|  |  |
| --- | --- |
| **TEMA:** | **Big Data y Machine Learning** |

|  |
| --- |
| **PLANTEAMIENTO DEL TRABAJO** |
| **CASO DE USO: Predicción de Ventas de Helados**  Disponemos de los datos históricos de las ventas de 36 heladerías ubicadas en España, así como de la localización de las mismas, sus horarios y algunas características propias como la posibilidad de consumo de productos en la propia tienda o la posibilidad de adquisición de productos Premium. Además de esto, se han extraído datos de fuentes externas acerca del calendario laboral y las temperaturas registradas diariamente en los últimos años.  A partir de los datos cada heladería, combinados con los datos obtenidos de fuentes externas, se busca predecir las ventas en una fecha o rango de fechas futuro (como se puede ver en el Notebook de Zeppelin). Para realizar esto, es necesario construir un modelo predictivo que permita, a partir de las variables de entrada, determinar el valor de la variable objetivo (la cantidad de ventas diaria en euros).  Mediante el uso del algoritmo de regresión GBT (árboles potenciados por gradiente), es posible modelar el comportamiento de las ventas en base a una serie de reglas. Este algoritmo, basado en el uso de árboles de decisión, genera una secuencia lineal de árboles que le permiten realizar un buen ajuste sobre los datos de entrada y tratar tanto variables cuantitativas como cualitativas, sin necesidad de normalizar los datos.  El uso de Apache Spark, con la librería MlLib, permite realizar la construcción del modelo predictivo y la realización de predicciones en un clúster de hadoop, donde se encuentran disponibles las fuentes de  información citada.  Tras realizar pruebas con diferentes modelos de Machine Learning de regresión (como la regresión lineal, arboles de decisión o Random Forest), se ha determinado que GBT es el algoritmo que ofrece los mejores resultados tras analizar las métricas de RMSE y el coeficiente R2, siguiendo un procedimiento de validación cruzada de los modelos creados.  **MODELO DEL CASO DE USO:**  Para predecir las ventas de helados, se pueden utilizar herramientas de análisis de datos como Hadoop, Apache Spark y Apache Zeppelin. A continuación, te proporcionaré un ejemplo de cómo se podrían utilizar estas herramientas:   * Recopilación de datos: Se recopilan datos históricos de ventas de helados, así como información sobre el clima, la ubicación y otros factores que puedan afectar las ventas. * Almacenamiento de datos: Los datos se almacenan en un clúster Hadoop distribuido, como Apache HDFS, para su procesamiento posterior. * Procesamiento de datos: Se utiliza Apache Spark para procesar los datos y crear modelos de Machine Learning. Por ejemplo, se pueden utilizar técnicas de regresión para predecir las ventas futuras en función de los datos históricos y otros factores. * Visualización de datos: Apache Zeppelin se utiliza para visualizar los datos y crear gráficos interactivos que ayudan a los analistas de datos a identificar patrones y tendencias. Por ejemplo, se pueden crear gráficos que muestren las ventas por ubicación o por tipo de helado. * Implementación del modelo: Los modelos de Machine Learning se implementan en el sistema de ventas de la empresa para ofrecer recomendaciones sobre los niveles de inventario y las estrategias de precios.   Es importante tener en cuenta que la implementación de estas herramientas debe llevarse a cabo en un ambiente seguro, con medidas adecuadas para proteger la privacidad de los clientes y evitar la exposición de datos sensibles. Además, es importante tener en cuenta que la precisión de las predicciones puede verse afectada por factores externos, como cambios en el clima o la economía. Por lo tanto, es importante monitorear constantemente los datos y ajustar los modelos en consecuencia.  **Queda en el estudiante elaborar el grafico de caso de uso para este ejercicio.**  **ARQUITECTURA DEL MODELO**  En el caso de estudio que presentamos, hacemos uso de la herramienta Pentaho Data Integration para procesar datos de diferentes fuentes de datos. Esta información interna y externa incluye datos históricos sobre las ventas de las heladerías, así como datos acerca del calendario laboral, la temperatura, horarios de apertura etc.  Aunque la aplicación de demostración no contempla el procesamiento de datos de APIs o el uso de Wescraping, estas tecnologías se han implementado en otros proyectos y pueden ser integradas con Pentaho Data Integration. Tras la carga y procesamiento de datos, se ha generado un archivo de datos almacenado en un clúster de Hadoop (HDFS).  Se ha utilizado Apache Spark para la explotación de los datos mediante el uso de Spark SQL y Spark MlLib. Se ha construído un modelo predictivo para determinar las ventas de las heladerías partiendo de la fecha y las variables de entorno que podrían condicionar las mismas. Se ha utilizado el algoritmo GBT que ha demostrado ofrecer los resultados más adecuados en la evaluación de diferentes modelos de regresión.  Se ha utilizado la herramienta Apache Zeppelin para la visualización de los datos y la realización dinámica de predicciones por parte de los usuarios, utiizando los modelos predictivos generados. Las interfaces de usuario de Zeppelin pueden ser construídas de forma sencilla mediante el uso de formularios dinamicos de Zeppelin o bien de forma más personalizable mediante el uso de HTML, CSS, Javascript y Angular.  La configuración realizada en el clúster permite la utilización de otras herramientas de visualización como Apache Superset para la realización de cuadros de mandos, así como la conexión entre el clúster y herramientas externas de Business Intelligence como pueden ser Tableau, PowerBI o Pentaho.  **PASO 1: Arquitectura Propuesta**  Distribución de Hadoop Hortonworks (HDP) open source para el despliegue de un clúster de procesamiento y almacenamiento distribuido y escalable. Incluye herramientas para almacenar y procesar cualquier tipo de fuente de datos:   * **Spark con Python, R, SQL o Scala para Machine Learning** * Hive para data warehousing y SQL, Kafka para adquisición real time, Kylin y Druid para OLAP * **Zeppelin** y Superset para visualización de datos   **Ejemplo de Arquitectura propuesta.**  Un ejemplo de arquitectura para la propuesta de predicción de ventas de helados utilizando Hadoop, Apache Spark y Apache Zeppelin:   * Recopilación de datos: Los datos se recopilan de múltiples fuentes, como sistemas de punto de venta, sensores meteorológicos y bases de datos externas. Estos datos se almacenan en un sistema de registro de eventos, como Apache Kafka, para garantizar que se capturen todos los datos relevantes. * Almacenamiento de datos: Los datos se almacenan en un clúster Hadoop distribuido, como Apache HDFS, para su procesamiento posterior. El clúster Hadoop proporciona almacenamiento escalable y tolerancia a fallos para garantizar que los datos estén siempre disponibles. * Procesamiento de datos: Se utiliza Apache Spark para procesar los datos y crear modelos de Machine Learning. Por ejemplo, se pueden utilizar técnicas de regresión para predecir las ventas futuras en función de los datos históricos y otros factores. Los modelos se pueden entrenar utilizando herramientas como MLlib y TensorFlow. * Visualización de datos: Apache Zeppelin se utiliza para visualizar los datos y crear gráficos interactivos que ayudan a los analistas de datos a identificar patrones y tendencias. Por ejemplo, se pueden crear gráficos que muestren las ventas por ubicación o por tipo de helado. * Implementación del modelo: Los modelos de Machine Learning se implementan en el sistema de ventas de la empresa para ofrecer recomendaciones sobre los niveles de inventario y las estrategias de precios. Por ejemplo, se pueden utilizar los modelos para ajustar automáticamente los precios en función de la demanda prevista. * Monitoreo y mantenimiento: Es importante monitorear constantemente los datos y ajustar los modelos en consecuencia. Se pueden utilizar herramientas como Apache NiFi y Grafana para monitorear el rendimiento del sistema y detectar posibles problemas.   Es importante tener en cuenta que la arquitectura exacta puede variar según las necesidades específicas de la empresa y los requisitos del sistema. Además, es importante implementar medidas adecuadas para proteger la privacidad de los clientes y evitar la exposición de datos sensibles**.**  **PASO 2: Preparación y carga de datos**  Se dispone de un archivo CSV generado a partir de un proceso ETL con Pentaho Data Integration (PDI) que ha sido almacenado en un clúster de Hadoop. Los datos del archivo pueden ser procesados con Apache Spark a través del uso de objetos DataFrame para realizar la analítica y predicciones de ventas futuras, mediante la aplicación de algoritmos de Machine Learning.    **PASO 3: Analítica tradicional**  Una vez leídos los datos de las ventas, es posible analizar los mismos utilizando Spark SQL y realizar representaciones gráficas que permitan interpretar su estructura y contenido. Teniendo en cuenta la ubicación de la heladería, las fechas de venta y otros parámetros, es posible realizar consultas sobre los datos históricos para entender el comportamiento pasado de las ventas que se han realizado, las cuales varían ampliamente dependiendo de estas características.  **Analítica Tradicional**  Ejemplo de arquitectura para la analítica tradicional de las ventas de helados:   * Recopilación de datos: Los datos se recopilan de múltiples fuentes, como sistemas de punto de venta, sensores meteorológicos y bases de datos externas. Estos datos se almacenan en un sistema de almacenamiento de datos relacional, como MySQL o Oracle. * Procesamiento de datos: Se utilizan herramientas de ETL (Extract, Transform, Load) para procesar los datos y prepararlos para su análisis. Por ejemplo, se pueden utilizar herramientas como Talend o Pentaho para limpiar y transformar los datos. * Análisis de datos: Se utilizan herramientas de análisis de datos, como Tableau o Power BI, para analizar los datos y crear informes interactivos que ayuden a los analistas de datos a identificar patrones y tendencias. Por ejemplo, se pueden crear gráficos que muestren las ventas por ubicación o por tipo de helado. * Implementación de estrategias: Los informes y análisis se utilizan para implementar estrategias de ventas y marketing en la empresa. Por ejemplo, se pueden utilizar los análisis para ajustar los precios o para mejorar la gestión del inventario.   Es importante tener en cuenta que la arquitectura exacta puede variar según las necesidades específicas de la empresa y los requisitos del sistema. Además, es importante implementar medidas adecuadas para proteger la privacidad de los clientes y evitar la exposición de datos sensibles.  **Analítica Tradicional-Datos**  **DESCRIPCIÓN GENERAL DE CÓMO SE PUEDE REALIZAR UN ANÁLISIS DE DATOS TRADICIONAL UTILIZANDO PYTHON Y LA BIBLIOTECA PANDAS.**  En primer lugar, se pueden cargar los datos de ventas de helados desde un archivo CSV utilizando la función read\_csv de Pandas. Luego se pueden realizar diferentes operaciones para explorar y analizar los datos, como las siguientes:   * Mostrar los primeros registros del conjunto de datos utilizando el método head. * Mostrar información básica sobre el conjunto de datos utilizando el método info. * Calcular estadísticas descriptivas de las variables numéricas utilizando el método describe. * Visualizar los datos utilizando gráficos, como histogramas, diagramas de dispersión, diagramas de caja, etc. * Realizar análisis de correlación entre las variables utilizando la función corr de Pandas.   Por ejemplo, el siguiente código muestra cómo cargar y explorar los datos de ventas de helados utilizando Pandas:  **import pandas as pd**  **# Cargar los datos de ventas de helados**  **data = pd.read\_csv('ventas\_helados.csv')**  **# Mostrar los primeros registros del conjunto de datos**  **print(data.head())**  **# Mostrar información básica sobre el conjunto de datos**  **print(data.info())**  **# Calcular estadísticas descriptivas de las variables numéricas**  **print(data.describe())**  **# Visualizar los datos utilizando gráficos**  **data.plot(kind='hist', y='ventas')**  **data.plot(kind='scatter', x='temperatura\_media', y='ventas')**  **data.plot(kind='box', y='ventas')**  **# Realizar análisis de correlación entre las variables**  **corr\_matrix = data.corr()**  **print(corr\_matrix)**  **HISTÓRICO DE VENTAS POR MES**  Ejemplo de cómo se puede utilizar Pandas y Matplotlib para visualizar el histórico de ventas de helados por mes.  **import pandas as pd**  **import matplotlib.pyplot as plt**  **# Cargar los datos de ventas de helados**  **data = pd.read\_csv('ventas\_helados.csv')**  **# Convertir la columna de fechas a formato de fecha**  **data['fecha'] = pd.to\_datetime(data['fecha'])**  **# Agrupar las ventas por mes y calcular la suma**  **ventas\_por\_mes = data.groupby(pd.Grouper(key='fecha', freq='M'))['ventas'].sum()**  **# Visualizar el histórico de ventas por mes**  **plt.plot(ventas\_por\_mes.index, ventas\_por\_mes.values)**  **plt.xlabel('Mes')**  **plt.ylabel('Ventas')**  **plt.title('Histórico de ventas de helados por mes')**  **plt.show()**  En este ejemplo, se carga el conjunto de datos de ventas de helados utilizando la función read\_csv de Pandas. Luego se convierte la columna de fechas a formato de fecha utilizando la función to\_datetime. Se agrupan las ventas por mes utilizando la función groupby y se calcula la suma utilizando el método sum. Finalmente, se visualiza el histórico de ventas por mes utilizando la función plot de Matplotlib.  **PASO 4: Machine Learning. Algoritmo de regresión GBT para la predicción de ventas**  Se construirá un modelo de regresión de Machine Learning a partir de los datos de las ventas de las heladerías, generadas en base a las diferentes variables de entorno. Se utilizará la librería Spark MlLib (Machine Learning) para la construcción de los modelos necesarios para predecir las ventas. Tras analizar los resultados obtenidos con múltiples algoritmos de regresión (regresión lineal, árboles de decisión, Random Forest y GBT), se ha determinado que el algoritmo GBT es el que ofrece mejores resultados en las predicciones y, por tanto, se ha seleccionado para realizar las predicciones.  **PASO 5: Creación del modelo**  **Archivo de Codigo Fuente**  Ejemplo de cómo se puede implementar una regresión de Gradient Boosted Trees (GBT) en Python utilizando la biblioteca Scikit-learn. El siguiente código es un ejemplo de cómo entrenar y evaluar un modelo de GBT para predecir las ventas de helados:  **from sklearn.ensemble import GradientBoostingRegressor**  **from sklearn.metrics import mean\_squared\_error**  **import pandas as pd**  **# Cargar los datos de ventas de helados**  **data = pd.read\_csv('ventas\_helados.csv')**  **# Separar los datos en características y etiquetas**  **X = data.drop('ventas', axis=1)**  **y = data['ventas']**  **# Dividir los datos en conjuntos de entrenamiento y prueba**  **X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)**  **# Crear el modelo de GBT**  **gbt = GradientBoostingRegressor(n\_estimators=100, learning\_rate=0.1, max\_depth=3, random\_state=42)**  **# Entrenar el modelo con los datos de entrenamiento**  **gbt.fit(X\_train, y\_train)**  **# Evaluar el modelo con los datos de prueba**  **y\_pred = gbt.predict(X\_test)**  **mse = mean\_squared\_error(y\_test, y\_pred)**  **print('Mean squared error:', mse)**  En este ejemplo, los datos de ventas de helados se cargan desde un archivo CSV y se separan en características (X) y etiquetas (y). Luego se dividen los datos en conjuntos de entrenamiento y prueba utilizando la función train\_test\_split de Scikit-learn. Se crea el modelo de GBT utilizando la clase GradientBoostingRegressor y se entrena con los datos de entrenamiento utilizando el método fit. Finalmente, se evalúa el modelo con los datos de prueba utilizando la métrica del error cuadrático medio (MSE) mediante la función mean\_squared\_error de Scikit-learn.  **PASO 6: Evaluación del Modelo**  Una vez construido el modelo, es posible evaluarlo siguiendo la técnica de la validación cruzada. Para ello se realizan predicciones sobre los datos de test y se comparan los valores predecidos con los valores reales. Por último, se calcula el RMSE y el coeficiente r2 como métricas de evaluación de la calidad de los resultados.  Para conseguir una predicción más ajustada, se ha construido un modelo de regresión específico para cada heladería y se han calculado las métricas de evaluación de las predicciones realizadas. A continuación, se muestran los resultados obtenidos para cada una de las heladerías. Es posible comprobar las heladerías cuyas predicciones son más fiables (las que presentan un valor más cercano a 1 en el coeficiente r2).  **Archivo de evaluación del modelo**  Ejemplo de cómo se puede realizar la validación cruzada y calcular las métricas de evaluación de un modelo de regresión utilizando Python y la biblioteca Scikit-learn.  **from sklearn.model\_selection import cross\_val\_predict**  **from sklearn.metrics import mean\_squared\_error, r2\_score**  **import pandas as pd**  **import numpy as np**  **from sklearn.ensemble import GradientBoostingRegressor**  **# Cargar los datos de ventas de helados**  **data = pd.read\_csv('ventas\_helados.csv')**  **# Separar los datos en características y etiquetas**  **X = data.drop('ventas', axis=1)**  **y = data['ventas']**  **# Crear el modelo de GBT**  **gbt = GradientBoostingRegressor(n\_estimators=100, learning\_rate=0.1, max\_depth=3, random\_state=42)**  **# Realizar la validación cruzada**  **y\_pred = cross\_val\_predict(gbt, X, y, cv=10)**  **# Calcular las métricas de evaluación**  **rmse = np.sqrt(mean\_squared\_error(y, y\_pred))**  **r2 = r2\_score(y, y\_pred)**  **# Imprimir los resultados**  **print('RMSE:', rmse)**  **print('R2 score:', r2)**  En este ejemplo, se cargan los datos de ventas de helados desde un archivo CSV y se separan en características (X) y etiquetas (y). Se crea el modelo de GBT utilizando la clase GradientBoostingRegressor. Luego se realiza la validación cruzada utilizando la función cross\_val\_predict de Scikit-learn con un parámetro cv que indica el número de pliegues en la validación cruzada. Finalmente, se calculan las métricas de evaluación del modelo utilizando las funciones mean\_squared\_error y r2\_score de Scikit-learn.  **PASO 7: Ejecución del modelo. Generación de Predicciones de ventas (utilizando regresión)**  Una vez construido el modelo de regresión para la predicción de ventas, es posible realizar predicciones para una fecha concreta o un rango de fechas para una heladería específica (teniendo en cuenta las diferentes variables de entrada que fueron utilizadas para construir el modelo).  En esta aplicación de demostración, se consulta el calendario para cada día a predecir para comprobar si es laborable, sábado, domingo o festivo y se pondera la temperatura media según el mes del año para conseguir predicciones fiables en rangos de fechas. En lo referente a las horas de apertura, consumo in situ y venta de productos Premium, se asume que estas características se mantienen a lo largo del periodo seleccionado en la predicción.  **Ejecución del Modelo**  Ejemplo de cómo se pueden realizar predicciones utilizando un modelo de regresión de Gradient Boosted Trees (GBT) en Python y la biblioteca Scikit-learn.  **import pandas as pd**  **from sklearn.ensemble import GradientBoostingRegressor**  **# Cargar los datos de ventas de helados**  **data = pd.read\_csv('ventas\_helados.csv')**  **# Separar los datos en características y etiquetas**  **X = data.drop('ventas', axis=1)**  **y = data['ventas']**  **# Crear el modelo de GBT**  **gbt = GradientBoostingRegressor(n\_estimators=100, learning\_rate=0.1, max\_depth=3, random\_state=42)**  **# Entrenar el modelo con todos los datos**  **gbt.fit(X, y)**  **# Realizar una predicción para una fecha concreta**  **fecha = '2023-05-05'**  **caracteristicas = [28, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0]**  **prediccion = gbt.predict([caracteristicas])**  **print('La predicción de ventas para la fecha', fecha, 'es:', prediccion)**  En este ejemplo, se cargan los datos de ventas de helados desde un archivo CSV y se separan en características (X) y etiquetas (y). Se crea el modelo de GBT utilizando la clase GradientBoostingRegressor y se entrena con todos los datos. Luego se realiza una predicción para una fecha concreta utilizando las características correspondientes a esa fecha. En este caso, las características se representan como una lista de valores numéricos que representan la temperatura media, el día de la semana, el mes del año y los días festivos. Finalmente, se imprime la predicción de ventas para esa fecha. |