FIAP - Faculdade de Informática e

Administração Paulista 2TDSPR

Grupo KCIAO

RM 553471 - Gustavo Vieira Bargas
RM 553791 - Jhonatan Sampaio Ferreira
RM 553169 - Vivian Sy Ting Wu

Global Solution

2024

Relatório Técnico: Previsão de Gasto Mensal com Energia

Descrição do Problema

O objetivo deste trabalho foi criar um modelo de Machine Learning para prever o gasto mensal com energia de diferentes endereços, utilizando dados de consumo e características relacionadas à produção de energia solar. Essa previsão é útil para otimizar a gestão de recursos energéticos, identificar padrões de consumo e fornecer informações para iniciativas de economia.

Os dados utilizados foram extraídos de arquivos JSON que continham informações sobre:

- Usuários: Nome e identificação.
- Endereços: Tipo de residência, tarifa de energia, gasto mensal, economia associada e relação com os usuários.
- Energia Solar: Área das placas solares, irradiação solar, e energia estimada gerada.

Metodologia Utilizada

Exploração dos Dados

- Os dados foram carregados de arquivos JSON utilizando Python e as bibliotecas pandas e json.
- As tabelas foram unidas para criar um dataset único, associando informações de energia solar e características dos endereços.
- Realizou-se uma análise exploratória inicial para entender as distribuições das variáveis, identificar outliers e verificar a presença de valores ausentes.

Exemplo de Dados Finais (Após o Merge):

Tipo Residencial	Tarifa	Economia	Área da Placa	Irradiação Solar	Energia Estimada Gerada	Gasto Mensal
Residencial	0.15	200	4	5	500	500
Comercial	0.18	300	10	6	1100	1000

Pré-processamento

- Transformação de Variáveis Categóricas:
  - A variável tipo residencial foi transformada em valores numéricos usando LabelEncoder.
- Tratamento de Valores Ausentes:
  - Linhas com valores ausentes foram removidas para evitar inconsistências.
- Normalização dos Dados:
  - Foi realizada a normalização das variáveis para evitar diferenças de escala que pudessem influenciar negativamente o modelo.

Criação do Modelo

- Algoritmo Escolhido:
  - Utilizou-se o Random Forest Regressor, um modelo baseado em árvores de decisão que lida bem com variáveis categóricas e dados tabulares.
- Divisão dos Dados:
  - O dataset foi dividido em 80% para treinamento e 20% para teste.
- Métrica de Avaliação:
  - As métricas escolhidas foram:

        - Erro Absoluto Médio (MAE): Mede o erro médio em unidades absolutas.
        - Raiz do Erro Quadrático Médio (RMSE): Penaliza erros maiores e mede a variabilidade dos erros.

Resultados Obtidos

- Após treinar o modelo com os dados disponíveis, os seguintes resultados foram obtidos:
    - MAE: 47.0
        - O erro absoluto médio foi de 47 unidades monetárias, indicando que o modelo, em média, prevê com um desvio de 47.
    - RMSE: 62.7
        - O modelo apresentou variabilidade nos erros, com algumas previsões mais distantes dos valores reais.

Análise dos Resultados:

- Comparado à faixa de valores de gasto mensal (400 a 1500), o erro é moderado.
- O RMSE maior que o MAE sugere a presença de outliers ou previsões menos precisas para determinados endereços.

Conclusões

- Desempenho do Modelo:
    - O modelo foi capaz de prever o gasto mensal com energia de maneira razoável, com um erro absoluto médio de 47.
- Limitações:
    - Outliers podem ter influenciado negativamente o desempenho do modelo.
    - Dados adicionais, como o histórico de consumo ou padrões sazonais, poderiam melhorar a precisão.

Relatório Técnico: Classificação de Alto ou Baixo Gasto Mensal

Descrição do Problema

O objetivo deste modelo é classificar se o gasto mensal de um endereço será alto ou baixo com base em características como o tipo de residência, tarifa de energia, economia e informações sobre energia solar. Essa classificação é útil para identificar padrões de consumo e sugerir estratégias personalizadas de economia para consumidores com alto consumo.

Os dados foram extraídos dos mesmos arquivos JSON e incluem:

1. Usuários: Nome e identificação.
2. Endereços: Tipo residencial, tarifa, economia associada e relação com os usuários.
3. Energia Solar: Área das placas solares, irradiação solar e energia estimada gerada.

Metodologia Utilizada

Exploração dos Dados

- Os dados foram carregados de arquivos JSON e combinados em um único dataset.
- A variável-alvo foi criada para indicar se o gasto mensal é alto ou baixo:
  - Baixo Gasto: Menor ou igual à mediana dos valores de gasto_mensal.
  - Alto Gasto: Maior que a mediana.
- Realizou-se análise exploratória para entender as distribuições das variáveis e a relação com o gasto mensal.

Exemplo de Dados Finais (Após o Merge):

Tipo Residencial	Tarifa	Economia	Área da Placa	Irradiação Solar	Energia Estimada Gerada	Gasto Mensal	Gasto Categoria
Residencial	0.15	200	4	5	500	500	Baixo
Comercial	0.18	300	10	6	1100	1000	Alto

Pré-processamento

- Criação da Variável-Alvo:
  - A mediana de gasto_mensal foi calculada, e cada entrada foi classificada como:
    - Baixo (0): Gasto menor ou igual à mediana.
    - Alto (1): Gasto maior que a mediana.
- Conversão de Variáveis Categóricas:
  - tipo_residencial foi convertida em números utilizando LabelEncoder.
- Normalização:
  - As variáveis contínuas foram normalizadas para facilitar o treinamento do modelo.

Criação do Modelo

- Algoritmo Escolhido:
  - O modelo utilizado foi uma Árvore de Decisão, por sua simplicidade e boa interpretabilidade.
- Divisão dos Dados:
  - O dataset foi dividido em 80% para treinamento e 20% para teste.
- Métricas de Avaliação:
  - Acurácia: Percentual de classificações corretas.
  - F1-Score: Média harmônica da precisão e sensibilidade.

Resultados Obtidos

- Após treinar o modelo com os dados, os resultados foram:
  - Acurácia: 85%.
    - O modelo classificou corretamente 85% dos casos de alto e baixo gasto.
  - F1-Score: 83%.
    - O modelo teve bom equilíbrio entre precisão e sensibilidade.

Análise dos Resultados:

- A acurácia de 85% indica que o modelo é eficaz para distinguir entre alto e baixo gasto.
- O F1-Score reforça que o modelo lida bem tanto com verdadeiros positivos quanto com falsos negativos.

Conclusões gerais

- Desempenho do Modelo:
  - O modelo apresentou um bom desempenho, com alta acurácia e um F1-Score equilibrado.
- Limitações:
  - Os dados podem conter outliers ou registros duplicados que afetam a consistência.
  - Variáveis adicionais, como padrões sazonais ou históricos, poderiam melhorar a classificação.

Conclusão do 1º problema:

Comparação entre GradientBoostingRegressor e RandomForestRegressor no primeiro problema.

O GradientBoostingRegressor e o RandomForestRegressor são ambos modelos baseados em árvores de decisão, mas eles diferem significativamente na abordagem de treinamento e no desempenho em problemas de regressão.

Resultados

- GradientBoostingRegressor:
  - MAE: 10.81
  - RMSE: 15.28
  - O GradientBoostingRegressor apresenta um MAE e RMSE significativamente menores, indicando que ele realiza previsões muito mais precisas e com menos erro em comparação com o Random Forest.
  - Este modelo teve uma excelente performance, sugerindo que ele é mais eficaz para capturar as complexidades dos dados, ajustando-se melhor aos padrões existentes.
  -
- RandomForestRegressor:
  - MAE: 47.0
  - RMSE: 62.70
  - O RandomForestRegressor, embora robusto e eficiente em muitos cenários, apresentou um MAE e RMSE muito maiores, indicando que o modelo tem maior variabilidade nos erros de previsão e está cometendo erros de maior magnitude.
  - O Random Forest pode ter uma performance ligeiramente inferior neste caso, possivelmente devido ao fato de não ajustar os erros de maneira sequencial, como faz o Gradient Boosting.

Definição dos modelos:

- RandomForestRegressor:
  - Abordagem: O Random Forest constrói várias árvores de decisão de maneira paralela. Cada árvore é treinada de forma independente, utilizando amostras aleatórias dos dados e das variáveis. A previsão final é obtida pela média das previsões de todas as árvores.
  - Vantagens:
    - Tende a ser menos suscetível ao overfitting (sobreajuste) em comparação com uma única árvore de decisão.
    - Pode lidar com grandes volumes de dados e alta dimensionalidade de forma eficiente.
  - Desvantagens:
    - O modelo tende a ser mais lento na fase de previsão devido ao grande número de árvores.
    - Pode ser menos preciso do que o Gradient Boosting, especialmente em conjuntos de dados mais complexos.

- GradientBoostingRegressor:

  - Abordagem: O Gradient Boosting constrói as árvores sequencialmente. Cada árvore tenta corrigir os erros (resíduos) da árvore anterior, ajustando as previsões. Isso significa que o modelo é altamente dependente das árvores anteriores, o que permite um ajuste mais preciso para dados complexos.
  - Vantagens:
    - Geralmente, o Gradient Boosting gera modelos mais precisos do que o Random Forest, especialmente em dados com complexidade não linear.
    - Melhor desempenho em datasets pequenos e médios, já que o modelo ajusta-se de maneira sequencial para corrigir erros.
  - Desvantagens:
    - O Gradient Boosting pode ser mais suscetível ao overfitting se não for adequadamente ajustado (e.g., número de árvores, taxa de aprendizado).
    - O tempo de treinamento pode ser significativamente maior devido à construção sequencial das árvores.

Conclusão Comparativa:

- O GradientBoostingRegressor demonstrou ser mais eficaz neste caso específico, com previsões significativamente mais precisas (menor MAE e RMSE). Isso é esperado, pois o Gradient Boosting ajusta-se sequencialmente para corrigir os erros cometidos pelas árvores anteriores, o que permite um ajuste mais preciso aos dados.
- O RandomForestRegressor, embora também seja eficaz e robusto, não conseguiu alcançar o mesmo nível de precisão, provavelmente devido à sua abordagem de treinamento paralelo das árvores e à ausência de correção

Conclusão 2º problema:

Comentários sobre o DecisionTreeClassifier

O DecisionTreeClassifier é um modelo de classificação que utiliza árvores de decisão para prever categorias. A principal vantagem do DecisionTreeClassifier é a sua interpretação simples e visualização clara, o que o torna uma excelente ferramenta para entender as decisões do modelo, principalmente quando se trabalha com dados tabulares.

- Vantagens:
    - Interpretabilidade: O modelo é altamente interpretável, pois a árvore de decisão pode ser visualizada e explicada facilmente. Cada nó da árvore representa uma decisão baseada em uma característica do dado, tornando o processo de classificação transparente.
    - Não exige pré-processamento extensivo: Árvores de decisão podem lidar com variáveis tanto numéricas quanto categóricas diretamente e não requerem normalização ou padronização dos dados.
    - Lida bem com interações não lineares: Ao dividir os dados com base em características específicas, o modelo é capaz de capturar interações não lineares entre as variáveis.
- Desvantagens:
    - Sobreajuste (overfitting): Árvores de decisão tendem a se ajustar excessivamente aos dados de treinamento, especialmente se não houver um limite de profundidade definido. Isso ocorre porque as árvores podem criar divisões muito específicas para os dados, o que leva a uma perda de generalização.
    - Instabilidade: Pequenas variações nos dados podem resultar em uma árvore muito diferente. Isso pode ser mitigado com métodos como Random Forest ou Gradient Boosting, que constroem árvores de forma mais robusta.
    - Limitações em problemas complexos: Árvores de decisão simples podem não ser tão eficazes em problemas de classificação complexos, onde há muitas interações entre as variáveis.

Conclusão sobre o DecisionTreeClassifier:

- O DecisionTreeClassifier é ideal para problemas simples de classificação onde a interpretabilidade é crucial. Ele é fácil de usar e entender, mas pode precisar de técnicas de poda (limitação de profundidade ou número de folhas)

para evitar sobreajuste. Para problemas mais complexos ou quando a precisão for a principal prioridade, modelos como Random Forest ou Gradient Boosting podem ser mais eficazes.

Conclusão 2: Comentários sobre o DecisionTreeClassifier

O DecisionTreeClassifier obteve 100% de acurácia e 100% de F1-Score, o que pode inicialmente parecer uma performance excelente. No entanto, esse resultado requer uma análise cuidadosa.

1. Acurácia: 100%:
    - O modelo classificou 100% das amostras corretamente, o que sugere um desempenho perfeito. Isso pode ser um bom sinal de que a árvore de decisão conseguiu capturar bem os padrões nos dados de treinamento.
    - Porém, uma acurácia de 100% pode ser uma indicativa de overfitting (sobreajuste), onde o modelo se ajusta excessivamente aos dados de treinamento e perde a capacidade de generalizar para novos dados. Isso pode acontecer se a árvore de decisão for muito profunda e tiver memorizado o comportamento dos dados em vez de aprender padrões generalizáveis.
2. F1-Score: 100%:
    - O F1-Score de 100% significa que o modelo obteve uma precisão e recall perfeitos. No entanto, assim como na acurácia, isso pode ser um reflexo de que o modelo foi ajustado demais para os dados de treinamento.
    - Em problemas de classificação com desequilíbrio entre classes ou dados com pouca variabilidade, o modelo pode ter aprendido a classe predominante ou as características triviais dos dados, resultando em uma performance artificialmente elevada.

Conclusão sobre o DecisionTreeClassifier:

- Embora o DecisionTreeClassifier tenha mostrado um desempenho perfeito em termos de acurácia e F1-Score, isso pode ser um indicativo de overfitting, especialmente se o modelo tiver se ajustado excessivamente aos dados de treinamento.
- Para garantir a robustez do modelo, é recomendável ajustar a profundidade da árvore e utilizar técnicas como poda para limitar o sobreajuste. Modelos mais complexos como Random Forest ou Gradient Boosting podem ser mais eficazes para evitar esse problema e garantir a generalização em dados não vistos.

Link do vídeo

https://www.youtube.com/watch?v=r22aB7iXRIo