Principios de machine Learning

Universidad Nacional de Colombia
Profesor : Carlos Daniel Jiménez

2024

Contenido

- Conociendo el paquete de Scikit-learn
- Problemas de regresión
- Problemas de clasificación
- Mejorando nuestros modelos
- Preprocesamiento

Bibliografía recomendada

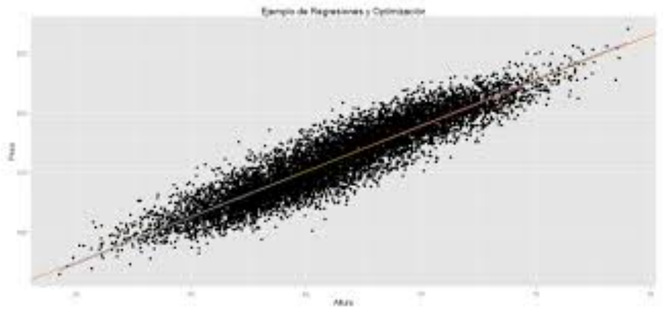
- Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems
- Scikit-learn Crash Course - Machine Learning Library for Python
- Predict childcare costs in US counties with xgboost and early stopping

Conociendo Scikit Learn

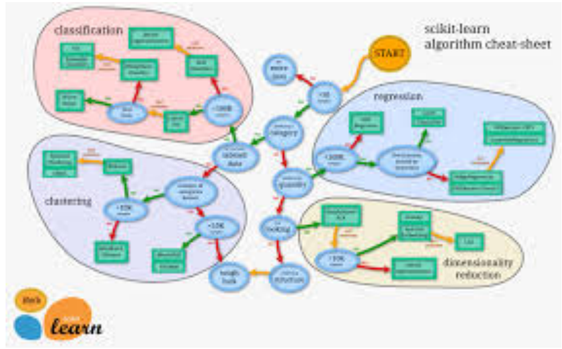
Documentación Oficial

- https://scikit-learn.org/stable/index.html
- https://matplotlib.org/
- https://seaborn.pydata.org/

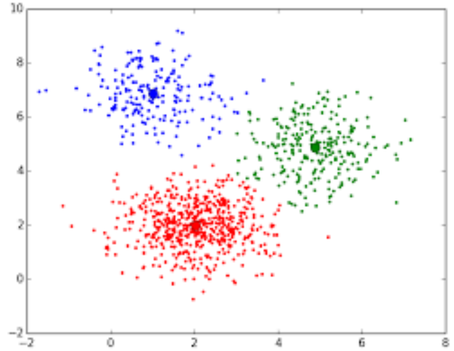
Conociendo Scikit Learn



Regresión



Clasificación



Clusterización

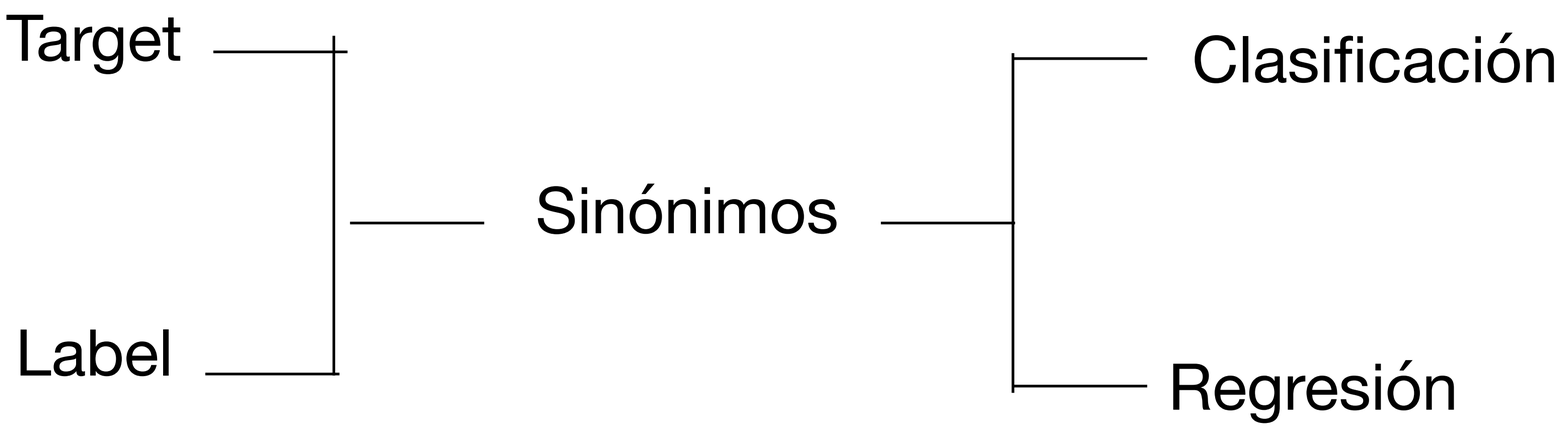
Conociendo Scikit-learn

Tipos de modelos

- Supervised Learning: Son aquellos problemas tabulares que incluyen el factor que se quiere solucionar, como un label dentro de una base de datos
  - Generalmente suelen ser asociados a problemas de clasificación , por ejemplo el típico caso de predecir si un correo es Spam o no
  - Cuando los problemas son del tipo predecir un valor , tienden a ser problemas supervisados del tipo regresión

Conociendo Scikit-learn

Tipos de modelos



Conociendo Scikit-learn

Tipos de modelos

- Unsupervised Learning: Son aquellos problemas donde no conocemos la variable que vamos a predecir o clasificar , como por ejemplo los clusters
  - Este tipo de modelos es muy común en problemas de dimensionalidad (censos por ejemplo, o predecir un error en un video)
  - Uno de sus usos más importantes tiene que ver con el mundo del procesamiento de lenguaje natural , la detección de anomalías , entre otros
  - Hay un tipo de caso muy interesante para los problemas no supervisados y tiene que ver con las reglas de asociación -> Descubir en grandes cantidades de datos las tracciones que existen entre lso atributos.

Conociendo Scikit-learn

Tipos de modelos

- Semi-supervised Learning: Cuando se trata de datos parcialmente etiquetados , por ejemplo como asocia google photos los nombres de las personas a través del feed del contenido en el celular

- Self Supervised: Generar un conjunto de datos completamente etiquetado a partir de uno completamente sin etiquetar (Deep Face)

- Reinforcement Learning: Es donde se crean agentes que aprenden a interactuar frente a problemas que se les imponga, un caso muy bueno de analizar es el tesla engine de los autos autónomos

Conociendo Scikit-learn

Tipos de modelos

- Algunas consideraciones para trabajar con estos modelos :
  - No deben tener valores ausentes
  - Los datos deben ser transformados en carácter numérico
  - Deben estar almacenados en valores matriciales

Conociendo Scikit-learn

Cómo funciona Scikit-learn

- Se debe importar un modelo

```
from sklearn.module import model
```

- Se crea un modelo ingenuo

```
model=Model()
```

- El modelo se ajusta a los datos

```
model.fit(x,y)
```

Conociendo Scikit-learn

Cómo funciona Scikit-learn

- Se hacen predicciones

```
pred= model.predict()
```

- Después hablaremos de como evaluar los modelos

Resolviendo problemas de clasificación con Scikit-learn

Conociendo Scikit-learn

Problemas de clasificación

- Pasos para resolver un problema de clasificación (iremos de lo fácil a lo complejo)
  - Se recolectan los datos
  - Se procesan los datos (recuerde que no pueden venir datos vacíos, y quizás toque hacerle un proceso de limpieza, donde nuestro mejor amigo será pandas)
  - Se construye un modelo
  - EL modelo aprende de los datos
  - Se hacen predicciones con los modelos y se evalúa cual es la mejor métrica

Conociendo Scikit-learn

Problemas de clasificación

Vamos a la Notebook #1

Conociendo Scikit-learn

Problemas de clasificación - Métricas

- Accuracy: en problemas de clasificación es una métrica que nos indica qué tan bien está funcionando nuestro modelo. Se define como la proporción de predicciones correctas hechas por el modelo sobre el total de predicciones. En otras palabras, mide el porcentaje de ejemplos que el modelo clasificó correctamente.

Conociendo Scikit-learn

Problemas de clasificación - Métricas

Conociendo Scikit-learn

Problemas de clasificación

- Algunos modelos tienen complejidades en sus parámetros y por lo cual hay que evaluar su poder de clasificación o predicción de manera interactiva , veamos un ejemplo en la notebook #1

```
train_accuracies = {}
test_accuracies = {}
neighbors = np.arange(1, 26)
for neighbor in neighbors:
    knn = KNeighborsClassifier(n_neighbors=neighbor)
    knn.fit(X_train, y_train)
    train_accuracies[neighbor] = knn.score(X_train, y_train)
    test_accuracies[neighbor] = knn.score(X_test, y_test)
```

Resolviendo problemas de regresión con Scikit-learn

Conociendo Scikit-learn

Modelos de regresión

- La regresión es una técnica de aprendizaje supervisado que se utiliza para predecir valores continuos. A diferencia de la clasificación, donde el objetivo es asignar etiquetas categóricas, en regresión el objetivo es predecir una variable de salida continua.
    - Predice el valor de la variable dependiente como una combinación lineal de las variables independientes.
        - Ideal para relaciones lineales entre la variable independiente y dependiente.

Conociendo Scikit-learn

Modelos de regresión

- Regresión Polinómica:
  - Extiende la regresión lineal para modelar relaciones no lineales.

- Regresión Ridge y Lasso:
  - Regresión Ridge (L2): Añade una penalización por la magnitud de los coeficientes para evitar el sobreajuste.
  - Regresión Lasso (L1): Añade una penalización que puede hacer que algunos coeficientes se reduzcan a cero, proporcionando así una forma de selección de características.

Conociendo Scikit-learn

Modelos de regresión

- Regresión con Soporte Vectorial (SVR):
  - Extiende el concepto de máquinas de soporte vectorial (SVM) a problemas de regresión.
    - Utiliza el margen y los vectores de soporte para predecir valores contiuos.

Conociendo Scikit-learn

Modelos de regresión

- Árboles de Decisión y Bosques Aleatorios para Regresión:
  - Árboles de decisión dividen los datos en subconjuntos basados en características y valores.
  - Bosques aleatorios combinan múltiples árboles de decisión para mejorar la precisión y reducir el sobreajuste.

Conociendo Scikit-learn

Modelos de regresión - Métricas

- Error Absoluto Medio (MAE):
  - Promedio de las diferencias absolutas entre las predicciones y los valores reales.
  - Cuando se prefiere una interpretación más directa de los errores sin penalizar más a los errores grandes.

Conociendo Scikit-learn

Modelos de regresión - Métricas

- Error Cuadrático Medio (MSE):
  - Promedio de los cuadrados de las diferencias entre las predicciones y los valores reales.
  - Cuando se desea penalizar más a los errores grandes, lo que puede ser útil en aplicaciones donde los grandes errores son particularmente indeseables.

Conociendo Scikit-learn

Modelos de regresión - Métricas

- Raíz del Error Cuadrático Medio (RMSE):
  - Raíz cuadrada del MSE.
  - Similar al MSE pero en la misma escala que los datos originales, lo que puede ser más interpretable.

Conociendo Scikit-learn

Modelos de regresión - Métricas

- Coeficiente de Determinación (R^2):
  - Medida de qué tan bien se ajustan las predicciones a los valores reales.
  - Para evaluar la proporción de la varianza en los datos de respuesta que es explicada por el modelo.

Ahora veamos como trabajar con problemas de regresión en la notebook # 1

Conociendo Scikit-learn

Modelos de regresión - Métricas

- Función de perdida
  - Las funciones de pérdida miden el error entre las predicciones del modelo y los valores reales. Son fundamentales para entrenar los modelos de regresión.
  - Tipos de funciones de perdida (ya las vimos):
    - Mae
    - RMSE
    - MSE
    - MAE

Conociendo Scikit-learn

Modelos de regresión - Métricas

- Función de costo
  - Las funciones de costo son el promedio de las pérdidas en el conjunto de datos de entrenamiento y se utilizan para guiar el proceso de entrenamiento del modelo, ajustando sus parámetros para minimizar el costo
  - Evaluar el rendimiento del modelo en el conjunto de entrenamiento.
  - Guía el algoritmo de optimización durante el entrenamiento.

Conociendo Scikit-learn

Modelos de regresión - Métricas

Ahora pasemos a la notebook # 1 para ver como se hace esto

Conociendo Scikit-learn

Cross Validation

- El cross validation (validación cruzada) es una técnica utilizada para evaluar el rendimiento de un modelo de aprendizaje automático de manera más robusta y fiable.
  - Ayuda a asegurar que el modelo se generaliza bien a datos no vistos, evitando el sobreajuste (overfitting) y el subajuste (underfitting).
  - Divide el conjunto de datos en K subconjuntos (folds) de tamaño aproximadamente igual.
  - Entrena el modelo $_k$ veces, cada vez utilizando `k-1` folds para el entrenamiento y 1 fold diferente para la validación.
  - Calcula la métrica de evaluación (por ejemplo, precisión) para cada fold y promedia los resultados.

Conociendo Scikit-learn

Cross Validation

- Proporciona una mejor estimación del rendimiento del modelo en datos no vistos.
- Reduce la variabilidad de las evaluaciones debido a la partición de datos específica.

Ahora veamos como aplicarlo en la Notebook#1