

Descripción del proyecto

Rusty Bargain es un servicio de venta de coches de segunda mano que está desarrollando una app para atraer a nuevos clientes. Gracias a esa app, puedes averiguar rápidamente el valor de mercado de tu coche. Tienes acceso al historial, especificaciones técnicas, versiones de equipamiento y precios. Tienes que crear un modelo que determine el valor de mercado.

A Rusty Bargain le interesa:

- la calidad de la predicción
- la velocidad de la predicción
- el tiempo requerido para el entrenamiento

Instrucciones del proyecto

1. Descarga y examina los datos.
2. Entrena diferentes modelos con varios hiperparámetros (debes hacer al menos dos modelos diferentes, pero más es mejor. Recuerda, varias implementaciones de potenciación del gradiente no cuentan como modelos diferentes). El punto principal de este paso es comparar métodos de potenciación del gradiente con bosque aleatorio, árbol de decisión y regresión lineal.
3. Analiza la velocidad y la calidad de los modelos.

Observaciones:

- Utiliza la métrica RECM para evaluar los modelos.
- La regresión lineal no es muy buena para el ajuste de hiperparámetros, pero es perfecta para hacer una prueba de cordura de otros métodos. Si la potenciación del gradiente funciona peor que la regresión lineal, definitivamente algo salió mal.
- Aprende por tu propia cuenta sobre la librería LightGBM y sus herramientas para crear modelos de potenciación del gradiente (gradient boosting).
- Idealmente, tu proyecto debe tener regresión lineal para una prueba de cordura, un algoritmo basado en árbol con ajuste de hiperparámetros (preferiblemente, bosque

aleatorio), LightGBM con ajuste de hiperparámetros (prueba un par de conjuntos), y CatBoost y XGBoost con ajuste de hiperparámetros (opcional).

- Toma nota de la codificación de características categóricas para algoritmos simples. LightGBM y CatBoost tienen su implementación, pero XGBoost requiere OHE.
- Puedes usar un comando especial para encontrar el tiempo de ejecución del código de celda en Jupyter Notebook. Encuentra ese comando.
- Dado que el entrenamiento de un modelo de potenciación del gradiente puede llevar mucho tiempo, cambia solo algunos parámetros del modelo.
- Si Jupyter Notebook deja de funcionar, elimina las variables excesivas por medio del operador del: `del features_train`

Descripción de los datos

Características

- *DateCrawled* — fecha en la que se descargó el perfil de la base de datos
- *VehicleType* — tipo de carrocería del vehículo
- *RegistrationYear* — año de matriculación del vehículo
- *Gearbox* — tipo de caja de cambios
- *Power* — potencia (CV)
- *Model* — modelo del vehículo
- *Mileage* — kilometraje (medido en km de acuerdo con las especificidades regionales del conjunto de datos)
- *RegistrationMonth* — mes de matriculación del vehículo
- *FuelType* — tipo de combustible
- *Brand* — marca del vehículo
- *NotRepaired* — vehículo con o sin reparación
- *DateCreated* — fecha de creación del perfil
- *NumberOfPictures* — número de fotos del vehículo

- *PostalCode* — código postal del propietario del perfil (usuario)
- *LastSeen* — fecha de la última vez que el usuario estuvo activo

Objetivo

Price — precio (en euros)

Evaluación del proyecto

Hemos definido los criterios de evaluación para el proyecto. Léelos con atención antes de pasar al ejercicio.

Esto es en lo que se fijarán los revisores al examinar tu proyecto:

- ¿Seguiste todos los pasos de las instrucciones?
- ¿Cómo preparaste los datos?
- ¿Qué modelos e hiperparámetros consideraste?
- ¿Conseguiste evitar la duplicación del código?
- ¿Cuáles son tus hallazgos?
- ¿Mantuviste la estructura del proyecto?
- ¿Mantuviste el código ordenado?