Instituto Politécnico Nacional

Escuela Superior de Cómputo

Ejemplos Scikit-Learn

Carlos Emiliano Sepúlveda Ramírez 5° IIA

Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"

Nombre de la materia: Machine Learning

Docente: Andres García Floriano

Ciudad de México a 18 de septiembre de 2024

Introducción

Este informe presenta una serie de códigos extraídos de la documentación de Scikit-lear utilizados como demostraciones de las capacidades de este módulo usando diferentes técnicas de machine learning en Python, con un enfoque en la regresión, clasificación y Clustering. Los códigos incluidos muestran cómo aplicar algoritmos avanzados como HistGradientBoostingRegressor para tareas de regresión cuantílica, modelos de regresión logística con preprocesamiento y selección de características, y el uso del algoritmo KMeans para la agrupación no supervisada de datos. A lo largo del reporte, se destacan tanto el procesamiento de los datos como la visualización de los resultados, lo que permite analizar de manera efectiva el comportamiento de los modelos en distintos escenarios. Cada sección del informe incluye una explicación detallada de los métodos utilizados y gráficos que visualizan los resultados obtenidos.

Código #1: HistGradientBoostingRegressor y Regresión Cuantil

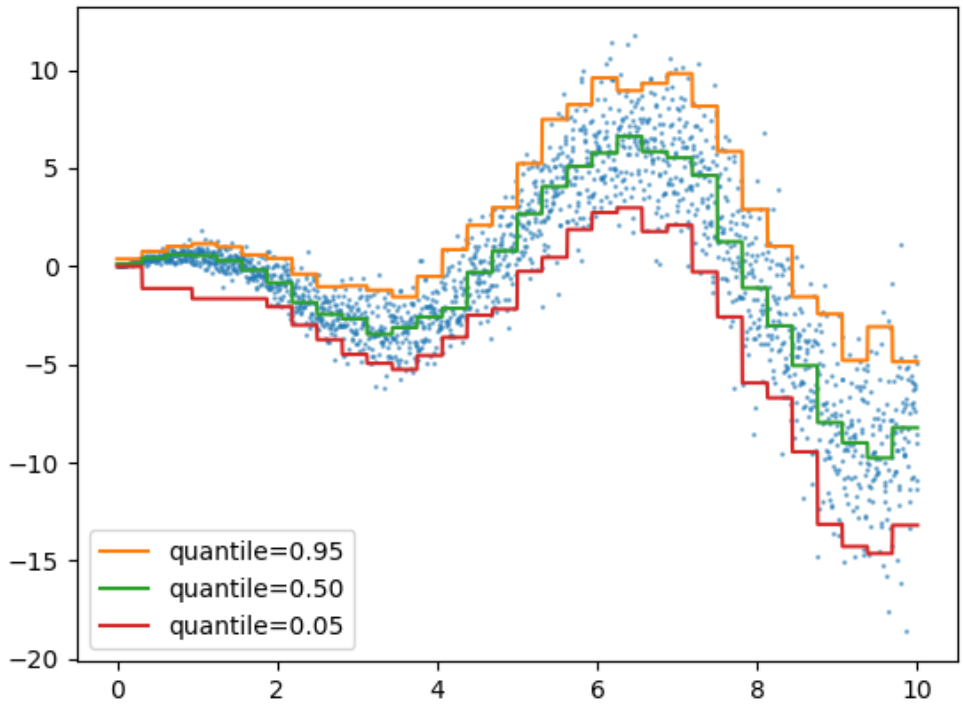
Regresión cuantil

La regresión cuantil es una técnica estadística que permite modelar la relación entre las variables predictoras y los cuantiles específicos de la variable dependiente. A diferencia de la regresión lineal, que se centra en predecir la media condicional de la variable dependiente, la regresión cuantil predice distintos cuantiles, lo que permite una mejor comprensión de la variabilidad de los datos. Esto es especialmente útil en contextos donde no se desea un ajuste medio, sino una estimación de extremos o diferentes puntos en la distribución de los datos.

Por ejemplo, en el análisis de riesgos financieros, los modelos de regresión cuantil son útiles para predecir pérdidas extremas, y en el campo de la medicina, para estimar los efectos de un tratamiento en la cola superior o inferior de una distribución de resultados.

Características del Código:

1. **Generación de datos**: Los datos son generados usando una función no lineal $X \cdot cos\ (X)$ con ruido. Esto simula un escenario donde los datos tienen variabilidad y ruido, un caso común en aplicaciones reales.
2. **Modelos de regresión cuantil**: Se entrenan tres modelos de regresión cuantil utilizando HistGradientBoostingRegressor para los cuantiles 0.95, 0.5 (mediana) y 0.05. Estos tres modelos permiten observar cómo varía la relación entre las variables predictoras y la variable objetivo en diferentes puntos de la distribución.



Se muestra un gráfico que compara los puntos de datos originales con las predicciones de los tres cuantiles, lo que proporciona una visión clara de cómo los modelos capturan la estructura de los datos en distintos niveles.

Regresión Logística, Preprocesamiento y Selección de Características

La regresión logística es una técnica de clasificación que modela la probabilidad de que una observación pertenezca a una de dos clases. Se basa en la función logística y se utiliza ampliamente en problemas de clasificación binaria. Este modelo es especialmente útil cuando el objetivo es predecir probabilidades de eventos binarios, como la supervivencia o no de los pasajeros en el Titanic, que es el conjunto de datos utilizado en este ejemplo.

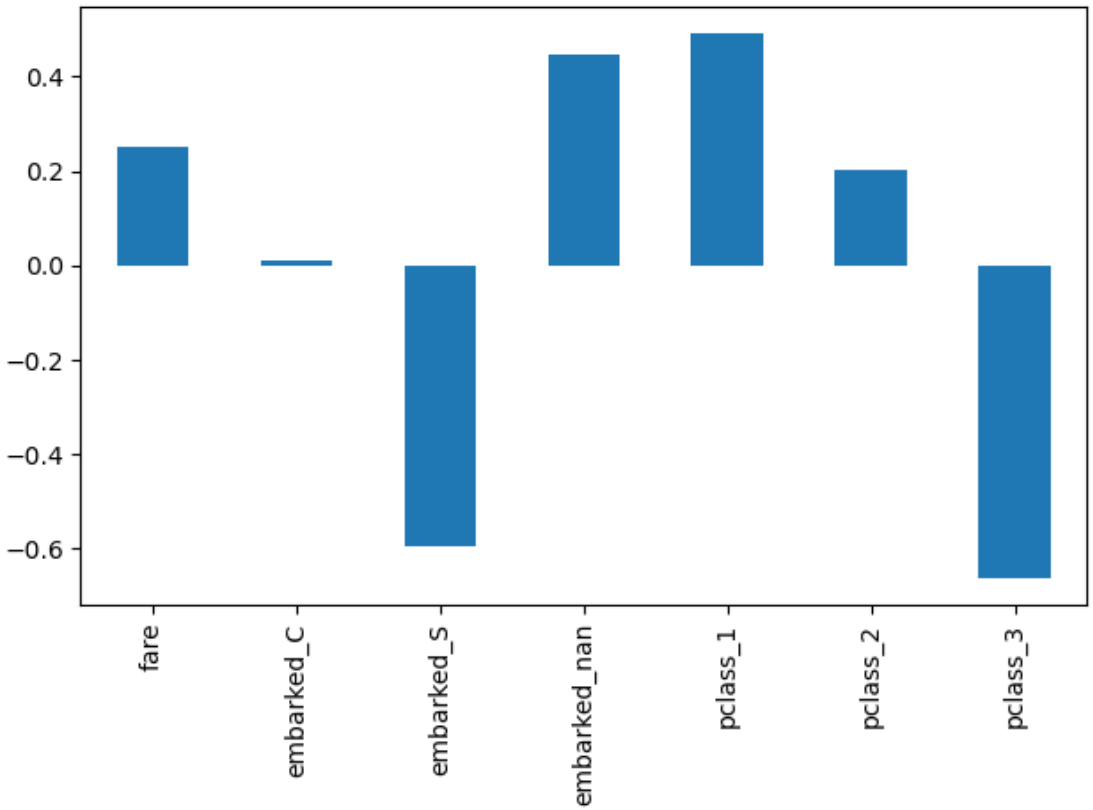
Además, el preprocesamiento de datos es un paso crucial en cualquier pipeline de machine learning. Este proceso puede incluir la imputación de valores faltantes, la normalización de características y la codificación de variables categóricas, todos los cuales son esenciales para que los algoritmos funcionen correctamente con los datos reales. La selección de características también juega un papel importante, ya que ayuda a identificar y retener solo las variables más relevantes para mejorar la eficiencia y el rendimiento del modelo.

Características del Código:

1. **Preprocesamiento**:

- **Imputación y escalado**: Las variables numéricas (como "age" y "fare") se imputan con la mediana para tratar valores faltantes y luego se escalan para evitar que una variable domine sobre otra en términos de magnitud.
- **Codificación categórica**: Las variables categóricas (como "embarked" y "pclass") se convierten en variables binarias utilizando OneHotEncoder, lo que permite que el modelo trabaje con información cualitativa.

2. **Selección de características**: El código utiliza SelectKBest para elegir las 7 características más importantes, lo que ayuda a reducir la dimensionalidad y mejora la eficiencia del modelo de regresión logística.



Los coeficientes del modelo de regresión logística se visualizan en un gráfico de barras, lo que muestra la importancia relativa de cada característica seleccionada.

Código #2: Clustering con KMeans en el conjunto de datos Iris

Clustering y Algoritmo Kmeans

El clustering es una técnica de aprendizaje no supervisado utilizada para agrupar observaciones en grupos (clusters) según su similitud. El algoritmo KMeans es uno de los métodos más populares para el clustering. Su objetivo es dividir los datos en

k grupos, minimizando la variación dentro de los clusters. Cada observación se asigna al cluster más cercano a su centroide, y los centroides se ajustan iterativamente para mejorar la agrupación.

El conjunto de datos Iris es un clásico en machine learning, que contiene tres clases de flores (Setosa, Versicolor, Virginica), con características como el largo y ancho de los pétalos y sépalos. Al aplicar clustering en este conjunto de datos, se puede observar cómo el algoritmo agrupa las observaciones basándose en las similitudes de las características de las flores, independientemente de las etiquetas.

Aquí tienes una versión más detallada del reporte, en la que se habla primero del tema abordado en cada código, seguido de las características específicas de cada uno.

Código 1: HistGradientBoostingRegressor y Regresión Cuantil

Tema Abordado: Regresión Cuantil

La regresión cuantil es una técnica estadística que permite modelar la relación entre las variables predictoras y los cuantiles específicos de la variable dependiente. A diferencia de la regresión lineal, que se centra en predecir la media condicional de la variable dependiente, la regresión cuantil predice distintos cuantiles, lo que permite una mejor comprensión de la variabilidad de los datos. Esto es especialmente útil en contextos donde no se desea un ajuste medio, sino una estimación de extremos o diferentes puntos en la distribución de los datos.

Por ejemplo, en el análisis de riesgos financieros, los modelos de regresión cuantil son útiles para predecir pérdidas extremas, y en el campo de la medicina, para estimar los efectos de un tratamiento en la cola superior o inferior de una distribución de resultados.

Características del Código:

1. **Generación de datos: Los datos son generados usando una función no lineal X·cos⁡(X)X \cdot \cos(X)X·cos(X) con ruido. Esto simula un escenario donde los datos tienen variabilidad y ruido, un caso común en aplicaciones reales.**
2. **Modelos de regresión cuantil: Se entrenan tres modelos de regresión cuantil utilizando HistGradientBoostingRegressor para los cuantiles 0.95, 0.5 (mediana) y 0.05. Estos tres modelos permiten observar cómo varía la relación entre las variables**

predictoras y la variable objetivo en diferentes puntos de la distribución.

3. **Visualización: Se muestra un gráfico que compara los puntos de datos originales con las predicciones de los tres cuantiles, lo que proporciona una visión clara de cómo los modelos capturan la estructura de los datos en distintos niveles.**

Código 2: Regresión Logística, Preprocesamiento y Selección de Características

Tema Abordado: Regresión Logística y Preprocesamiento de Datos

La regresión logística es una técnica de clasificación que modela la probabilidad de que una observación pertenezca a una de dos clases. Se basa en la función logística y se utiliza ampliamente en problemas de clasificación binaria. Este modelo es especialmente útil cuando el objetivo es predecir probabilidades de eventos binarios, como la supervivencia o no de los pasajeros en el Titanic, que es el conjunto de datos utilizado en este ejemplo.

Además, el preprocesamiento de datos es un paso crucial en cualquier pipeline de machine learning. Este proceso puede incluir la imputación de valores faltantes, la normalización de características y la codificación de variables categóricas, todos los cuales son esenciales para que los algoritmos funcionen correctamente con los datos reales. La selección de características también juega un papel importante, ya que ayuda a identificar y retener solo las variables más relevantes para mejorar la eficiencia y el rendimiento del modelo.

Características del Código:

1. Preprocesamiento:
    - Imputación y escalado: Las variables numéricas (como "age" y "fare") se imputan con la mediana para tratar valores faltantes y luego se escalan para evitar que una variable domine sobre otra en términos de magnitud.
    - Codificación categórica: Las variables categóricas (como "embarked" y "pclass") se convierten en variables binarias utilizando OneHotEncoder, lo que permite que el modelo trabaje con información cualitativa.
2. Selección de características: El código utiliza SelectKBest para elegir las 7 características más importantes, lo que ayuda a reducir la dimensionalidad y mejora la eficiencia del modelo de regresión logística.
3. Visualización: Los coeficientes del modelo de regresión logística se visualizan en un gráfico de barras, lo que muestra la importancia relativa de cada característica seleccionada.

Código 3: Clustering con KMeans en el conjunto de datos Iris

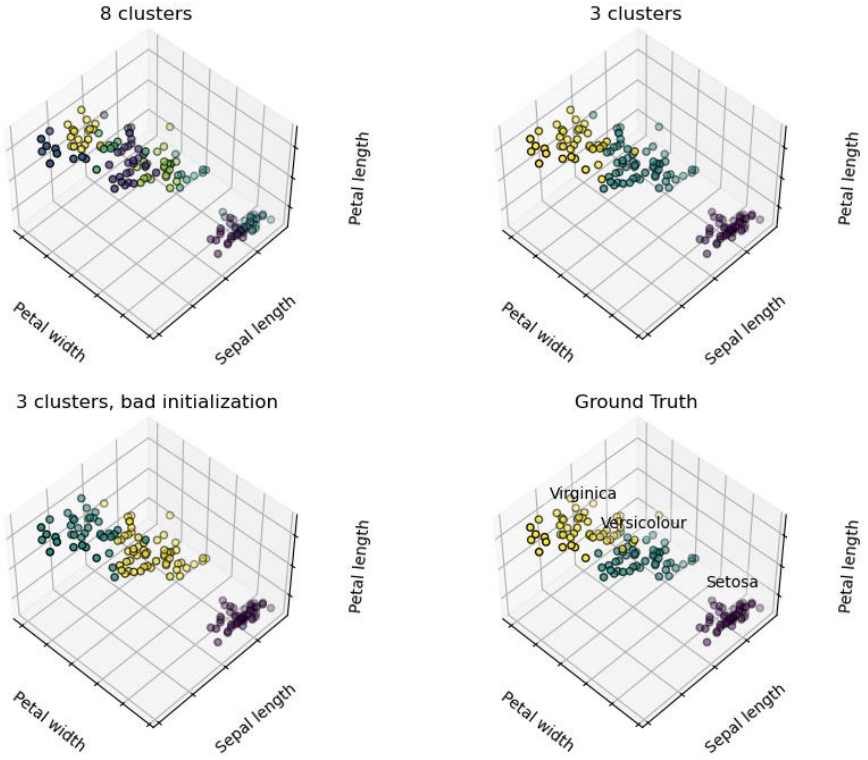
Tema Abordado: Clustering y Algoritmo KMeans

El clustering es una técnica de aprendizaje no supervisado utilizada para agrupar observaciones en grupos (clusters) según su similitud. El algoritmo KMeans es uno de los métodos más populares para el clustering. Su objetivo es dividir los datos en kkk grupos, minimizando la variación dentro de los clusters. Cada observación se asigna al cluster más cercano a su centroide, y los centroides se ajustan iterativamente para mejorar la agrupación.

El conjunto de datos Iris es un clásico en machine learning, que contiene tres clases de flores (Setosa, Versicolor, Virginica), con características como el largo y ancho de los pétalos y sépalos. Al aplicar clustering en este conjunto de datos, se puede observar cómo el algoritmo agrupa las observaciones basándose en las similitudes de las características de las flores, independientemente de las etiquetas.

Características del Código:

1. Configuraciones de KMeans:
    - 8 clusters: Se aplica KMeans con 8 clusters, lo que resulta en una sobre-segmentación de los datos, proporcionando más grupos de los que realmente existen.
    - 3 clusters: Se ejecuta KMeans con 3 clusters, que corresponde al número real de clases en el conjunto de datos Iris, para observar si el algoritmo es capaz de agrupar las flores de manera coherente con las especies.
    - Mala inicialización: Se prueba una configuración con una mala inicialización (solo una iteración y una inicialización aleatoria), lo que muestra el impacto de una mala configuración en el rendimiento del algoritmo.



2. Visualización:
    - Se generan gráficos 3D que muestran cómo el algoritmo agrupa las observaciones para cada configuración.
    - Finalmente, se compara con el gráfico que muestra las etiquetas reales del conjunto de datos, lo que permite evaluar qué tan bien o mal el algoritmo logró agrupar las observaciones en relación con las clases reales.

Código 4: Clustering con KMeans y Reducción de Dimensionalidad con PCA

Tema Abordado: Clustering con KMeans y Reducción de Dimensionalidad

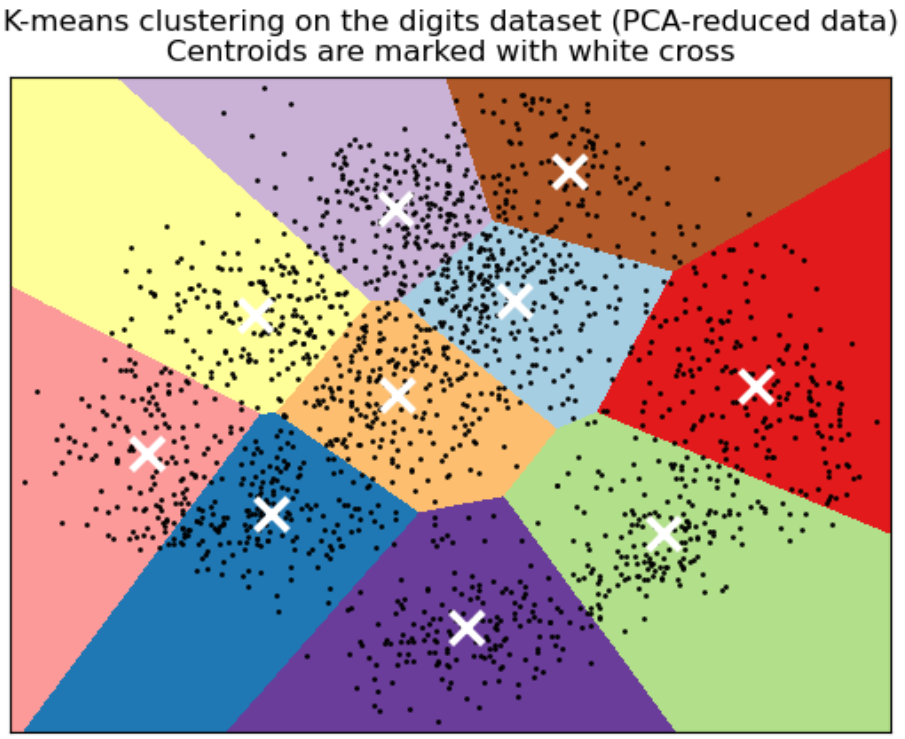
Este código aborda el uso del algoritmo KMeans para realizar clustering en el conjunto de datos de dígitos (digits) de sklearn. Los dígitos son un conjunto de imágenes de 8x8 píxeles que representan números escritos a mano, con etiquetas que van del 0 al 9. El clustering es utilizado para agrupar los datos sin considerar sus etiquetas, con el fin de evaluar la capacidad del algoritmo de identificar patrones similares entre los dígitos.

Además, el código utiliza la reducción de dimensionalidad con PCA (Análisis de Componentes Principales), que es una técnica para proyectar datos de alta

dimensión a un espacio de menor dimensión (en este caso, a 2 componentes principales). PCA es útil para la visualización de datos y para mejorar la eficiencia del modelo en problemas de clustering cuando los datos tienen muchas características.

Características del Código:

1. Cálculo de métricas: Se utiliza una función personalizada para evaluar el rendimiento del algoritmo KMeans con diferentes métodos de inicialización (k-means++, aleatorio, basado en PCA). Las métricas calculadas incluyen la homogeneidad, completitud, V-measure, ARI, AMI y el coeficiente de silueta, todas diseñadas para medir qué tan bien los clusters encontrados corresponden a las etiquetas originales.
2. Comparación de inicializaciones: Se prueban tres estrategias de inicialización:
    - k-means++: Un método optimizado que mejora la convergencia.
    - Inicialización aleatoria: Donde los centroides iniciales son seleccionados de manera aleatoria.
    - Inicialización basada en PCA: Donde se utilizan las componentes principales obtenidas de PCA para inicializar los centroides.
3. Reducción de dimensionalidad para visualización: El código también realiza una reducción de dimensionalidad a dos componentes principales mediante PCA para visualizar los resultados de clustering en 2D.



4. Visualización de fronteras de decisión: Se traza un gráfico de las fronteras de decisión del modelo KMeans en el espacio 2D reducido por PCA, donde los centroides de cada clúster se marcan con una cruz blanca. Esto proporciona una representación clara de cómo el algoritmo KMeans agrupa los datos.

Conclusión General del Reporte

A lo largo de este reporte, hemos explorado distintas técnicas de machine learning relacionadas con el clustering, la regresión cuantílica y la preprocesamiento de datos. Estos conceptos son fundamentales en análisis de datos y modelado predictivo, permitiendo no solo la clasificación o agrupación de datos, sino también el ajuste preciso a distribuciones y la correcta manipulación de datos complejos.

A través de los códigos presentados, se demostró cómo el clustering KMeans, con sus diversas inicializaciones, puede adaptarse a diferentes conjuntos de datos para agrupar elementos en categorías significativas. Además, se observó cómo la reducción de dimensionalidad mediante PCA facilita la visualización y mejora la eficiencia en el agrupamiento, revelando patrones ocultos en los datos originales.

En el contexto de la regresión cuantílica, vimos cómo predecir diferentes cuantiles permite modelar de manera más robusta la incertidumbre en los datos y las posibles distribuciones de resultados, lo que es clave en aplicaciones donde los valores extremos o las probabilidades de eventos son relevantes.

Finalmente, el preprocesamiento de datos, como el manejo de variables categóricas y numéricas, es un componente esencial en la construcción de modelos eficaces. Herramientas como la imputación de valores faltantes, el escalado de características y la codificación categórica fueron fundamentales en el pipeline para lograr modelos que puedan generalizar bien y producir predicciones precisas.

En conclusión, el conocimiento y la aplicación práctica de estas técnicas no solo permiten una mejor comprensión de los datos, sino que son clave para construir modelos de machine learning más eficientes y precisos, lo que es esencial en un mundo cada vez más impulsado por la inteligencia artificial y la analítica avanzada.