Clase 2 - Resolviendo un problema con Machine Learning

Facundo González

11 de abril de 2024

Outline

Pipeline de ML

1. **Manejo de datos:** análisis de los datos, procesar la entrada de los datos crudos, aplicar pipeline de preprocesamiento, dividir dataset en train y test.

Pipeline de ML

1. **Manejo de datos:** análisis de los datos, procesar la entrada de los datos crudos, aplicar pipeline de preprocesamiento, dividir dataset en train y test.
2. **Entrenamiento de modelo:** utilizar datos de entrenamiento para entrenar un modelo, elección del algoritmo o modelo a utilizar.

Pipeline de ML

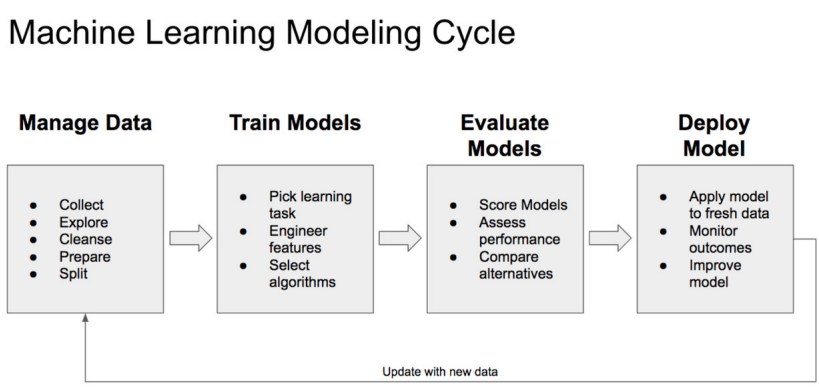
1. **Manejo de datos:** análisis de los datos, procesar la entrada de los datos crudos, aplicar pipeline de preprocesamiento, dividir dataset en train y test.
2. **Entrenamiento de modelo:** utilizar datos de entrenamiento para entrenar un modelo, elección del algoritmo o modelo a utilizar.
3. **Evaluación del modelo:** métricas del modelo, evaluar la performance.

Pipeline de ML

1. **Manejo de datos:** análisis de los datos, procesar la entrada de los datos crudos, aplicar pipeline de preprocesamiento, dividir dataset en train y test.
2. **Entrenamiento de modelo:** utilizar datos de entrenamiento para entrenar un modelo, elección del algoritmo o modelo a utilizar.
3. **Evaluación del modelo:** métricas del modelo, evaluar la performance.
4. **Funcionamiento del modelo:** utilizar el modelo para realizar predicciones e inferencias.

Pipeline de ML

Figura: Etapas del pipeline de ML.



Manejo de datos

Es importante entender nuestros datos antes de empezar a entrenar un modelo de ML. Muchas veces tenemos que investigar y aprender sobre el **dominio** del problema, conocer la terminología y el significado de cada feature.

- **Exploratory Data Analysis:** análisis general del dataset, encontrando patrones y comportamiento de los datos. Nos sirve para mejorar la calidad de los datos, por ejemplo encontrado problemas como datos erróneos o *missing values*. Esta fase es manual y depende de cada problema, vamos a usar librerías que nos van a ayudar: Pandas y Seaborn

Manejo de datos

Es importante entender nuestros datos antes de empezar a entrenar un modelo de ML. Muchas veces tenemos que investigar y aprender sobre el **dominio** del problema, conocer la terminología y el significado de cada feature.

- **Exploratory Data Analysis:** análisis general del dataset, encontrando patrones y comportamiento de los datos. Nos sirve para mejorar la calidad de los datos, por ejemplo encontrado problemas como datos erróneos o *missing values*. Esta fase es manual y depende de cada problema, vamos a usar librerías que nos van a ayudar: Pandas y Seaborn
- **Pipeline de preprocesamiento:** armado de un pipeline (secuencia de pasos) para el preprocesamiento de los datos crudos. Compactar las transformaciones en un pipeline nos permite aplicarselo a los nuevos datos para las predicciones. A esto se lo conoce como **Feature Engineering**.

Manejo de datos

Es importante entender nuestros datos antes de empezar a entrenar un modelo de ML. Muchas veces tenemos que investigar y aprender sobre el **dominio** del problema, conocer la terminología y el significado de cada feature.

- **Exploratory Data Analysis:** análisis general del dataset, encontrando patrones y comportamiento de los datos. Nos sirve para mejorar la calidad de los datos, por ejemplo encontrado problemas como datos erróneos o *missing values*. Esta fase es manual y depende de cada problema, vamos a usar librerías que nos van a ayudar: Pandas y Seaborn
- **Pipeline de preprocesamiento:** armado de un pipeline (secuencia de pasos) para el preprocesamiento de los datos crudos. Compactar las transformaciones en un pipeline nos permite aplicarselo a los nuevos datos para las predicciones. A esto se lo conoce como **Feature Engineering**.

  En esta etapa también hacemos el split en train y test.

Entrenamiento del modelo

En esta etapa ya tenemos los datos preparados para entrenar al modelo. Debemos seleccionar qué modelo vamos a utilizar, y los hiper parámetros correspondientes. Para esto se puede utilizar **hyperparameter tuning**.

Evaluación del modelo

En esta etapa evaluamos al modelo ya entrenado utilizando los datos de test. Utilizamos diferentes métricas para poder entender la performance del modelo. Esta etapa es importante para comparar la performance entre diferentes modelos para seleccionar el que funcione mejor.

Features

Las **features** o atributos son las características del dataset que vamos a utilizar para entrenar y predecir en los modelos.
Vamos a distinguir dos tipos de features:

- **Numéricas:** son características que tienen como valor números continuos. Ejemplo: edad, peso, salario. Generalmente las podemos utilizar directamente en los modelos.

Features

Las **features** o atributos son las características del dataset que vamos a utilizar para entrenar y predecir en los modelos.
Vamos a distinguir dos tipos de features:

- **Numéricas:** son características que tienen como valor números continuos. Ejemplo: edad, peso, salario. Generalmente las podemos utilizar directamente en los modelos.
- **Categóricas:** tienen como valor una categoría discreta, que puede ser en forma de string. Ejemplo: país de nacimiento, marca de producto, categoría de producto. También pueden ser en forma de números, por ejemplo número que salió al tirar un dado.

Features

Las **features** o atributos son las características del dataset que vamos a utilizar para entrenar y predecir en los modelos.
Vamos a distinguir dos tipos de features:

- **Numéricas:** son características que tienen como valor números continuos. Ejemplo: edad, peso, salario. Generalmente las podemos utilizar directamente en los modelos.
- **Categóricas:** tienen como valor una categoría discreta, que puede ser en forma de string. Ejemplo: país de nacimiento, marca de producto, categoría de producto. También pueden ser en forma de números, por ejemplo número que salió al tirar un dado.

  En general vamos a tratarlos de manera diferente, haciendo un tipo de transformación para features numéricas y otro para categóricas.

Features numéricas

Al momento de tratar features numéricas, hay varias cosas que podemos hacer:

- Elegir cómo tratar missing values. Si faltan valores, tenemos que tomar una decisión. Podría ser descartar ese dato, o completar con el valor promedio de esa feature.

Features numéricas

Al momento de tratar features numéricas, hay varias cosas que podemos hacer:

- Elegir cómo tratar missing values. Si faltan valores, tenemos que tomar una decisión. Podría ser descartar ese dato, o completar con el valor promedio de esa feature.
- Estandarizar y normalizar valores. Esta técnica lleva los valores a una distribución con media 0 y desvío 1. Suele mejorar la performance de los modelos.

Features numéricas

Al momento de tratar features numéricas, hay varias cosas que podemos hacer:

- Elegir cómo tratar missing values. Si faltan valores, tenemos que tomar una decisión. Podría ser descartar ese dato, o completar con el valor promedio de esa feature.
- Estandarizar y normalizar valores. Esta técnica lleva los valores a una distribución con media 0 y desvío 1. Suele mejorar la performance de los modelos.
- Discretizar una variable numérica. Dependiendo del contexto, puede ser útil descartar la variable numérica y armar una feature categórica. Ejemplo: convertir de horario a categoría mañana, tarde y noche.

Features categóricas

En el caso de features categóricas, lo primero que debemos hacer es transformarla a alguna representación numérica para que el modelo pueda entender. Hay varias técnincas para hacer esto, la más conocida es **One Hot Encoding**.

- One Hot Encoding: técnica que transforma una variable categórica en un vector que representa a qué categoría pertenece.

Features categóricas

En el caso de features categóricas, lo primero que debemos hacer es transformarla a alguna representación numérica para que el modelo pueda entender. Hay varias técnincas para hacer esto, la más conocida es **One Hot Encoding**.

- One Hot Encoding: técnica que transforma una variable categórica en un vector que representa a qué categoría pertenece.

id	color
1	red
2	blue
3	green
4	blue

id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

Features categóricas

En el caso de features categóricas, lo primero que debemos hacer es transformarla a alguna representación numérica para que el modelo pueda entender. Hay varias técnincas para hacer esto, la más conocida es **One Hot Encoding**.

- One Hot Encoding: técnica que transforma una variable categórica en un vector que representa a qué categoría pertenece.
- Tratar missing values. Se pueden descartar los datos que faltan, o se puede reemplazar por la categoría más frecuente.

Kaggle

Kaggle es una página web que sirve para encontrar datasets y practicar resolver problemas con Machine Learning. Además suele tener ejemplos de código que suben los usuarios.

Para el ejercicio práctico vamos a utilizar el siguiente dataset: California Housing Prices. Contiene precios de casas de California del censo de 1990.

El objetivo es hacer un modelo que pueda predecir el precio de una casa

Notebook en Google Colab

Vamos a ejercitar esto en Google Colab con un Notebook:

1. Descargar e importar el dataset.
2. Análisis de datos.
3. Armado de pipeline de preprocesado.
4. Entrenar y evaluar modelos: **LinearRegression**, **DecisionTreeRegressor**, **RandomForestRegressor**, **XGBoost**.
5. Hacer *hyperparameter tunning* en el modelo **XGBoost** y ver si logramos mejorar la performance.