usa la librería langdetect con el modulo DetectoryFactory.
- **Evaluar la validez:**

  Los valores se las reviews tienen sentido para el negocio

- **Proceso de normalización**

  Se realiza la conversión de todo el texto a minúsculas para mantener la uniformidad. Se usa la librería pandas para el método applymap.
- **Transformaciones**
  - Lematizar para pasar las palabras a su significado base. Se usa la librería spacy para cargar el modelo que tiene un tokenizador, etiquetador, analizador sintáctico, reconocedor de entidades nombradas y vectores de palabras.
  - Remover stop words (artículos, pronombres y conjunciones): Se usa la librería nltk para cargar los stopwords del español.
- **Tokenización**
  - Se dividen en unigramas cada uno de los reviews. Se usa la librería nltk con los módulos word_tokenize, sent_tokenize.

Modelado y evaluación

Algoritmo 1 – Logistic Regression

Explicación:

La regresión logística es un modelo estadístico utilizado principalmente para la clasificación binaria, aunque puede extenderse a clasificaciones multiclase. Su funcionamiento se basa en estimar probabilidades utilizando la función logística, también conocida como *sigmoide*, que transforma cualquier valor numérico de entrada en un valor entre 0 y 1. Esta característica la hace ideal para predecir la probabilidad de que una observación pertenezca a una de las dos clases en un problema de clasificación.

En el contexto de la regresión logística, la variable dependiente (la que se quiere predecir) es categórica y representa la clase o categoría a la que pertenece cada observación. Las variables independientes (o predictores) pueden ser tanto numéricas como categóricas y se utilizan para predecir la clase de la variable dependiente.

El modelo trabaja calculando un conjunto de coeficientes, uno para cada variable independiente, mediante el método de máxima verosimilitud. Estos coeficientes determinan la importancia y el tipo de relación (positiva o negativa) de cada variable independiente con la probabilidad de que la observación pertenezca a la clase de interés.

La probabilidad estimada se obtiene introduciendo los valores de las variables independientes y los coeficientes calculados en la función logística. Si esta probabilidad es mayor a un umbral específico, generalmente 0.5, la observación se clasifica en la clase de interés; de lo contrario, se clasifica en la otra clase.

Implementación:

```
tfidf_vectorizer = TfidfVectorizer(min_df=0.01, max_df=0.5, ngram_range=(1, 3),sublinear_tf=True)
```

Para la implementación de este algoritmo se debe comenzar vectorizando (en este caso se usó **TfidfVectorizer**) el cual transforma el texto tokenizado en características numéricas. Configurando *min_df=0.01* y *max_df=0.5*, se filtran términos extremadamente raros o frecuentes, y con *ngram_range=(1, 3)*, se capturan desde palabras individuales hasta trigramas, captando así mejor el contexto. La opción *sublinear_tf* aplica una transformación logarítmica a las frecuencias de los términos, ayudando a manejar disparidades en las cuentas de términos y mejorando la relevancia de las características para la clasificación.

```
log_reg = LogisticRegression(max_iter=1000,class_weight='balanced' )
```

Después, se configura el modelo para iterar hasta 1000 veces en la búsqueda de la convergencia, lo cual es útil para asegurar que el modelo tenga suficientes oportunidades de aprender de los datos, especialmente en casos donde la convergencia es difícil de alcanzar. El parámetro *class_weight* ajusta automáticamente los pesos inversamente proporcionales a las frecuencias de clase en los datos de entrada, lo cual es crucial para tratar con conjuntos de datos desbalanceados, asegurando que el modelo no esté sesgado hacia las clases más frecuentes. Esta configuración ayuda a mejorar la equidad del modelo en la clasificación de observaciones de clases minoritarias, potencialmente mejorando la precisión y exhaustividad para todas las clases en el conjunto de datos. Tras la inicialización, el modelo se entrena, ajustándose para minimizar el error en la predicción de las clases objetivo, aprendiendo así la relación entre las características de los textos y las categorías a las que pertenecen.

Resultados:

Para analizar el resultado del algoritmos decidimos ilustrarlo en una matriz de confusión:

	precision	recall	f1-score	support
1	0.43	0.58	0.50	151
2	0.37	0.37	0.37	231
3	0.37	0.36	0.36	281
4	0.46	0.45	0.46	401
5	0.67	0.63	0.65	495
accuracy			0.49	1559

Con una precisión general del **49.33%**, el modelo muestra **un rendimiento moderado** en la clasificación de las reseñas en las 5 clases indicadas.

- **La clase 1** muestra una precisión del 43% y un recall del 58%, resultando en un F1-score de 0.50. Este alto recall indica que el modelo es relativamente bueno identificando esta clase, aunque la precisión sugiere que muchas reseñas no pertenecientes a la clase 1 son incorrectamente clasificadas como tales.
- **La clase 2 y clase 3** tienen puntuaciones más bajas en todas las métricas, con precisión y recall alrededor del 37%, lo que indica una dificultad del modelo para distinguir correctamente estas categorías.
- **La clase 4** mejora ligeramente con una precisión del 46% y un recall del 45%, lo que da como resultado F1-score de 0.46, mostrando un balance más equitativo entre precisión y recall en comparación con las clases más bajas.
- **La clase 5** tiene la mejor actuación con una precisión del 67% y un recall del 63%, alcanzando un F1-score de 0.65. Esto indica que el modelo es bastante eficaz en identificar correctamente las reseñas de esta clase.

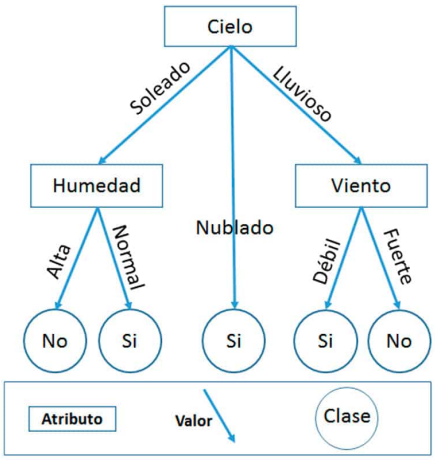
El modelo tiene mucho espacio para ser mejorado antes de ser recomendado completamente. Se llevaron a cabo diferentes técnicas como ajuste en el pre procesamiento dejando de lado las palabras más repetidas y con menor significado. Sin embargo, obtuvimos como resultado que al dejar de lado estas palabras el modelo no mejoraba si no por el contrario disminuía su precisión. Por último, el ajuste de parámetros en el vectorizador y el modelo se realizó hasta obtener el mejor resultado.

Algoritmo 2 – Random Forest

Explicación del algoritmo

Árbol de decisión

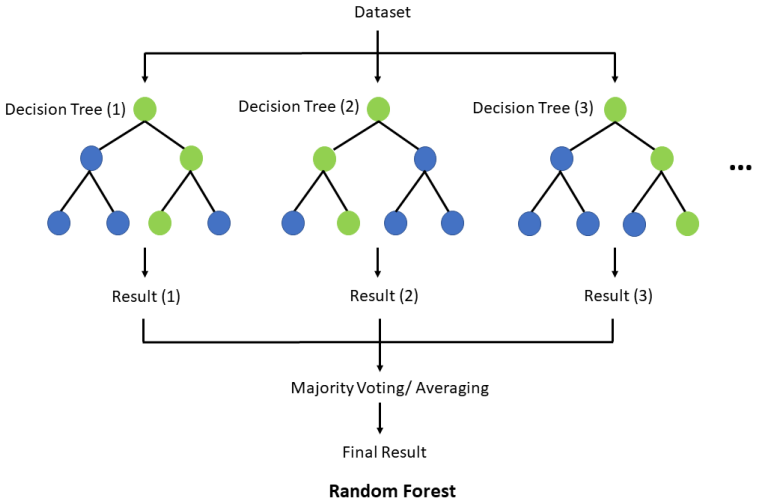
Un árbol de decisión es un modelo que toma decisiones basadas en una estructura de árbol. En este árbol, cada nodo interno representa una pregunta sobre una característica del conjunto de datos, y cada rama representa una posible respuesta a esa pregunta



Existen métricas asociadas a los árboles de decisión. Estas métricas me permiten evaluar qué tan buenas decisiones toma mi árbol.

- Precisión: Una alta precisión indica que el modelo tiene pocos falsos positivos, es decir, pocos ejemplos que fueron clasificados incorrectamente como positivos.
- Recall: Un alto recall indica que el modelo tiene pocos falsos negativos, es decir, pocos ejemplos que fueron clasificados incorrectamente como negativos.
- Puntaje F1 (F1 Score): El puntaje F1 es una medida de precisión y recall combinadas. El puntaje F1 alcanza su mejor valor en 1 (perfecta precisión y recall) y su peor valor en 0.

Random Forest



Se basa en la idea de crear múltiples árboles de decisión durante el entrenamiento y luego combinar sus predicciones para obtener un resultado final más robusto y preciso. Cada árbol de decisión en el bosque se entrena con una muestra aleatoria del conjunto de datos original y utiliza una selección aleatoria de características para tomar decisiones.

Luego, las predicciones de cada árbol se promedian o se votan para producir una predicción final.

Ejecución del algoritmo:

El algoritmo en su forma base tiene unos resultados de:

	precision	recall	f1-score	support
1	0.52	0.30	0.38	151
2	0.35	0.32	0.33	231
3	0.33	0.31	0.32	281
4	0.40	0.36	0.38	401
5	0.55	0.72	0.62	495
accuracy			0.45	1559
macro avg	0.43	0.40	0.41	1559
weighted avg	0.44	0.45	0.44	1559

Lugo a través de la estimación de los hiperparámetros, se busca que el modelo mejore y sea más preciso. Para ello, se evalúa el rango de los hiperpárametros (notebook).

- Máxima profundidad de los árboles: Se puede decir que una mayor profundidad conduce a tener una mejor precisión.
- Número de estimadores: Un número de estimadores de 100 o 500 podría ser la mejor opción.
- Número de hojas mínimas de los árboles: Un menor número de hojas de cada árbol puede conducir a un mejor resultado.
- Número de muestras a ejecutar: Un número de ejemplos cercanos a 100 brinda en teoría una mejor precisión.

Palabras más frecuentes:

Clase 1: hotel, poder y sucio. Clase 2: hotel, habitación y servicio. Clase 3: buen, lugar y bien. Clase 4: buen, lugar y poder. Clase 5: excelente, servicio y lugar.

Al definir el modelo de nuevo usando los hiperparámetros definidos, se espera que el resultado mejore. A continuación, el resultado del modelo:

	precision	recall	f1-score	support
1	0.64	0.14	0.23	151
2	0.40	0.35	0.37	231
3	0.31	0.19	0.24	281
4	0.37	0.24	0.29	401
5	0.48	0.85	0.61	495
accuracy			0.43	1559
macro avg	0.44	0.35	0.35	1559
weighted avg	0.42	0.43	0.39	1559

Modelo ajustado con hiperparámetros

Como conclusión, el modelo ajustado con los hiperparámetros no mejora las métricas de precisión, recall ni f1_score. Por tanto, se toma el modelo inicial debido a que dio mejores resultados. Estos resultados son:

	precision	recall	f1-score	support
1	0.52	0.30	0.38	151
2	0.35	0.32	0.33	231
3	0.33	0.31	0.32	281
4	0.40	0.36	0.38	401
5	0.55	0.72	0.62	495
accuracy			0.45	1559
macro avg	0.43	0.40	0.41	1559
weighted avg	0.44	0.45	0.44	1559

Modelo inicial de random forest

Del primer modelo se puede decir que para cada uno de los conjuntos por clasificar:

Grupo1: Al tener una precisión algo alta aunque bajo recall y f1_score, se puede decir que el modelo está prediciendo correctamente la clase positiva en la mayoría de los casos, pero está perdiendo muchos casos positivos que debería haber identificado.

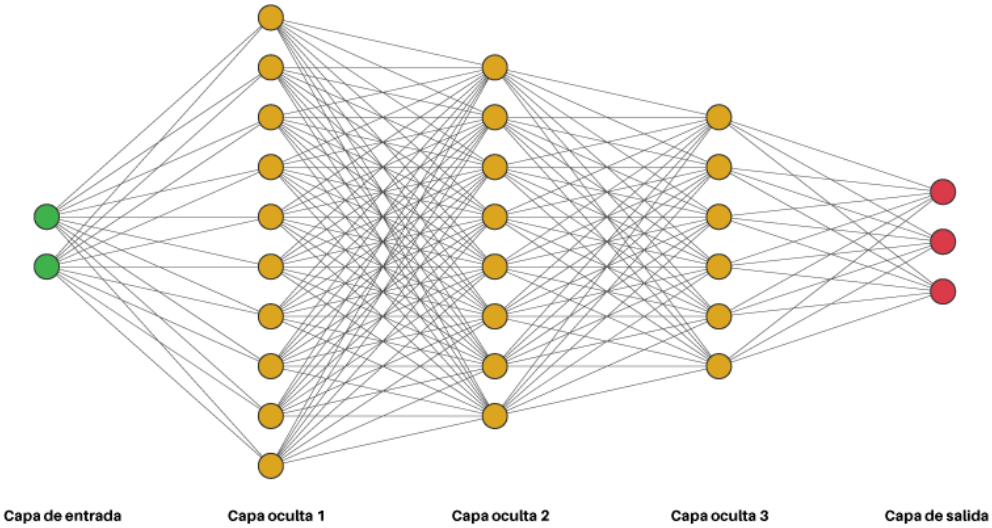
Grupo2,3,4: Debido a que todas sus métricas son bajas, se puede concluir que el modelo está teniendo dificultades para realizar predicciones precisas en general y está perdiendo tanto instancias positivas como negativas de manera significativa.

Grupo5: En este caso al haber una precisión algo alta y recall alto, se puede pensar que todo está bien. Sin embargo, el f1_score es bajo, esto significa que el modelo está favoreciendo una clase sobre la otra y puede haber un desequilibrio entre precisión y recall.

En general, para turismo de los Alpes, el algoritmo se random forest solo puede garantizar una certeza ligera el su modelo de predicción para poder determinar la calificación de los reviews que reciben una calificación de 1 o 5.

Algoritmo 3 – Neural Networks

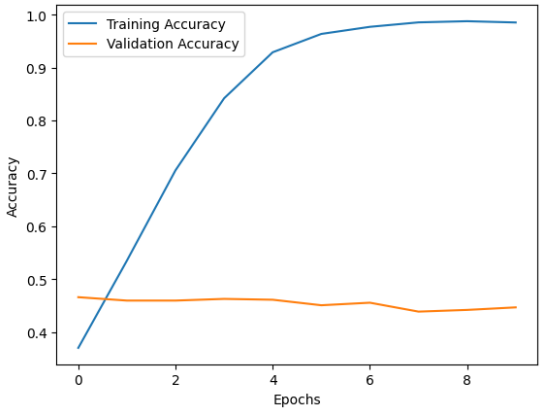
El algoritmo de redes neuronales se funciona con una arquitectura de capas de neuronas. En cada capa se asigna un peso dependiendo de los datos de entrenamiento. De esta manera, el algoritmo logra reconocer patrones en los vectores y clasificar cada uno de los datos a predecir.



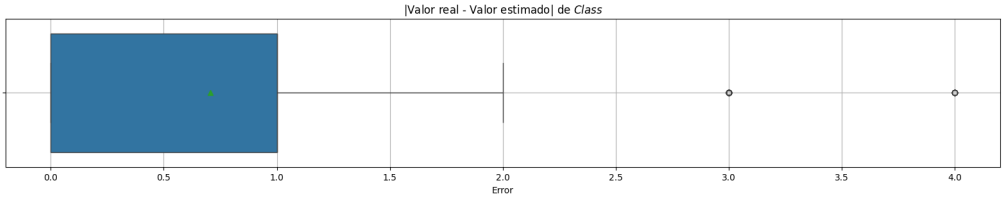
Para la definición del modelo para Turismo de los Alpes se definió cuatro capas densas (grafo conectado) que reduce la cantidad de nodos a la mitad en el avance de las capas. También se agregó dos capas Dropout que ayuda al overfitting desactivando la mitad de las neuronas seleccionadas aleatoriamente en cada iteración del entrenamiento.

Para el entrenamiento de la red se va a hacer uno de la función de perdida sparse_categorical_crossentropy debido a que es un problema de clasificación de etiquetas enteras y el optimizador adam para reducir el tiempo de entrenamiento.

Se obtuvo una precisión de 0.45 en las predicciones. También se puede observar que el modelo es mejor detectando reviews muy buenas o muy malas ya que la precisión en la clasificación de 1 y 5 son mayores que las de calificaciones intermedias. Con respecto al f1-score también podemos observar la dificultad que se le presenta al modelo sobre todo con las calificaciones de 2 y 3.



En esta gráfica podemos ver que en el paso de las iteraciones de entrenamiento el modelo fue mejorando con respecto al modelo de entrenamiento. Pero en el caso del conjunto de prueba no mejoró mucho y se mantuvo constante desde la primera.



Aquí se puede observar que la mayoría de los datos predichos tienen un error de 1 y hay muy poco error con un margen muy alto de diferencia. En los percentiles del error podemos ver que el 75% de las calificaciones que predijo el modelo tienen un valor máximo de 1. Esto nos dice que el modelo es muy bueno detectando si una review es buena o mala, pero le cuesta detectar la puntuación exacta.

Justificación

Logistic Regression: Se implementó este algoritmo ya que la regresión logística es ampliamente utilizada debido a su simplicidad, eficiencia y la interpretabilidad de sus resultados. Permite no solo clasificar nuevas observaciones, sino también entender la importancia relativa de las variables independientes en la predicción de la clase. Además, a través de los coeficientes estimados, es posible determinar cómo afecta cada variable independiente a la log-odds de pertenecer a la clase de interés, proporcionando así valiosos insights sobre los datos analizados.

Random Forest: Se uso este algoritmo debido a que es muy robusto y preciso. Las técnicas de usar muestras de datos aleatorias para cada árbol y al final, promediar o escoger el mejor valor, permiten reducir el sesgo y la varianza del modelo, lo que generalmente resulta en un mejor rendimiento predictivo. Además, reduce la tendencia de los árboles individuales a sobreajustarse al conjunto de datos de entrenamiento. Esto significa que Random Forest tiende a generalizar bien a datos no vistos y a evitar el sobreajuste.

Neural Networks: Se escogió este algoritmo debido a su capacidad para capturar patrones complejos en los datos, su flexibilidad para aprender representaciones de características automáticamente, su escalabilidad con conjuntos de datos grandes, su historial de buen desempeño en tareas de clasificación de texto y la disponibilidad de herramientas y bibliotecas que facilitan su implementación. Sin embargo, se debe tener en cuenta que el uso de redes neuronales también puede presentar desafíos, como la necesidad de grandes conjuntos de datos y el riesgo de sobreajuste, por lo que la elección debe basarse en las características específicas del problema y los recursos disponibles.

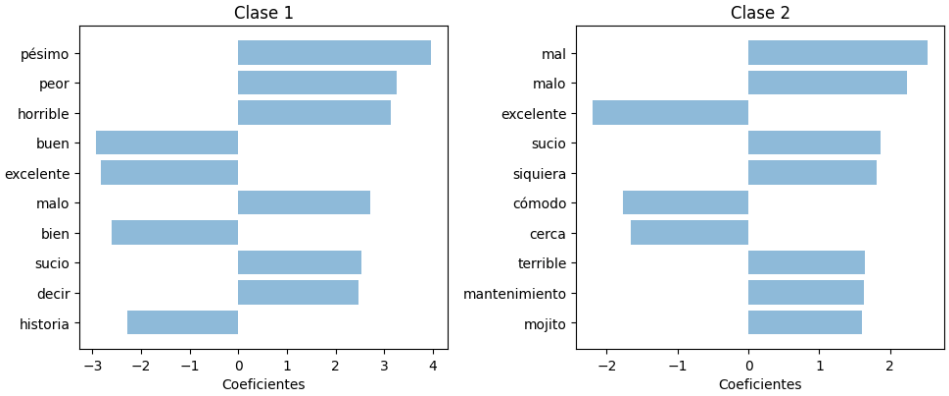
Resultados

Logistic Regression

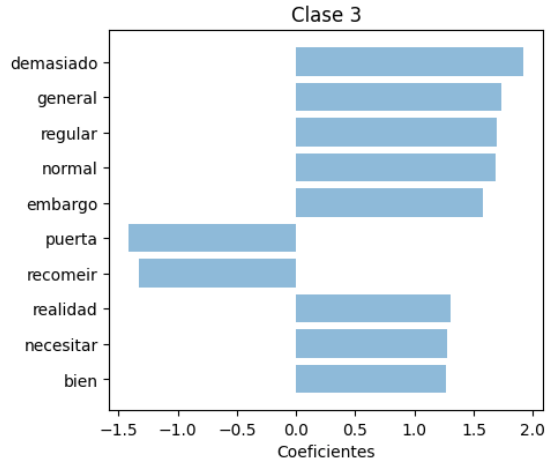
Turismo de los Alpes se puede beneficiar del modelo implmentado con logistic regression ya que presenta una precisión del 49.33%. Aunque se recomienda

primeramente entrenar el modelo con mayor datos (Se espera una mejor métrica de precisión).

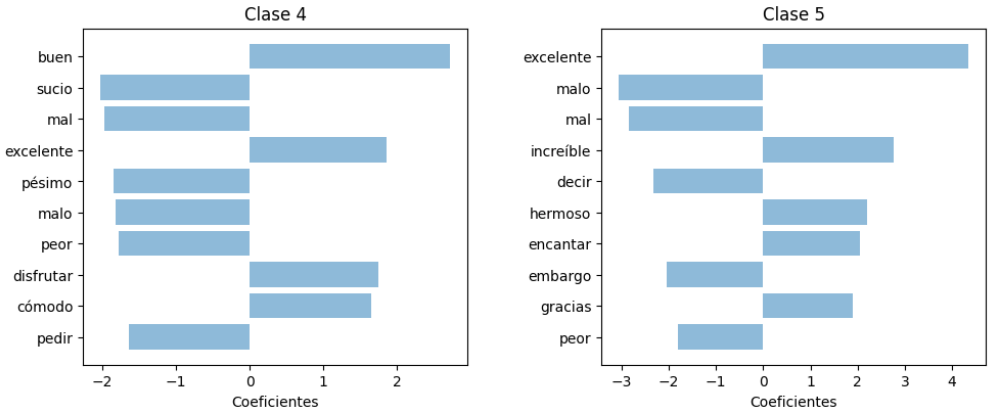
Para que el negocio pueda dar dé cuenta de que se está hablando y porque le puede beneficiar este modelo nos remitimos a los siguientes gráficos que nos indican la distribución de la influencia de las palabras a la hora de clasificarlas en una clase (1,2,3,4,5).



Para las clases 1 y 2 podemos dar de cuenta que los clientes que están teniendo una pésima experiencia, en sus reseñas habitualmente surge el tema de la limpieza y posiblemente el servicio no solo de acomodación pero de consumibles, ya que se resaltan palabras como ‘sucio’, ‘pésimo’, ‘horrible’, ‘mantenimiento’, ‘mojito’.



En la clase 3 identificamos que se encuentran los clientes que no tienen mayor queja y solo expresan conformismo. Por palabras como ‘general’, ‘regular’, ‘normal’ y ‘bien’. Sin embargo, no podemos ofrecer un análisis más profundo de porque los clientes tienen esta opinión. (Se recomendaría alimentar el modelo con un dataset entrenado más grande para obtener mejores resultados sobre esta clase).



Por último, en las clases 4 y 5 se encuentran los clientes más satisfechos que según los datos se debe a una experiencia cómoda y encantadora. Puede deberse a un buen servicio y acomodación. Tampoco se puede indagar mucho en las razones de porque los clientes dan una buena calificación más allá de las ya dichas, por lo que se recomienda también alimentar el modelo con más datos entrenados.

Random Forest

En general, para turismo de los Alpes, el algoritmo se random forest solo puede garantizar una certeza ligera el su modelo de predicción para poder determinar la calificación de los reviews que reciben una calificación de 1 o 5.

La oportunidad de mejora de este modelo podría estar en buscar si hay un desequilibrio de clases y tratar de ajustarlo. También, se podría buscar cómo es el error en la clasificación, por ejemplo: si el modelo falla con un valor de calificación de 2 cuando debería ser 5, o si solo se equivoca ligeramente, es decir, cuando la calificación resultante es 2 pero debería ser 3.

Neural Networks

En el modelo de redes neuronales se obtuvo una precisión general de 0.46 y con respecto a las precisiones especificas se obtuvo una mayor precisión en las calificaciones 1 y 5 con precisiones de 0.67 y 0.52 y una menor precisión en las calificaciones de 2, 3 y 4 con precisiones de 0.39, 0.31 y 0.41 respectivamente. Esto nos muestra una clara falencia del modelo para identificar clasificaciones intermedias. Con respecto al recall se pudo identificar un sesgo en el modelo que lo hace tender a calificar todas las reviews un poco negativas como 2 ya que el recall de esta clase es 0.26 y con respecto a las positivas está menos sesgado y es capaz de diferenciar una calificación de 5 y de 4 teniendo unos valores de recall de 0.57 y 0.50.

En general, el modelo realmente es capaz de diferenciar un comentario bueno de uno se obtuvo una precisión general de 0.46 y con respecto a las precisiones especificas se obtuvo una mayor precisión en las calificaciones 1 y 5 con precisiones de 0.52 y 0.67 y una menor precisión en las calificaciones de 2, 3 y 4 con precisiones de 0.39, 0.31 y 0.41 respectivamente. Esto nos muestra una clara falencia del modelo para identificar clasificaciones intermedias. Con respecto al recall se pudo identificar un

sesgo en el modelo que lo hace tender a calificar todas las reviews un poco negativas como 2 ya que el recall de esta clase es 0.26 y con respecto a las positivas está menos sesgado y es capaz de diferenciar una calificación de 5 y de 4 teniendo unos valores de recall de 0.57 y 0.50.malo, pero le cuesta calificarlo con un número exacto.

Mapa de actores relacionado con el producto de datos creado

Rol dentro de la empresa	Tipo de actor	Beneficio	Riesgo
Turistas	Cliente	Tendrían acceso a servicios de mayor calidad y que vayan de la mano con sus deseos y expectativas.	Puede que las características de los hoteles no vayan de la mano con ciertos turistas. Un alto grado de inconformidad afecta la imagen de la empresa de turismo.
Inversionistas	Financiadores	Mejores retornos para quienes se mueven en estos ambientes. Nueva oportunidad de diversificar para otros inversionistas.	Puede que las empresas no sean lo suficientemente competentes y tomen decisiones erróneas frente a variables externas del ambiente (competencia, intervención del Estado o crisis), afectando las metas de la empresa y genere bajas rentabilidades.
Hoteles o agencias de viajes	Proveedor	Pueden recibir mejores ingresos al trabajar con empresas que dan buena experiencia a los clientes.	Si no cumplen con los estándares de calidad o expectativas del cliente. La experiencia del cliente se verá afectada y puede dañar la reputación de la empresa.

Comunidad local	Beneficiado	Podrían beneficiarse al recibir empleos gracias a las inversiones y oportunidades que recibe la empresa.	Si el turismo incrementa mucho, puede haber problemas como: incremento en el precio de la vivienda, incremento del costo de vida y la gentrificación.

Trabajo en equipo

Presente en la wiki