

UNIVERSIDADE POSITIVO - GRUPO CRUZEIRO DO SUL

Bacharelado Ciência da Computação

Bianca de Oliveira dos Santos

Carolina Dobjanski

Laura Ramos

Laura Santos Oliveira

Yasmin Faraj

**DOCUMENTAÇÃO DA APLICAÇÃO WEB COM ANÁLISE DE DADOS E
MACHINE LEARNING**

Curitiba

Novembro, 2024

SUMÁRIO

DESCRIÇÃO DAS FUNÇÕES DA MAIN.....	2
LIMPAR_COLUNAS_NULAS.....	2
OTIMIZAR_PARAMETROS.....	2
REGISTER.....	3
LOGIN.....	3
LOGOUT.....	3
HOME.....	3
CONFIGURA_MODELO.....	4
GRÁFICOS.....	5
DESCRIÇÃO DO MODELS.....	6
DESCRIÇÃO DAS CLASSES DO FORMS.....	6
CLASSE REGISTERFORM.....	6
CLASSE LOGINFORM.....	6
DESCRIÇÃO DO CRIANDO_DB.....	6
DESCRIÇÃO DO CONFIG.....	6

DESCRIÇÃO DAS FUNÇÕES DA MAIN

LIMPAR_COLUNAS_NULAS

Esta função recebe um conjunto de dados (data) e o nome da coluna de preço (preco_col). Sua principal função é realizar a limpeza dos dados para preparar o conjunto para a modelagem. Primeiramente, ela elimina colunas com valores nulos, exceto a coluna de preço e as colunas que não foram selecionadas. Depois, qualquer coluna categórica é transformada em valores numéricos usando o LabelEncoder, garantindo que as variáveis possam ser interpretadas por modelos que só aceitam entradas numéricas. Em seguida, a função remove todas as linhas onde a variável dependente (coluna de preço) possui valores nulos. Por fim, a função retorna duas partes: X (variáveis independentes) e y (variável dependente, ou seja, o preço).

OTIMIZAR_PARAMETROS

Esta função busca otimizar os hiperparâmetros de um modelo de aprendizado de máquina utilizando o GridSearchCV. O usuário fornece o modelo inicial (modelo), os dados de treinamento (X_train e y_train) e um dicionário de parâmetros possíveis para ajuste. A lógica da função é realizar uma validação cruzada para testar diferentes combinações dos parâmetros fornecidos e encontrar a configuração que proporciona o melhor desempenho. No final, a função imprime os melhores parâmetros encontrados e retorna o modelo ajustado com esses parâmetros.

Vale ressaltar que, o GridSearchCV só foi utilizado inicialmente para encontrar possíveis melhores valores dos hiperparâmetros, e quando esses foram encontrados a chamada da função ficou comentada para não afetar a dinamicidade do projeto.

REGISTER

É usada para o registro de novos usuários no sistema. Ao acessar a página, um formulário de registro (RegisterForm) é exibido. Quando os dados são enviados, o sistema verifica se o formulário é válido. Se for, a senha fornecida pelo usuário é criptografada usando `generate_password_hash`. Em seguida, o sistema verifica se já existe um usuário com o mesmo nome no banco de dados. Se o nome já estiver em uso, uma mensagem de erro é exibida usando o flash. Caso contrário, um novo objeto de usuário é criado, salvo no banco de dados e uma mensagem de sucesso é exibida. Por fim, o sistema redireciona o usuário para a página de login.

LOGIN

Similar à rota de registro, ela utiliza um formulário (LoginForm) para capturar as credenciais do usuário. Após validar os dados, o sistema busca o usuário no banco de dados. Se o usuário for encontrado, a senha inserida é comparada com a senha armazenada no banco, utilizando `check_password_hash` para garantir a segurança. Caso a autenticação seja bem-sucedida, o usuário é logado e redirecionado para a página inicial. Em caso de erro (nome de usuário ou senha incorretos), mensagens informativas são exibidas.

LOGOUT

Usada para deslogar o usuário atual e redirecioná-lo para a página de login.

HOME

É a entrada principal para o sistema após o login. Ela permite que o usuário envie um arquivo CSV com dados que serão processados

posteriormente. Caso o arquivo seja válido, ele é salvo temporariamente no servidor para ser usado nas etapas seguintes. Essa abordagem utiliza o módulo tempfile, que cria um arquivo temporário seguro para armazenar o CSV enviado. Após o envio bem-sucedido, o sistema redireciona o usuário para a página de seleção de colunas.

SELECAO_COLUNAS

Esta rota permite ao usuário selecionar quais colunas do arquivo CSV enviado serão utilizadas no modelo. O arquivo CSV é carregado da sessão e suas colunas são listadas. O usuário seleciona as colunas relevantes para a análise, e o sistema valida essas escolhas. A lógica inclui verificações para garantir que pelo menos quatro colunas (além do preço) sejam selecionadas, e que a coluna de preço seja obrigatoriamente incluída. As colunas selecionadas são salvas em um novo arquivo temporário e na sessão. Dependendo da escolha do usuário, o sistema redireciona para a configuração do modelo ou para a página de gráficos.

CONFIGURA_MODELO

É onde o modelo de aprendizado de máquina é configurado e treinado. Primeiramente, os dados processados e filtrados são carregados e limpos pela função `limpar_colunas_nulas`. Em seguida, o usuário escolhe o tipo de modelo (como `RandomForest`, `KNN`, `DecisionTree` ou `XGBoost`) e ajusta os hiperparâmetros específicos de cada modelo. Os dados são então divididos em conjuntos de treino e teste, e o modelo é treinado com os dados de treino. Após o treinamento, o sistema salva o modelo treinado em um arquivo temporário para uso futuro e calcula métricas de desempenho, como o RMSE (erro médio quadrático) e o R^2 , exibindo os resultados para o usuário.

Ao selecionar um modelo, o usuário poderá mudar os valores dos hiperparâmetros, mas os melhores valores que a equipe encontrou com a

função `otimizar_parametros` foram selecionados como padrão a fim de facilitar a usabilidade da aplicação.

Por fim, foi durante a configuração da função `configura_modelo` que foi vista a necessidade de criar modelos que usam regressão. A decisão foi tomada com base em uma pesquisa, onde descobrimos que modelos de classificação, que poderiam dizer “sim”, “caro” e “barato”, não se encaixam no objetivo do projeto, que é prever um valor específico para um determinado imóvel. A regressão lida bem com dados complexos, não-lineares e lineares, podendo relacionar as variáveis.

RESULTADO_PREVISÃO

Após o treinamento do modelo, esta rota permite que o usuário realize previsões com base em novos dados inseridos manualmente. Os dados fornecidos são processados para garantir que estejam no mesmo formato do modelo treinado. Isso inclui a aplicação de codificação (como one-hot encoding) para variáveis categóricas. O sistema realiza a previsão e exibe o valor estimado do imóvel (ou outro objetivo definido) para o usuário. Caso ocorra algum erro durante o processo, uma mensagem é exibida.

GRÁFICOS

A funcionalidade desta rota é gerar visualizações gráficas para os dados carregados. As visualizações incluem um gráfico de pizza para distribuição por faixa de preço e outro gráfico de barra interativo da faixa de preço em relação aos metros do imóvel. Esses gráficos são gerados usando bibliotecas como `matplotlib` e `plotly`. Além disso, um mapa pode ser gerado para exibir a distribuição geográfica dos dados, desde que as colunas de latitude e longitude tenham sido selecionadas. Mensagens informativas são exibidas se as colunas necessárias para os gráficos não forem encontradas.

DESCRIÇÃO DO MODELS

Essa parte define um modelo de banco de dados para usuários, configura a integração com Flask-Login para gerenciar o login e logout de usuários e também permite que o Flask-Login busque o usuário correto com base no ID da sessão.

DESCRIÇÃO DAS CLASSES DO FORMS

CLASSE REGISTERFORM

A classe RegisterForm herda de FlaskForm, que fornece funcionalidades para criar e validar formulários.

CLASSE LOGINFORM

A classe LoginForm é utilizada para criar o formulário de login e verifica se os campos estão preenchidos.

DESCRIÇÃO DO CRIANDO_DB

Esse código é essencial para configurar o banco de dados e preparar a aplicação para armazenar dados. Ele só precisa ser executado ao inicializar o projeto ou quando novos modelos forem adicionados.

DESCRIÇÃO DO CONFIG

Esse código cria uma instância da aplicação Flask, que é o núcleo da aplicação web e configura ela para usar Flask-Login (para autenticação de usuários) e SQLAlchemy (para interagir com o banco de dados).