Introdução ao aprendizado de máquina: Projeto início até o fim

Oscar J. O. Ayala

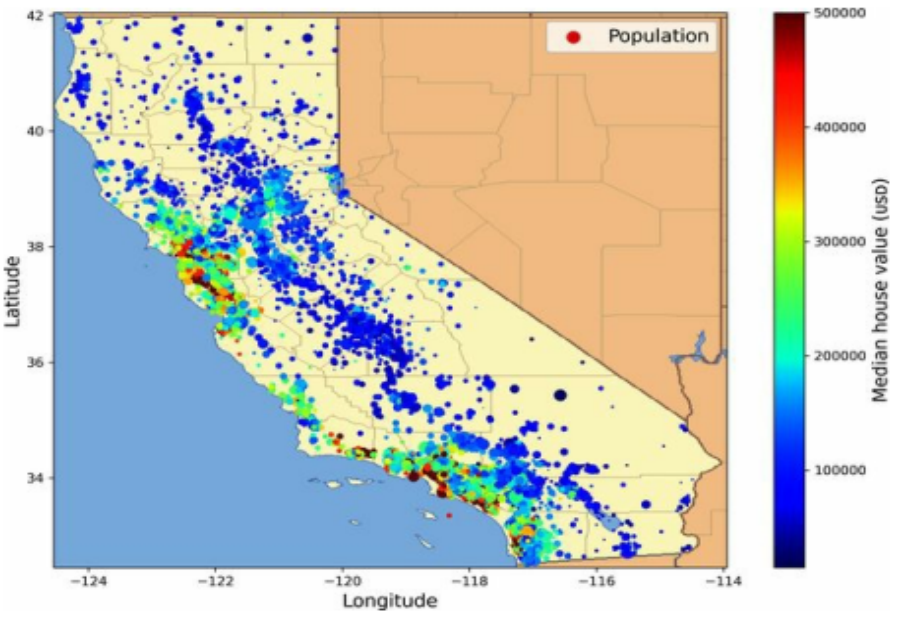
Guia para projetos de aprendizado de máquina

- Defina o desafio e olhe para o quadro geral.
- Obtenha os dados (criar objeto `dataset`)
- Análises descritiva.
- Definir conjunto de treinamento e teste (trainSet, testSet)
- Explore os dados para obter insights (`dataset = trainSet.copy()`)
  - Visualizando dados geográficos (opcional)
  - Correlações
  - Experimentar combinações (opcional)
- Prepare os dados para expor melhor os padrões subjacentes aos algoritmos de aprendizado de máquina (`dataset = trainSet.drop("target", axis = 1).copy()`, `target = strat_trainSet[["target"]].copy()`)
  - Tratar campos redundantes
  - Tratar valores ausentes
  - Tratar nao consistencias
  - Lidar com textos categóricos

- Explore muitos modelos diferentes e liste os melhores (Usa-se apenas o conjunto *training*)
    - Criar classes de estimadores *Pipeline* para realizar o pré-processamento e ajuste do modelo.
    - Avaliação cruzada dos modelos propostos, obtendo uma lista de modelos candidatos.
    - *GridSearchCV*: se os hiper-parâmetros tivessem um espaço paramétrico com alguns pontos.
    - *RandomizedSearchCV*: se os hiper-parâmetros tivessem um espaço paramétrico muito grande, infinito numerável ou infinito não numerável.
- Aperfeiçoe seus modelos e combine-os em uma ótima solução.
- Analises dos melhores modelos e seus erros.
    - Se cria o objeto com o modelo final (`final_model`)
    - Analise as estimativa dos parâmetros, para realizar possíveis reduções.
- Avaliar seu sistema no conjunto de testes.
- Apresente sua solução.
- Lance, monitorize e mantenha seu sistem:
    - Salvar todos os modelos tentado, para que se volte facilmente a qualquer modelo que se desejar.
    - Também salvar as pontuações de validação cruzada e talvez as previsões reais no conjunto de validação. Isso permitirá comparar facilmente as pontuações entre diferentes tipos de modelo e compare os tipos de erros que eles cometem.

1. Definição do desafio: *California Housing Prices*.

Considere o conjunto de dados *California Housing Prices* do repositório StatLib. Este conjunto de dados é baseado em dados do censo da Califórnia de 1990. Não é exatamente recente (uma casa agradável na região da Baía ainda era acessível na época), mas possui muitas qualidades para aprendizado, então fingiremos que são dados re
e removeram alguns atributos. Na
dos dados.



1. Quadro geral do desafio: perguntas

- Como a empresa espera usar e se beneficiar desse modelo? (Saber o objetivo é importante porque determinará como se enquadra o problema, quais algoritmos seleccionará, qual medida de desempenho usará para avaliar seu modelo e quanto esforço se dedicará para ajustá-lo).
- Como é a solução atual (se houver)? (A situação atual frequentemente fornecerá uma referência para o desempenho, bem como insights sobre como resolver o problema).

1. Quadro geral do desafio: respostas

- Como a empresa espera usar e se beneficiar desse modelo? A saída do modelo (uma previsão do preço médio da habitação de um distrito) será alimentada a outro sistema de aprendizado de máquina, juntamente com muitos outros sinais. Esse sistema *downstream* determinará se vale a pena investir em uma determinada área.
- Como é a solução atual (se houver)? Uma equipe reúne informações atualizadas sobre um distrito e, quando não consegue obter o preço médio da habitação, o estima usando regras complexas.

1. Quadro geral do desafio: sistema, tarefa e ténica de *ML*

Com base nas informações fornecidas anteriormente, é possível responder às seguintes perguntas:

- Qual tipo de supervisão de treinamento será necessário para o modelo? Será necessário um sistema de *ML* supervisionado, uma vez que cada instância ou registro vem com o preço ou *target* (metas).
- Qual será a tarefa desempenhada pelo modelo? A tarefa é de previsão dos preços das casas através de um modelo de regressão múltivariável.
- Deve-se utilizar técnicas de aprendizado em lote ou em tempo real? Em lote, dado que os preços das casas não mudam rapidamente e o tamanho dos dados são pequenos.

Obtenha os dados

- Se usa a linguagem de programação `Python` versão `3.9.13`.
- Se usa o ambiente de Jupyter Notebook. Outra opção é o ambiente de Google Colab, um serviço gratuito que permite executar qualquer `notebook Jupyter` diretamente online, sem precisar instalar nada em sua máquina.
- Se cria uma função `Python` que abra os dados automaticamente ou os procure em um endereço requerido, veja:

```
# liberarias e funções
from pathlib import Path
import pandas as pd
import tarfile
import urllib.request
```

Obtenha os dados

```
# Função para Obter dados
def dados_alojamento():
    tarball_path = Path("datasets/housing.tgz")                    # ca
    Path("datasets").mkdir(parents=True, exist_ok=True)            # cr
    url = "https://github.com/ageron/data/raw/main/housing.tgz"    # cr
    urllib.request.urlretrieve(url, tarball_path)                  # De
    with tarfile.open(tarball_path) as housing_tarball:            # ab
        housing_tarball.extractall(path="datasets")                # ex
    return pd.read_csv(Path("datasets/housing/housing.csv"))       # re

alojamento = dados_alojamento()                                    # ca
```