**Descrição do projeto**

Rusty Bargain é um serviço de venda de carros usados que está desenvolvendo um aplicativo para atrair novos clientes. Nesse aplicativo, você pode descobrir rapidamente o valor de mercado do seu carro. Você tem acesso a dados históricos, especificações técnicas, versões de acabamento e preços. Você precisa construir o modelo para determinar o valor.

Rusty Bargain está interessado em:

* a qualidade da predição
* a velocidade da predição
* o tempo necessário para o treinamento

**Instruções do projeto**

1. Baixe e veja os dados.
2. Treine modelos diferentes com vários hiperparâmetros (Você deve fazer pelo menos dois modelos diferentes, mas quanto mais, melhor. Lembre-se, várias implementações do gradient boosting não contam como modelos diferentes.) O ponto principal desta etapa é comparar métodos de gradient boosting com floresta aleatória, árvore de decisão e regressão linear.
3. Analisar a velocidade e a qualidade dos modelos.

Notas:

* Use a métrica *REQM* para avaliar os modelos.
* A regressão linear não é muito boa para ajuste de hiperparâmetros, mas é perfeita para tirar a prova real de outros métodos. Se o gradient boosting funciona pior do que a regressão linear, algo definitivamente deu errado.
* Por conta própria, domine a biblioteca LightGBM e use suas ferramentas para criar modelos de gradient boosting.
* Idealmente, seu projeto deve incluir regressão linear para tirar a prova real, um algoritmo baseado em árvore com ajuste de hiperparâmetro (de preferência, floresta aleatória), LightGBM com ajuste de hiperparâmetro (tente alguns conjuntos) e CatBoost e XGBoost com ajuste de hiperparâmetro (opcional).
* Tome nota da codificação de características categóricas para algoritmos simples. LightGBM e CatBoost têm sua implementação, mas o XGBoost requer OHE.
* Você pode usar um comando especial para encontrar o tempo de execução do código da célula no Jupyter Notebook. Encontre esse comando.
* Como o treinamento de um modelo de gradient boosting pode levar muito tempo, altere apenas alguns parâmetros do modelo.
* Se o Jupyter Notebook parar de funcionar, exclua as variáveis em excesso usando o operador del:
* del features\_train

**Descrição de dados**

O conjunto de dados é armazenado no arquivo /datasets/car\_data.csv . Clique aqui para [baixar o conjunto de dados](https://practicum-content.s3.us-west-1.amazonaws.com/datasets/car_data.csv).

Características

* *DateCrawled* — data em que o perfil foi baixado do banco de dados
* *VehicleType* — tipo de carroçaria do veículo
* *RegistrationYear* — ano de matrícula do veículo
* *Gearbox* — tipo de caixa de transmissão
* *Power* — potência (hp)
* *Model* — modelo do veículo
* Mileage — quilometragem (medida em km devido às especificidades regionais do conjunto de dados)
* *RegistrationMonth* — mês de registro do veículo
* *FuelType* — tipo de combustível
* *Brand* — marca do veículo
* *NotRepaired* — veículo reparado ou não
* *DateCreated* — data de criação do perfil
* *NumberOfPictures* — número de fotos do veículo
* *PostalCode* — código postal do proprietário do perfil (usuário)
* *LastSeen* — data da última atividade do usuário

Objetivo

*Price* — preço (Euro)