UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERIA
DEPATAMENTO DE INGENIERIA DE SISTEMAS Y COMPUTACIÓN
INTELIGENCIA DE NEGOCIOS

PROYECTO 1- ETAPA 1
Apoyo a la detección de intentos de suicidio

Grupo 4
Daniela Ricaurte - 201822966
Melissa Contreras - 202011876
Julián Mora - 202012747

Profesor
María Del Pilar Villamil

Bogotá, Octubre 19 de 2022

Tabla de Contenidos

a) Contexto de salud mental. Apoyo a la detección de intentos de suicidio a partir de información recolectada de Reddit a nivel de comunidades que sufren de depresión o han intentado suicidarse. Los datos originales los pueden encontrar en este enlace (https://www.kaggle.com/datasets/nikhileswarkomati/suicidewatch?resource=download) . Sin embargo, los datos a utilizar en el proyecto no son los mismo y los encuentra disponibles en la sección proyecto de BloqueNeón curso Unificado de BI.

0. Trabajo en equipo

Nombre	Rol	Tareas	Tiempo asignado	Tiempo dedicado	Algoritmo trabajado
Melissa Contreras	Líder de proyectos y líder de datos	Organización y seguimiento Algoritmo Label Spreading	7 horas	7 horas	Label Spreading
Julián Mora	Líder de negocio	Algoritmo Naive Bayes Enfoque Analítico	7 horas	7 horas	Naive Bayes
Daniela Ricaurte	Líder de analítica	Perfilamiento y Preparación de los datos. Algoritmo SVM	8 horas	8 horas	SVM

Nombre	Retos enfrentados	Formas planteadas para resolverlos
Melissa Contreras	Al momento de vectorizar y crear el modelo predictivo, el notebook retornaba errores distintos.	Probar muchas formas de vectorización, buscar en internet los errores.
Julián Mora	El algoritmo no es difícil de implementar, más generar el modelo fue complicado.	Ver muchos videos hasta encontrar una manera de vectorizar que fue compatible con el proyecto.
Daniela Ricaurte	El algoritmo SVM se demora mucho tiempo en correr al igual que la Lematización y stemming de palabras.	Probar primero con menos datos, para ver si está funcionando correctamente para luego cuando se sabe que todo está bien, usar los datos completos.

Reuniones

Tema por tratar	Fecha
Reunión de lanzamiento y planeación	10 octubre 2022
Reunión de ideación	14 octubre 2022
Reuniones de seguimiento	17 octubre 2022
Reunión de finalización	19 octubre 2022

Repartición de los 100 puntos

- Julián Mora: 33,33 puntos.
- Melissa Contreras: 33,33 puntos.
- Daniela Ricaurte: 33,33 puntos.

Puntos de mejora

Comenzar a hacer los trabajos con mayor antelación, e intentar resolver los problemas en conjunto para así ser más efectivos y aprendemos mucho mejor.

1. Comprensión del negocio y enfoque analítico.

Definición de los objetivos y criterios de éxito desde el punto de vista del negocio.
Determinación del enfoque analítico para alcanzar los objetivos del negocio.
Descripción de cómo el requerimiento de negocio es resuelto por el enfoque analítico propuesto, para lo cual debe diligenciar la tabla que se presenta a continuación:

Oportunidad/problema Negocio	Debido al incremento exponencial de los casos de depresión a lo largo de los años, nos vemos enfrentando una crisis de salud mental, concentrada en personas jóvenes. Esto nos presenta la oportunidad de identificar a aquellos que son más vulnerables, para brindar apoyo oportuno y evitar el suicidio.
Enfoque analítico (Descripción del requerimiento desde el punto de vista de aprendizaje de máquina)	Identificar patrones en los textos de personas que necesitan ayuda, pero no saben cómo pedirla. A través de modelos de lenguaje natural.
Organización y rol dentro de ella que se beneficia con la oportunidad definida	Cualquier empresa de redes sociales puede beneficiarse de estos modelos, ya que siempre debe de ser prioridad para estos medir el bienestar de sus usuarios.
Técnicas y algoritmos a utilizar	Máquinas de Vectores de Soporte (SVM) y como técnica de vectorización se va a usar TF-IDF Algoritmo Naive Bayes Algoritmo Label Spreading

2. Entendimiento y preparación de los datos.

2.1 Perfilamiento y análisis de la calidad de los datos

2.2 Tratamiento de los datos

Tratamiento de los datos (preparación o transformaciones requeridas), de acuerdo con el dominio, las técnicas y los algoritmos seleccionados para resolver las tareas.

a. Algoritmo SVM

Para la preparación de los datos se va a:

- Eliminar filas en blanco en datos.
- Cambiar todo el texto a minúsculas.
- Tokenización de palabras.
- Eliminar palabras vacías.
- Eliminar texto no alfabético.
- Lematización de palabras.
- Identificar los conjuntos de entrenamiento y de prueba, el cual va a ser 70% y 30% respectivamente.
- Volver las variables categóricas a una representación numéricas, codificar los datos categóricos.
- Vectorización de los datos, para el caso consideramos frecuencia de término – frecuencia inversa de documento (TF-IDF por sus siglas en inglés).

b. Algoritmo Label Spreading

Para la preparación de los datos se va a:

- Eliminar filas en blanco en datos.
- Tokenización de palabras.
- Eliminar palabras vacías.
- Eliminar texto no alfabético.
- Lematización de palabras.
- Definir el conjunto de datos con dos clases, dos variables de entrada y 1000 ejemplos.
- Definir un conjunto de datos de clasificación sintética.
- Dividir datos por la mitad para formar los conjuntos de entrenamiento y prueba.
- Dividir el conjunto de datos de entrenamiento por la mitad en una parte que tiene etiquetas y otra que no.
- Concatenar el conjunto de datos de entrenamiento en una sola matriz.
- Crear una lista de valores -1 (sin etiquetar) para cada fila en la parte sin etiquetar del conjunto de datos de entrenamiento.
- Concatenar la anterior lista con las etiquetas de la parte etiquetada del conjunto de datos de entrenamiento para que se corresponda con la matriz de entrada para el conjunto de datos de entrenamiento.

c. Algoritmo Naive Bayes

Para la preparación de los datos se va a:

- Eliminar filas en blanco en datos.
- Tokenización de palabras.

- Eliminar palabras vacías.
- Eliminar texto no alfabético.
- Vectorización de los datos, a través de la función CountVectorizer y un transformador para conocer la frecuencia de algunas palabras.
- Utilizar el modelo Word embedding puede ser de utilidad para encontrar palabras clave con una representación similar. Con el fin de usar estas al momento de realizar las pruebas probabilísticas de causas dados los efectos según el teorema de Bayes.
- Identificar los conjuntos de entrenamiento y de prueba después de aplicar la transformación, dada la presencia o no de las palabras clave.

2.3 Criterios para determinar los tokens

Recuerde que es importante definir criterios para determinar los tokens que van a utilizar a nivel del modelo ya que esto afecta la calidad y el tiempo de construcción del modelo. Por ejemplo, incluir tokens que tienen alta frecuencia no genera información útil para el modelo.

Para los algoritmos se va a usar la frecuencia de término – frecuencia inversa de documento (**TF-IDF** por sus siglas en inglés) como criterio para determinar los tokens debido a que se evidencia que los modelos son más exactos cuando se utiliza la frecuencia de las palabras que más aparecen a lo largo de los documentos en vez de las palabras más frecuentes en un solo documento. Se considero el más pertinente debido a que se busca patrones a lo largo de todos los datos entregados.

3. Modelado y evaluación.

3.1 Algoritmo SVM

Estudiante encargado: Daniela Ricaurte

Descripción del algoritmo:

Máquinas de Vectores de Soporte o SVM, por sus siglas en inglés, es un algoritmo de aprendizaje supervisado para solucionar problemas de clasificación y regresión. Este determina el mejor resultado entre los vectores que pertenecen a una categoría determinada, así como los que no pertenecen, básicamente esta toma una línea divisora entre los puntos de datos (límite de decisión) y un lado indica que pertenece a la categoría y el otro que no pertenece a la categoría, es decir genera 2 subespacios. Este se puede adaptar hacia clasificaciones no binarias.

El algoritmo elige aquel hiperplano que clasifica con mayor precisión los datos en vez de maximizar el margen de distancias. Se puede programar para que no tome en cuenta aquellos datos segregados.

Este algoritmo toma un porcentaje de los datos para el entrenamiento del modelo y el porcentaje restante para probar la efectividad de la clasificación en el modelo.

Debido a que trabaja distancias euclidianas este solo puede usar valores numéricos, por lo tanto, es importante representar los datos categóricos de forma numérica

La vectorización de las palabras nos va a permitir que la colección de texto de los documentos se transforme a vectores numéricos, aquellos pueden ser basado en la frecuencia en que la palabra aparece dentro del documento, o la frecuencia en que la palabra aparece dentro de la colección de documentos

Este algoritmo se puede usar para la clasificación de texto que es el caso que aplica al proyecto.

Resultados de la evaluación cuantitativa:

	precision	recall	f1-score	support
0	0.93	0.95	0.94	33042
1	0.93	0.91	0.92	25668
accuracy			0.93	58710
macro avg	0.93	0.93	0.93	58710
weighted avg	0.93	0.93	0.93	58710

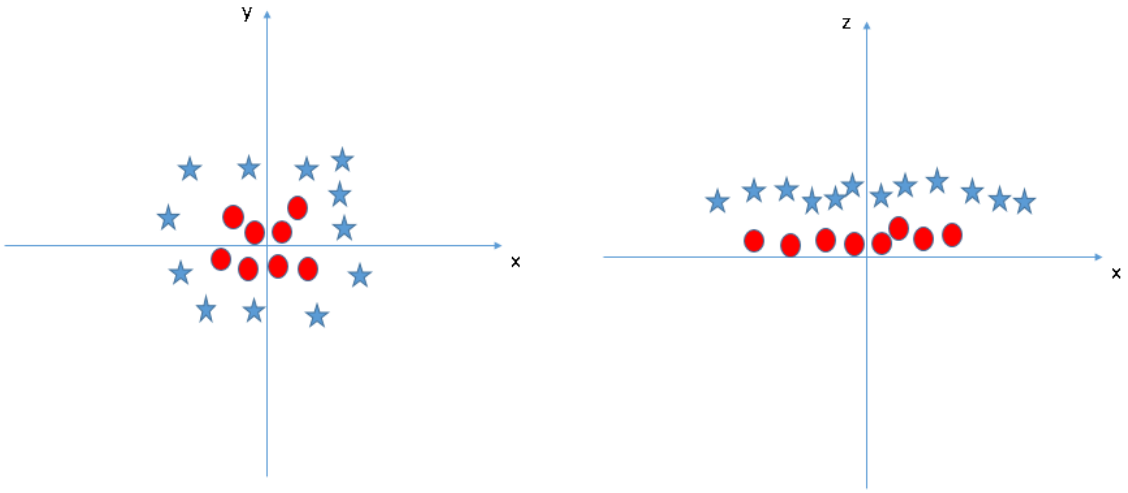
Justificación de la selección del modelo:

Es un algoritmo que no requiere muchos datos de entrenamiento para producir resultados precisos.

Es muy bueno en identificar una margen clara de separación de los datos.

Es eficiente con la memoria

Los datos no necesaria mente deben de estar separados de forma lineal ya que el algoritmo cuenta con la técnica del truco de kernel, en la cual toma espacio de entrada de baja dimensión y lo transforma en un espacio de mayor dimensión, como podemos ver en las imágenes a continuación



3.2 Algoritmo Label Spreading

Estudiante encargado: Melissa Contreras

Descripción del algoritmo:

Es un algoritmo de aprendizaje semi-supervisado que consiste en crear una matriz con ayuda de un kernel RBF (Función de Base Radial) la cual será utilizada para determinar los pesos de los bordes. Esta matriz tiene ceros en su diagonal porque un borde no debe conectarse consigo mismo.

Squared Euclidean distance

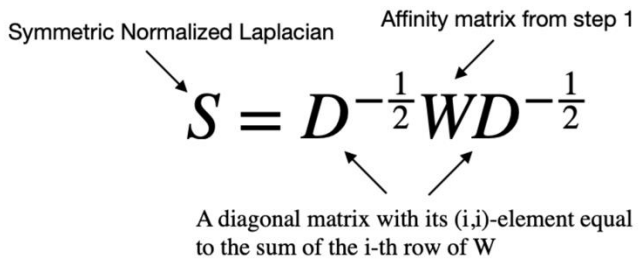
$$w_{ij} = \underbrace{e^{\frac{-||x_i - x_j||^2}{2\sigma^2}}}_{\text{RBF Kernel}}, \; where \; i \neq j \; . \; w_{ii} = 0$$

Cálculo de pesos para aristas que conectan cada par de puntos. Imagen del autor .

$$W = \begin{bmatrix} 0 & w_{21} & \cdots & w_{1j} \\ w_{12} & 0 & & \\ \vdots & & 0 & \\ w_{i1} & & & 0 \end{bmatrix}$$

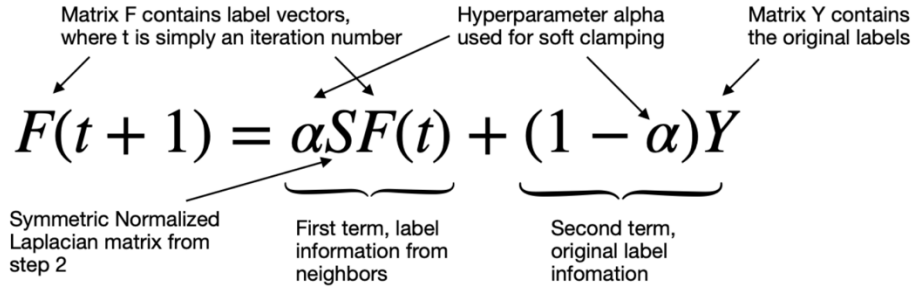
matriz de afinidad. Imagen del autor .

Después, se coge esa matriz de afinidad y la normaliza simétricamente dando como resultado una matriz laplaciana.



Gráfica normalizada simétrica Matriz laplaciana S. Imagen del autor .

Luego, realiza una multiplicación de matrices y así difunde la información desde los puntos etiquetados a puntos no etiquetados. La siguiente formula refleja que los puntos reciben la información de sus vecinos y retienen su información inicial. El parámetro alfa permite el soft clamping controlando proporción de información recibida de los vecinos contra la etiqueta original. Entonces, si alfa esta alrededor de 0, toda la información de la etiqueta original permanece, y si esta alrededor de 1 se puede reemplazar la mayor parte de información de la etiqueta original. Este proceso tiene una limitada cantidad de iteraciones.



Un proceso iterativo para encontrar las etiquetas. Imagen del autor .

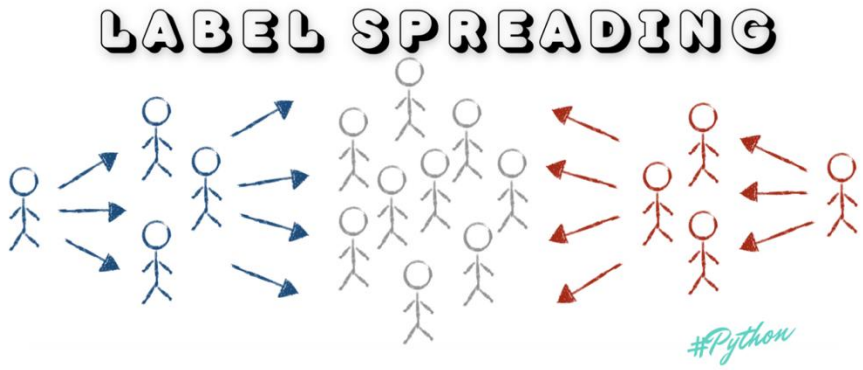
Una vez se llega al número máximo de iteraciones, la matriz resultante tendrá las probabilidades de que uno de los puntos tenga una de las etiquetas del problema. El algoritmo usa una función para encontrar la probabilidad más alta y así le asigna la etiqueta final.

Resultados de la evaluación cuantitativa: No se pudo dar un resultado cuantitativo ya que no se logró realizar el modelo predictivo por razones desconocidas.

Justificación de la selección del modelo:

Partiendo del tipo de aprendizaje, fue elegido por una importante ventaja y es que al utilizar un algoritmo semi-supervisado supera la precisión de cualquier algoritmo supervisado y no supervisado debido a que usa datos tanto etiquetados como no etiquetados. Además, estos algoritmos solo necesitan unos pocos datos etiquetados para entrenar los algoritmos.

El algoritmo Label Spreading nos permite clasificar fácilmente los datos y ponerles etiquetas y debido a que estamos trabajando con un tema tan delicado como la salud mental, es necesario que la predicción de los datos sea la más precisa y esto lo lograría el algoritmo.



3.3 Algoritmo Naive Bayes

Estudiante encargado: Julián Mora

Descripción del algoritmo:

Este algoritmo de clasificación, tal como su nombre lo indica, está basado en el teorema de probabilidad condicionada propuesto por el matemático Thomas Bayes. Entender este teorema es fundamental para comprender cómo trabaja el algoritmo, por lo cual se hará una explicación resumida.

En términos generales, este teorema vincula la probabilidad de un evento A dado un evento B, con la probabilidad de el evento B dado el evento A. Es decir, gracias a conjuntos de datos conocidos, por ejemplo, la probabilidad de A, dado B, podemos concluir que una hipótesis B sea verdadera si algún evento A ha sucedido.

La fórmula del teorema de Bayes es la siguiente, para el caso anterior:

$$P(B|A) = \frac{P(A|B) * P(B)}{P(A)}$$

Los conjuntos de datos son los siguientes:

- **P(B)** es el a priori, la probabilidad de que la hipótesis B sea verdadera antes de la evidencia.
- **P(A|B)** es el likelihood, probabilidad de que la hipótesis A sea verdadera dados los datos B.
- **P(A)** es el likelihood marginal o evidencia, la probabilidad de observar los datos A.
- **P(B|A)** es el a posteriori, la probabilidad final, en la cual B es verdadera dado A. Es la consecuencia lógica de haber usado un conjunto de datos, un likelihood y una a priori.

Resultados de la evaluación cuantitativa:

```
In [52]: y_pred = classifier.predict(Test_X)
         print(classification_report(Test_Y,y_pred))
```

```
              precision    recall  f1-score   support

 non-suicide       0.96      0.84      0.90     33024
     suicide       0.82      0.95      0.88     25686

    accuracy                           0.89     58710
   macro avg       0.89      0.90      0.89     58710
weighted avg       0.90      0.89      0.89     58710
```

Pruebas

```
In [53]: examples = ['My name is Daniela','I want to kill myself']
         example_counts = vectorizer.transform(examples)
         predictions = classifier.predict(example_counts)
         predictions
```

```
Out[53]: array(['non-suicide', 'suicide'], dtype='<U11')
```

Justificación de la selección del modelo:

Este modelo fue seleccionado por distintas razones. En primer lugar, la aplicación que tiene el teorema de bayes a problemas donde se quiere conocer la probabilidad de las causas dados los efectos.
Por poner caso, uno de los ejemplos más conocidos sobre la aplicación del teorema de Bayes es conocer la probabilidad de que un paciente con síntomas X tenga una enfermedad Y, cuando lo que normalmente se conoce es el porcentaje de pacientes de la enfermedad Y que tienen síntomas X. Este ejemplo puede ser escalado a nuestro caso de interés pues necesitamos identificar, a partir de las actitudes que tienen los usuarios en los foros de la fuente (síntomas X), cuales de ellos pueden estar atentando contra su integridad física (enfermedad Y).
Retomando lo anterior, con el enfoque analítico una vez establecido, investigamos sobre las aplicaciones del teorema de Bayes a aprendizaje automático, con el fin de cumplir los requisitos del caso. En este punto, identificamos que a través de Scikit-Learn, librería utilizada en el curso, se pueden implementar métodos enfocados al aprendizaje supervisado basados en aplicar el teorema de Bayes. Al analizar ejemplos de su uso, notamos que es bastante común que sea aplicado a la clasificación de texto, con el objetivo de detectar spam, dadas ciertas palabras clave. De esta manera, llegamos a la idea de llevar a cabo un modelo en el cual se detecte la probabilidad de que un mensaje se deba catalogar como “riesgoso para la integridad de un individuo” o en otras palabras relacionadas al contexto “suicida”, dadas algunas palabras que se repitan y presenten indicios de esto.
Así, el modelo podría asemejarse al siguiente, para dar una idea general de cómo sería su implementación, sin especificar los pronósticos:
Para el problema de clasificación de texto, S1 se interpretaría como “Peligroso” y S2 como “No peligroso”, pues ambos eventos se excluyen mutuamente. Además, M sería la ocurrencia de alguna palabra en los textos, como por ejemplo “muerte”, así M1 indicaría que “Muerte” hace parte del texto y M2 indicaría lo contrario.

- **P(S1)** es la probabilidad a priori, que un mensaje sea peligroso sin conocer el contenido del texto.
- **P(M1|S1)** es el likelihood.
- **P(M1)** es el likelihood marginal.
- **P(S1|M1)** es la probabilidad a posteriori.

$$P(S1|M1) = \frac{P(M1|S1) * P(S1)}{P(M1)}$$

Así, la ecuación anterior representaría la probabilidad de que un mensaje deba ser catalogado como peligroso dado que la presencia de palabras como “Muerte” sea verdadero.

4. Resultados.

Descripción de los resultados obtenidos, que permita a la organización comprenderlos, haciendo énfasis en el análisis de las medidas arrojadas por los modelos utilizados y cómo aportan en la consecución de los objetivos del negocio. Incluir posibles estrategias que la organización debe plantear relacionadas con los resultados obtenidos en los modelos y una justificación de por qué esa información es útil para ellos. Este resultado debe estar en el documento y deben generar un video y una presentación de máximo 7 minutos explicando su proyecto y resultados.

Algoritmo SVM

El hecho de que nos arroje unos resultados de precisión, Recall y f1 por encima de 90%, nos da a entender que con un alto grado de certeza el modelo lograra para el negocio indicar aquellos textos de Reddit que pueden ser casos de intento de suicidio o suicidio. Evidenciamos que el algoritmo dura mucho al momento de implementarse para su entrenamiento, pero una vez entrenado es altamente probable que dé resultados veraces, por lo tanto, es pertinente para resolver los objetivos del negocio para el proyecto

5. Referencias

Ray, S. (2015). *SVM | Support Vector Machine Algorithm in Machine Learning*. Analytics Vidhya. Accedido el 18 de octubre de 2022, de https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/#:~:text=%E2%80%9CSupport%20Vector%20Machine%E2%80%9D%20(SVM,mostly%20used%20in%20classification%20problems.

Bedi, G. (2018). *Simple guide to Text Classification(NLP) using SVM and Naive Bayes with Python*. Medium. Accedido el 18 de octubre de 2022, de https://medium.com/@bedigunjit/simple-guide-to-text-classification-nlp-using-svm-and-naive-bayes-with-python-421db3a72d34.

Rodrigo, J. (2017). *Máquinas de Vector Soporte (Support Vector Machines, SVMs)*. Cienciadedatos.net. Accedido el 18 de octubre de 2022, de https://www.cienciadedatos.net/documentos/34_maquinas_de_vector_soporte_support_vector_machines#M%C3%A1quinas_de_Vector_Soporte.

Shaikh, J. (2017). *Machine Learning, NLP: Text Classification using scikit-learn, python and NLTK.*. Medium. Accedido el 18 de octubre de 2022, de https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a#:~:text=Document%2FText%20classification%20is%20one,email%20routing%2C%20sentiment%20analysis%20etc.

Roldós, I. (2020). *Go-to Guide for Text Classification with Machine Learning*. Monkey Learn. Accedido el 18 de octubre de 2022, de https://monkeylearn.com/blog/text-classification-machine-learning/#:~:text=Text%20classification%20is%20a%20machine,and%20more%20accurately%20than%20humans.

Naive Bayes Classifier in Machine Learning - Javatpoint. Accedido el 18 de octubre de 2022, de https://www.javatpoint.com/machine-learning-naive-bayes-classifier#:~:text=Na%C3%AFve%20Bayes%20Classifier%20is%20one,the%20probability%20of%20an%20object.

Brownlee, J. (2020). *Naive Bayes for Machine Learning.* Accedido el 18 de octubre de 2022, de https://machinelearningmastery.com/naive-bayes-for-machine-learning/

Dobilas, S. (2021). *How to Benefit from the Semi-Supervised Learning with Label Spreading Algorithm.* Accedido el 18 de octubre de 2022, de https://towardsdatascience.com/how-to-benefit-from-the-semi-supervised-learning-with-label-spreading-algorithm-2f373ae5de96

Predictiva (s.f.*) Aprendizaje Semi-Supervisado.* Accedido el 18 de octubre de 2022, de https://www.predictiva.com.co/blog-predictiva/aprendizaje-semi-supervisado/

Brownlee, J. (2021). *Semi-Supervised Learning With Label Spreading.* Accedido el 18 de octubre de 2022, de https://machinelearningmastery.com/semi-supervised-learning-with-label-spreading/