UNIVERSIDADE FEDERAL DO CEARÁ

TECHNICAL REPORT

Aluno: Madson Luan Dias Nascimento
Matrícula: 558611

1. Introdução

Parte A - Classificação - Dataset (Diabetes)

O dataset selecionado para classificação trata da relação de pessoas que possuem diabetes. Gerados originalmente pelo Instituto Nacional de Diabetes e Doenças Digestivas e Renais. O objetivo principal desses dados é, como já mencionado, fazer uma previsão, com base na utilização de medidas diagnósticas incluídas nos dados, se um paciente possui ou não diabetes.

Esse dataset é relativamente grande, possuindo 9 preditores e 768 instâncias. Suas principais características envolvem os seguintes dados de análise:

- *Pregnances (Gravidez)*
- *Glucose (Glucose)*
- *BloodPressure (Pressão Arterial)*
- *SkinThickness (Espessura da pele)*
- *Insulin (Insulina)*
- *BMI (índice de massa corporal)*
- *DiabetesPedigreeFunction (Função de histórico familiar de diabetes)*
- *Age (Idade)*
- *Outcome (Resultado, 0 = não possui diabete, 1 = possui diabete).*

Enfatizando seu objetivo principal que é fazer uma previsão da variável alvo (Outcome) que corresponde ao número de ocorrência de diabetes com base nos atributos mencionados anteriormente, utilizando técnicas de normalização e padronização de dados, aplicando o algoritmo K-Nearest Neighbors (KNN).

Disponível em: https://www.kaggle.com/datasets/whenamancodes/predict-diabities

Parte B - Regressão - Dataset (Concrete)

Para análise e implementação de regressão, foi utilizado o dataset relacionado a concretos. Esse conjunto de dados possui 1.030 instâncias e 8 variáveis preditoras, todas relacionadas à composição de misturas de concreto.

Seus principais atributos de análise são as seguintes:

- *Cement (Cimento)*
- *Slag (Escoriação)*
- *Flyash (Cinzas volantes)*
- *Water (Água)*
- *Superplasticizer (Superplastificante)*
- *Coarseaggregate (Agregado graúdo)*
- *Fineaggregate (Agregado miúdo)*
- *Age (Idade)*
- *CsMPa (Resistência à compressão)*

O objetivo principal deste dataset é prever a resistência de concretos a partir da composição de materiais utilizados em seu período de fabricação. Esse tipo de previsão é importante para a engenharia, pois permite estimar a qualidade e segurança das estruturas .

Disponivel em: https://www.kaggle.com/datasets/maajdl/yeh-concret-data

2. Observações

Houve problemas na implementação do projeto. Especialmente na implementação manual do KNN (implementação e replicação) e testar a avaliação dos três modelos e análise manual para descobrir os melhores valores de k. Incluindo também a dificuldade para comparar o RMSE e R^2 *em uma tabela.*

3. Resultados e discussão

Parte A - Classificação (diabetes)

Questão 1 - Carregue o dataset com pandas.

- *1 - Carregamento do dataset*

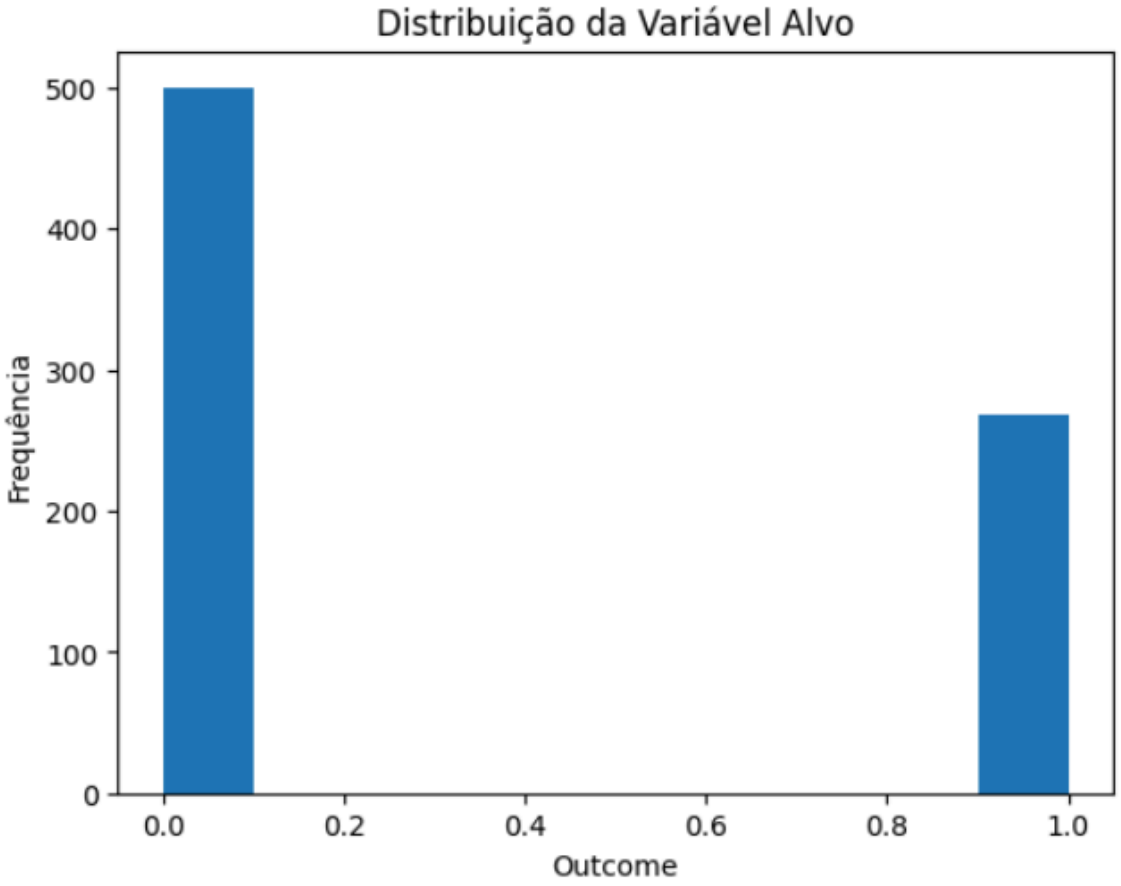
  Os dados foram carregados usando a biblioteca pandas, que permite a manipulação de tabelas em python. O arquivo foi lido e armazenado em um Dataframe.

- *2 - Tratamento de Valores Ausentes*

  A verificação de valores ausentes foi realizada. O resultado mostrou que o dataset não possui nenhum valor nulo, facilitando o processo de análise.

- *3 - Análise da Distribuição da Variável-Alvo*

  A variável-alvo é a Outcome, sua distribuição foi visualizada por meio de um histograma. O gráfico mostra a frequência de paciente com e sem diabetes (0 = Não, 1 = Sim).



- *4 - Codificação de Variáveis Categóricas*

  Etapa não necessária, pois todas as variáveis neste dataset são numéricas.

- *5 - Análise Estatística Exploratória*

  Foi utilizada a função describre() para gerar as estatísticas descritivas das variáveis. Essa análise retornou as informações como médias, desvio-padrão, mínimos e máximos entre outros.

Questão 2 - KNN Manual com Variáveis Distâncias

- *1 - Divida o dataset em treino e teste*

  Os dados foram divididos em conjuntos separados de treino e teste, utilizando a função train_test_split. Logo em seguida, foi implementado o KNN (K -Nearest Neighbors) da scikit-learn. O modelo foi treinado e avaliado, o resultado obtido da acurácia é considerado razoável.

- *Implemente manualmente o KNN*

  A implementação manual do KNN foi implementada utilizando a distância euclidiana como métrica base. Na implementação o algoritmo percorre todo o conjunto de teste, calcula suas distâncias entre os exemplos e os dados de treino, e vai retornar o rótulo mais comum entre os vizinhos mais próximos. O valor obtido foi de 0.7597 com k = 18 (melhor acurácia conhecida).

- *Avalie usando Euclidiana, Manhattan, Chebyshev, Mahalanobis.*

  Cada uma dessas funções foram testadas com o mesmo valor de k, e os resultados variaram levemente entre elas. Mostrando que essa variação de métricas escolhidas pode influenciar no desempenho dos dados.

Questão 3 - Normalização, alteração do valor de K e efeito no KNN.

- *1 - Aplique as normalizações: Logarítmica, MinMax, StandardScaler.*

  Foi aplicada as três métricas de normalização de dados. Essas transformações foram aplicadas separadamente aos conjuntos de treino e teste, justamente com o objetivo de avaliar individualmente como cada um afeta o desempenho do KNN.

- *2 - Reaplique o KNN manual com a melhor distância*

  A melhor métrica foi a euclidiana, no modelo KNN que foi reavaliado os resultados foram obtidos:

  Sem normalização: 0.6623

  Logarítimica: 0.6948

  MinMaxScaler: 0.6818

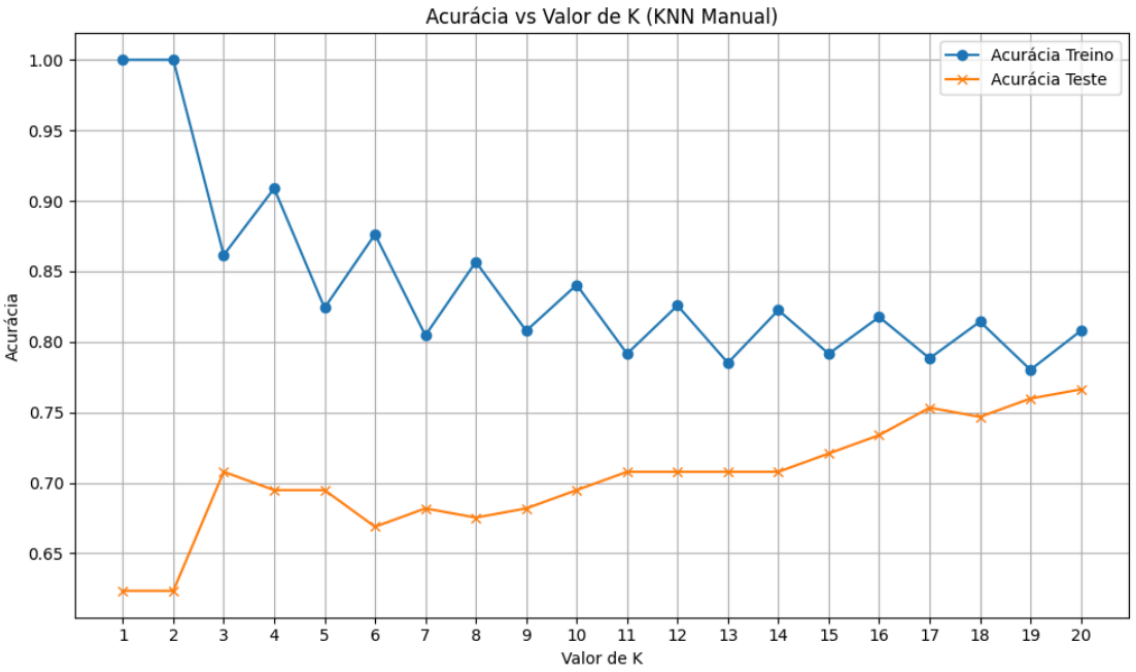
  StandardScaler: 0.6948

- *3 - Compare os desempenhos com e sem normalização*

  Na comparação dos resultados, é notório que sem normalização teve o pior desempenho. Reforçando a necessidade da aplicação de pré-processamento dos dados.

- *4- Faça também uma análise manual para descobrir o melhor valor de K para o KNN. Plote o gráfico de acurácia de treino e teste versus k.*

  Foi aplicado uma limpeza nos valores de k variando de 1 até 20. A acurácia foi registrada para o conjunto de treino e teste, logo, plotado o gráfico. O gráfico mostrou valores baixos de K. Um valor intermediário, como k=5 e/ou k=6, mostra um equilíbrio entre o erro de treino teste.



Questão 4 - KNN com sklearn + GridSearch

- *1 - Use KNeighborsClassifier da Sklearn.*

  Foi criado um pipeline com os componentes: StandardScaler, MinMaxScaler e FunctionTransformer para normalização logarítmica. E o modelo KNeighborsClassifier, com número de vizinhos k variando de 1 a 20. Utilizou-se o GridSearchCV com cv = 5 para testar as combinações possíveis entre os três tipos de normalização e o valores de k. O objetivo foi conhecer qual combinação teria a melhor acurácia, valor obtido foi cerca de 0.7606.

- *2 - Top 3 configurações*

  As normalizações encontradas na análise foram o StandardScaler quando k = 6 obtendo uma acurácia de 0.7606, k = 11 com acurácia de 0.7573 e k =16 com acurácia de 0.7509.

- *3 - Plote o gráfico dos resultados com as 3 melhores configurações.*

  O gráfico foi gerado destacando visualmente as 3 melhores, valorizando os valores intermediários de k.

Questão 5 - Cross-Validation e Avaliação Final

- *1 - Aplique cross_val_score (5 folds) com a melhor configuração possível que você determinou do seu classificador.*

  A melhor configuração encontrada foi a StandardScarler com número de vizinho k = 7. Essa configuração foi aplicada usando a cross_val_score com os 5 folds, o qual garante uma avaliação valorizada.

  Resultados obtidos: [0.7337 0.7077 0.7857 0.7581 0.7254]

- *Exiba média, desvio padrão, matriz de confusão e classification_report.*

  Média obtida: 0.7460

  Desvio padrão: 0.0175

  Logo, esses valores indicam uma boa estabilidade do modelo.

  Matriz de confusão: [[399, 53

  105, 115]]

Classification Report: Precision: 0.68, Recall: 0.52, f1-score: 0.59.

Parte B - Regressão

Questão 1 - Pré-processamento e Correlação

- *1 - Carregue o dataset*

  O dataset foi carregado com pandas, contendo suas variáveis relacionadas à composição de concretos.

- *2 - Trate os valores ausentes*

  Foi realizada a verificação para detectar se existem valores ausentes, e, foi confirmado que não havia valores ausentes nesse conjunto de dados.

- *3 - Verifique a correlação com a variável alvo.*

  A correlação com a variável alvo que é a csMPa (Resistência à compressão) destacou que a variável preditora Superplasticizer possui 0.45 de correlação e Cement com 0.50, ambos possuem correlação forte com a resistência do concreto.

- *4 - Normalize ou padronize os dados.*

  StandarScarler foi utilizado para a padronização do datase, para garantir que os as variáveis estejam na mesma escala.

Questão 2 - Regressão Linear Simples

- *1 - Aplique Regressão Linear (Sklearn)*

  Aplicou-se o modelo de regressão linear utilizando todas as variáveis preditoras que foram padronizadas exceto a variável-alvo. Retornando o coeficiente (Influência de cada variável na resistência do concreto), intercepto, treinamento ($R^2 = 0.6155$).

- *2 - Plote a reta de regressão (feature mais correlacionada)*

  Foi feita a seleção da variável age usando a regressão linear. A plotagem da reta de regressão foi realizada sobre esses dados. Visualmente, a relação desses dados apresentou uma dispersão bem alta e com pouca linearidade.

- *3 - Calcule RMSE e* R^2

R^2 = 0.1081, ou seja, apenas 10,81% da variância resistência é explicada pela idade do concreto.

RMSE = 0.9443 , logo, o modelo comete um erro de quase 1 desvio padrão, indicando baixa precisão.

O modelo com todas as variáveis teve um bom desempenho, demonstrando que várias variáveis com junção podem explicar a qualidade do concreto.

Questão 3 - Linear vs Ridge vs Lasso

- *1 - Aplique as regressões: Linear, Ridge e Lasso.*

  Os três modelos de regressão linear foram treinados com os dados padronizados. O modelo Linear e Ridge apresentaram desempenho semelhante de aproximadamente 0.108. Com essa informação deduzimos que apenas 10.8% da resistência é explicada pela idade.

- *2 - Use cross_val_score com 5 folds.*

  Os dados foram divididos em 5 partes, guardados na cv = 5. Onde foram usadas 4 partes para treino e 1 para teste. Esse processo é realizado 5 vezes, trocando a parte de teste a cada vez. Logo, é calculada a métrica escolhida em cada círculo de cálculo. Foram utilizados os modelos Linear, Ridge e Lasso. Após execução será retornado o resultado de scores_linear.

- *3 - Compare RMSE e R^2 em uma tabela.*

  A realização da comparação do desempenho dos três modelos de regressão já mencionados permite calcular as duas métricas para cada modelo (RMSE e R^2). O K-Fold com 5 divisões foi criado para garantir a aleatoriedade dos dados. Para cada um dos modelos, treina os dados, e testa a outra parte e posteriormente ele calcula as duas métricas.

  Após a execução e visualização dos resultados, nenhum dos modelos testados teve um desempenho bom.

Questão 4 - Coeficientes e Seleção de Atributos

- *1 - Visualize os coeficientes da Lasso.*

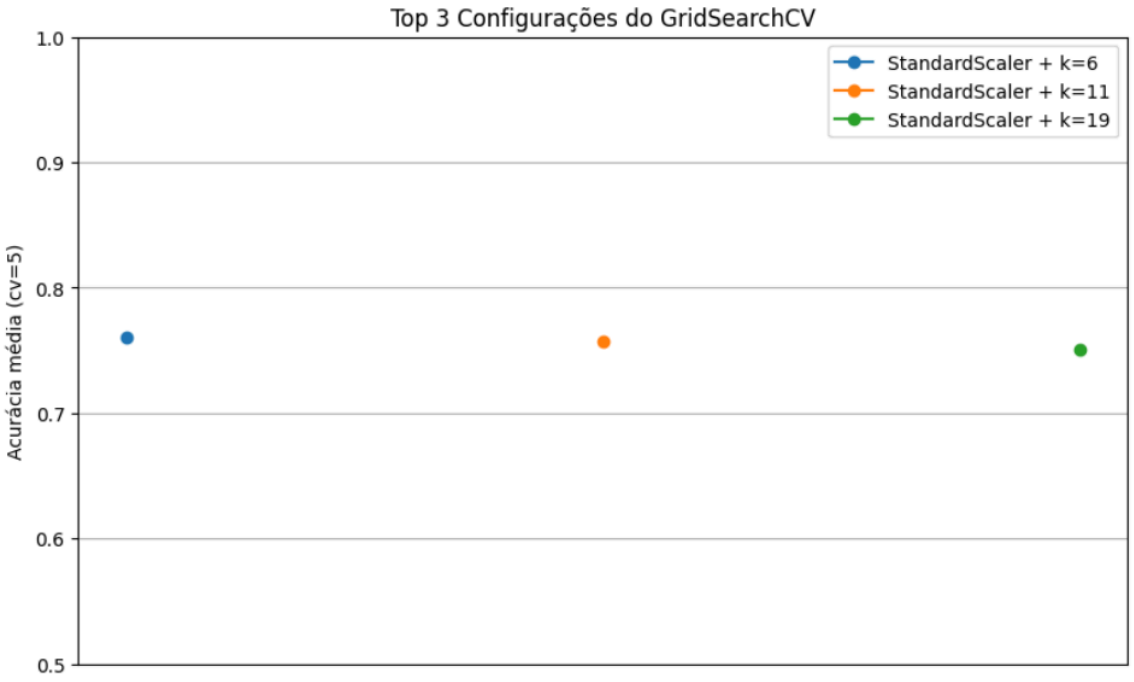
  As variáveis foram separadas em tipos independentes (x) e dependentes (y). Em seguida, o modelo foi treinado em Lasso com o alpha = 1.0. Logo, imprimimos os seus coeficientes, que representa o quanto cada variável pode influenciar na previsão da resistência do concreto.

- *2 - Identifique os atributos mais relevantes.*

  A organização em ordem de importância dos coeficientes de Lasso foi realizada, mostrando quais as variáveis que mais possuem influência na previsão. A coleta dos coeficientes e nome das colunas foram coletadas. Em seguida um dataframe foi criado com o nome da variável e coeficiente que o corresponde. E por fim, ordena do mais importante para o menos importante e exibe apenas as colunas mais relevantes.

- *3 - Plote um gráfico da importância*

  O gráfico foi plotado usando o matplotlib, de modo retornar a frequência dos coeficientes mais importantes.



4. Conclusões

Considerando a parte de classificação, utilizando KNN para fazer a previsão do dataset de diabetes testados com diferentes normatizações (StandardScaler, MinMax, Logaritimica) e testando a variação de vizinhos de 1 a 20. A sua melhor combinação foi alcançada apenas com o StandardScaler com k = 5, k = 6 ou k = 7, fazendo sua acurácia se aproximar de 76% , que indica que esse desempenho foi satisfatório para o problema. Na validação com 5 partes, confirmou a estabilidade do modelo, e sua matriz de confusão mostrou certo equilíbrio entre erros e acertos para as classes (0 e 1).

Em regressão, foram aplicados diferentes modelos de regressão (Linear, Ridge, Lasso) para fazer a previsão da resistência do concreto. Ainda assim, mesmo normalizado, os modelos demonstraram desempenho insatisfatório, com valores de R^2 *bem baixos e RMSE aproximados de 1. A análise com Lasso indicou que os indicou os atributos que possuem mais relevância mas a capacidade preditiva foi baixa.*

5. Próximos passos

Fazer uma análise e verificar o quão bom o modelo separa as classes de pessoas que possuem diabetes e as que não, usando a curva ROC.

Para regressão como os modelos apresentaram baixo desempenho, é possível em projetos futuros implementar Árvore de Regressão ou até mesmo Redes Neurais.