

Mapeamento de estações de carros elétricos.

1. Escopo do Projeto

O **EcoDrive** é um sistema de planejamento de rotas para motoristas de veículos elétricos, com o objetivo de otimizar a utilização das estações de carregamento e minimizar o impacto da baixa autonomia dos veículos. O projeto utiliza Machine Learning para fornecer informações precisas e relevantes aos motoristas, como:

- Predição do tempo necessário para o carregamento do veículo.
- Identificação do tipo de carregador mais adequado.
- Estimativa da autonomia restante do veículo para planejamento de trajetos.

2. Ferramentas Utilizadas

2.1 Python

Por quê? Python foi escolhido devido à sua versatilidade, ampla adoção em projetos de Machine Learning e rica coleção de bibliotecas.

2.2 Bibliotecas para Manipulação de Dados

- **Pandas:**

- Função: Manipulação e limpeza de dados tabulares.
 - Uso no projeto: Tratamento de valores ausentes, criação de novas variáveis (como o tempo real de carregamento) e transformação de dados categóricos em variáveis numéricas.
- **NumPy:**
 - Função: Suporte para cálculos matemáticos e manipulação de arrays.
 - Uso no projeto: Processamento eficiente de dados numéricos, como cálculo de médias para preenchimento de valores ausentes.
-

2.3 Bibliotecas para Modelagem de Machine Learning

- **Scikit-learn:**
 - Função: Fornece ferramentas robustas para aprendizado supervisionado e não supervisionado.
 - Uso no projeto:
 - **Modelos:**
 - **RandomForestRegressor**: Utilizado para prever o tempo de carregamento e a autonomia restante devido à sua capacidade de modelar relações não-lineares e lidar bem com dados tabulares.
 - **RandomForestClassifier**: Escolhido para classificar o tipo de carregador necessário, devido à sua alta acurácia em tarefas de classificação.
 - **Métricas:**
 - **mean_absolute_error**: Avaliar a precisão de modelos de regressão.
 - **accuracy_score** e **confusion_matrix**: Medir o desempenho do modelo de classificação.
 - **Divisão de dados:**



- `train_test_split`: Separar os dados em conjuntos de treino e teste, garantindo a avaliação confiável dos modelos.
-

2.4 Bibliotecas para Visualização de Dados

- **Matplotlib:**
 - Função: Criação de gráficos simples e detalhados.
 - Uso no projeto: Visualizar comparações entre previsões e valores reais, como:
 - Gráficos de barras para comparação do tempo de carregamento.
 - Gráficos de linha para analisar a autonomia restante.
 - **Seaborn:**
 - Função: Extensão de Matplotlib para visualizações mais avançadas e estilizadas.
 - Uso no projeto: Representação gráfica da importância das variáveis e análise dos resultados do modelo.
-

3. Fluxo do Projeto

1. **Carregamento dos Dados:**
 - Importamos o conjunto de dados para o ambiente Python e verificamos sua integridade.
2. **Pré-processamento dos Dados:**
 - Lidamos com valores ausentes e transformamos variáveis categóricas em numéricas.
 - Criamos novas colunas derivadas para enriquecer a análise.
3. **Treinamento dos Modelos:**
 - Utilizamos `RandomForestRegressor` e `RandomForestClassifier` para modelar o comportamento dos dados.
 - Dividimos os dados em conjuntos de treino e teste para validação.



4. Visualização e Interpretação:

- Geramos gráficos para ilustrar os resultados e evidenciar o impacto do sistema para o usuário final.

4. Considerações Técnicas

O uso dessas ferramentas permitiu:

- Um pipeline consistente de tratamento de dados e treinamento de modelos.
- Análises gráficas detalhadas que facilitam a compreensão do impacto do aplicativo.
- Modelos robustos que podem ser integrados diretamente ao sistema do EcoDrive para melhorar a experiência dos motoristas.

Essa documentação serve como base para o entendimento técnico do projeto, facilitando futuras implementações ou melhorias.

Link para pitch de apresentação do projeto:

<https://youtu.be/HrFmGBe4v9A>

