

* Juan Cruz Rey Álvarez.
* Estudiante Ingeniería en Sistemas.
* Universidad Tecnológica Nacional.
* 4 de Noviembre de 2022

**Modelo Machine Learning – Predicción Bitcoin**

* Desarrollo:

Nuestro sistema comienza con el buscado de páginas web donde persisten noticias ricas en información que nos ayudaran a desarrollar nuestro proyecto. Para el parseo de información en la web, utilizamos una herramienta llamada webDriver que nos lo provee Selenium, una biblioteca openSoure (código abierto) que se utiliza para automatizar las pruebas realizadas en los navegadores web.

Una vez que tenemos instanciado el web driver en nuestro sistema, procedemos a ejecutarlo con los links provenientes de tres fuentes distintas de información.

La primera fue de Google, la segunda fue de Bing, y la tercera fuente de información tomada fue de Cointelegraph.

Para poder extraer dicha información, se inspecciono la página web en busca de que clase, perteneciente a css, o división, perteneciente a html, se encontraba dicho texto, una vez localizado se automatizaba la extracción de información de dichas páginas.

Luego de juntar las tres fuentes de información, quedo una lista ,codificada en Python, con toda la información junta. Cada componente de la lista quedo como una oración independiente de las demás.

Antes de comenzar con el desarrollo de nuestro modelo, era necesario realizar ingeniería de datos a nuestra información que planeo predecir en nuestro futuro modelo.

Para eso conocimos las llamadas StopWords, palabras que ensucian nuestro modelo de Machine Learning sacándole performance a la hora de predecir, ya que estas palabras no son poderosas, no aportan nada significativo a nuestro modelo, son repetitivas , y no expresan sentimientos.

Una vez realizada la limpieza de nuestros datos, se procedió a realizar nuestro modelo que tenía como objetivo tomar oraciones en idioma inglés y poder predecir si el sentimiento asociado a esa información es positivo o negativo. Este es el **target**. En caso de ser positivo lo marcamos con uno (1) y en caso de ser negativo lo marcamos con cero (0).

Para el desarrollo y entrenamiento del modelo utilizamos 3 DataSets que nos aportaban dichas oraciones y expresaban el sentimiento de forma binaria. El primer DataSet es proveniente de Amazon, una corporación estadounidense de comercio electrónico y servicios de computación en la nube a todos los niveles con sede en la ciudad de Seattle, Washington.

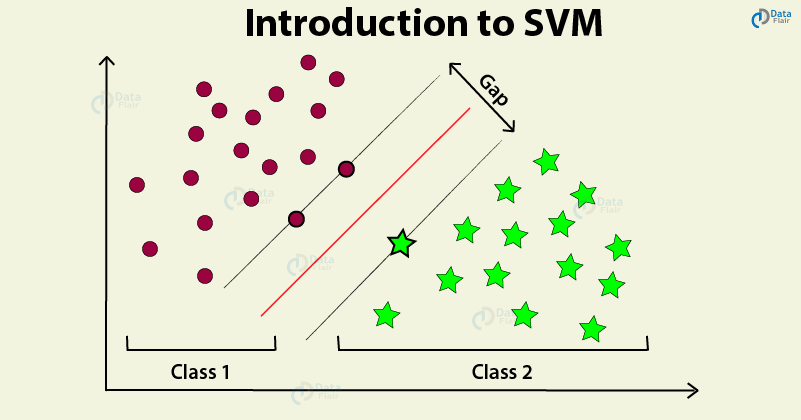
El segundo DataSet tiene origen en IMDb , Internet Movie Database es una base de datos en línea que en un principio almacenaba información relacionada con películas, y con el tiempo se transforma en la base de datos más grande del mundo.

Y, por último, el tercer DataSet es de Yelp, una empresa estadounidense que desarrolla el sitio web Yelp.com y la aplicación móvil Yelp, que publican reseñas de empresas de fuentes múltiples.

Una vez que se reunió la información, estaba todo listo en un DataSet, que se utilizaría tanto para el entrenamiento, como así también para el testeo del mismo.

Nuestro DataSet en total cuenta con 2745 filas y únicamente 2 columnas como se mencionaron anteriormente, la oración informativa y el resultado de la predicción de sentimiento en forma binaria.

A la hora de iniciar en la búsqueda del modelo, se optó por comenzar a trabajar en un modelo con un algoritmo SVM, es un algoritmo que representa a los puntos de muestra en el espacio, separando las clases a 2 espacios lo más amplios posibles mediante un hiperplano de separación definido como el vector entre los 2 puntos, de las 2 clases, más cercanos al que se llama vector soporte. Trabaja intentando encontrar el hiperplano mencionado para **maximizar la distancia** entre nuestras muestras clasificadas.



Para la construcción de este modelo utilizamos Scikit-Learn, una biblioteca core de Machine Learning. Utilizamos varias herramientas que las explicare mientras detallo el procedimiento.

Lo primero que realizamos fue vectorizar nuestro DataSet con una herramienta llamada CountVectorizer , la utilizamos para transformar el texto de nuestro DataSet a formato numérico, y de esta manera podemos acceder más rápido y con mayor eficiencia a nuestros datos.

Luego utilizamos una herramienta llamada TfidfVectorizer que nos ayuda a conseguir una medida que nos va a facilitar evaluar que tan relevante es una palabra en una colección. Es decir que, si dos DataSets tienen la misma palabra , esa palabra tiene menos poder a la hora de predecir comparado con una palabra única entre dos DataSets.

Como era de esperarse utilizamos el paquete accuracy score ,como lo indica el nombre es un paquete que nos va a ayudar a saber cuál es el accuracy de nuestro modelo a la hora de hacer una predicción. También train test split, este paquete lo Utilizamos para poder dividir nuestro DataSet en una parte de entrenamiento y otra parte de prueba o testeo.

Para ajustar nuestro modelo al conjunto de datos que tenemos durante la fase de entrenamiento utilizamos TransformerMixin. Con él nos aseguramos de que nuestro modelo aprenda patrones del conjunto de datos que nosotros le ofrecemos al modelo para aprender.

Para aumentar la prolijidad, expresividad y facilidad de nuestro modelo utilizamos Pipeline, una herramienta que nos deja automatizar el trabajo que haremos con las bibliotecas anteriormente mencionadas. Esto hace que el proceso de creación del modelo sea mucho más rápido y mucho más sencillo, ya que todas las etapas se agrupan en un solo proceso unitario.

Se termino el modelo, se probó y **los resultados iniciales no fueron buenos.** Arrojo un valor de accuracy de 0.999028 puntos. Claramente no es un valor real, y nuestro modelo estaba bastante sobre Ajustado.

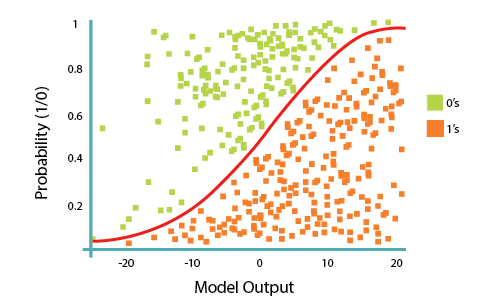
Se intento corregir, pero no hubo caso y **el modelo fue descartado**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Antes de comenzar a trabajar en un nuevo modelo, quise asegurarme de que los tres DataSets no eran el problema. Por lo tanto, los separa y procedí a probarlos por separado para ver cómo se comportaban individualmente con el nuevo modelo a desarrollar.

Las herramientas utilizadas para este nuevo modelo fueron las mismas q los anteriores menos pipelines que no lo tuve en cuenta. El modelo a desarrollar elegido fue una regresión logística, un tipo de análisis de regresión utilizado para predecir el resultado de una variable categórica en función de las variables independientes o predictoras.



Como lo mencionamos anteriormente, se utilizó un solo DataSet para comenzar a desarrollar el modelo, el DataSet elegido fue el de Yelp. Luego de vectorizar entrenar y probar arrojo unos magníficos 0.808 de accuracy.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Hasta el momento, parecía que íbamos por buen camino, por ende, lo probamos con los otros dos DataSets por separado para observar los resultados.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Efectivamente los tres DataSets por separado entrenando y testeando al modelo independientes entre cada uno, dieron grandísimos resultados. Resultados mucho más reales que nuestro primer modelo.

Por supuesto ahora unimos los DataSet y probamos nuestro modelo entrenando y testeando con la combinación de ellos.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Gran resultado para nuestro accuracy final.

Pero, como todos sabemos, no podemos evaluar nuestro modelo solo por el score o accuracy que arroja. Por lo tanto, evaluamos la curva ROC del mismo. La curva ROC nos mostró que nuestro modelo tiene buena sensibilidad, una buena representación de verdaderos positivos frente a la razón de falsos positivos.

Interfaz de usuario gráfica

Descripción generada automáticamente

Parece que este modelo sigue por buen camino, ya tenemos varias justificaciones para elegirlo como el modelo ganador hasta ahora. Pero sigamos evaluándolo.

También observamos un excelente valor de AUC, un área debajo de la curva ROC de 0.895627. Gran valor para nuestro modelo.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

El modelo se desempeña bien. Pero sigámoslo apretando para ver que valores arroja. Realizo una Matriz de confusión para poder tener una referencia visual de los errores tipo 1 y tipo 2 de nuestro modelo.

Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Un promedio mas que aceptable para nuestro modelo de errores tipo 1 y 2.

Estamos listo para ponerlo en funcionamiento y verificar cómo se comporta “en la vida real”.

Para esto, decidimos utilizar una base de datos, donde persistiremos la fecha de ejecución del modelo, el valor arrojado por nuestro modelo, el valor humano comprendido por dicho número arrojado, y la frecuencia de palabra de las noticias.

Una captura de pantalla de una red social

Descripción generada automáticamente

Trabajamos con MySQL Workbench para poder visualizar nuestra base de datos y por detrás trabajamos con MariaDB , un sistema de gestión de bases de datos derivado de MySQL.

# Referencias:

Lexalytics. (15 de octubre de 2022). *Machine Learning (ML) for Natural Language Processing (NLP)*. Lexalytics. https://www.lexalytics.com/blog/machine-learning-natural-language-processing/

Selenium. (29 de agosto de 2022). *The Selenium Browser Automation Project*. Selenium. https://www.selenium.dev/documentation/

Zipmex. (28 de octubre de 2022). *Bitcoin (BTC) Price Prediction 2022 – 2030 According To The Crypto Experts*. Zipmex. https://zipmex.com/learn/bitcoin-price-prediction/

Cointelegraph. (1 de noviembre de 2022). *The future of money - News*. Cointelegraph. https://cointelegraph.com/tags/bitcoin

Kavita-ganesan. (23 de septiembre de 2022). *Bitcoin (BTC) news today*. Kavita Ganesan. <https://kavita-ganesan.com/what-are-stop-words/#.Y2EjU3bMKUk>

Scikit-learn. (1 de noviembre de 2022). *CountVectorizer*. Scikit-learn. <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html>

Scikit-learn. (1 de noviembre de 2022). TfidfVectorizer. Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.feature\_extraction.text.TfidfVectorizer.html

Scikit-learn. (1 de noviembre de 2022). *TransformerMixin*. Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.base.TransformerMixin.html

Mariadb. (1 de noviembre de 2022). MariaDB Server Documentation*.*MariaDB. https://mariadb.com/kb/en/documentation/

* Conclusiones:

Se probo el modelo es 4 dias distintos para evaluar su comportamiento.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteHistorial grafico semanal de Bitcoin:

Historial Promedio semanal De Bitcoin:

Una captura de pantalla de una red social

Descripción generada automáticamente

Resultados del modelo:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Conclusión:

**Fecha 11/01/2022:**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Nuestro modelo nos predijo el día anterior, fecha 10/31/2022, que el valor del bitcoin iba a decaer dándonos un sentimiento negativo asociado a las noticias de ese día.

El valor real de la fecha 11/01/2022 fue:

Una captura de pantalla de una red social

Descripción generada automáticamente

Por lo tanto: La predicción de nuestro modelo fue acertada. Nos dijo que en base a información recopilada el 10/31/2022 , el bitcoin iba a decaer en la fecha 11/01/2022 y eso fue lo ocurrido.

**EL MODELO ACERTO.**

**Fecha 11/04/2022:**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Nuestro modelo nos predijo el día anterior, fecha 10/03/2022, que el valor del bitcoin iba a subir dándonos un sentimiento positivo asociado a las noticias de ese día.

Una captura de pantalla de una red social

Descripción generada automáticamenteEl valor real de la fecha 11/04/2022 fue:

Por lo tanto: La predicción de nuestro modelo fue acertada. Nos dijo que en base a información recopilada el 11/03/2022 , el bitcoin iba a subir en la fecha 11/04/2022 y eso fue lo ocurrido.

**EL MODELO ACERTO.**