UNIVERSIDADE FEDERAL DO CEARÁ

TECHNICAL REPORT

Aluno: ERYKA CARVALHO DA SILVA e LUIS SAVIO GOMES ROSA

1. Introdução

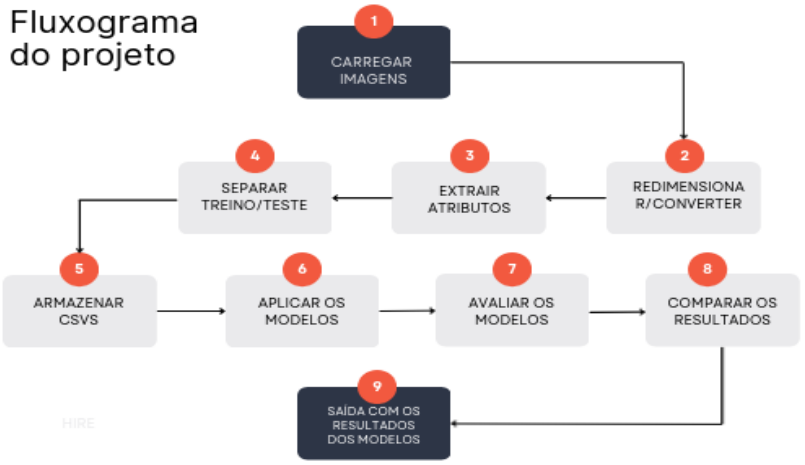
O objetivo deste projeto foi desenvolver um modelo de machine learning para classificar imagens em quatro categorias: "water," "green_area," "desert," e "cloudy." O dataset utilizado contém um número desigual de imagens por classe, com 1500 imagens para cada uma das classes "water," "green_area," e "cloudy," e 1131 imagens para a classe "desert." Este desbalanceamento pode impactar negativamente o desempenho do modelo de classificação. O foco principal foi abordar esse desbalanceamento e avaliar o impacto da amostragem em uma quantidade igual de 500 imagens por classe.

Descrição do Dataset

- ***Número de Classes**: 4 (water, green_area, desert, cloudy)*
- ***Número de Imagens por Classe**: Aproximadamente 1500, mas o código carrega uma amostra de 500 imagens de cada classe para a construção do modelo.*
- ***Dimensões das Imagens**: Redimensionadas para 128x128 pixels.*
- ***Formato das Imagens**: Todas as imagens são convertidas para o espaço de cores RGB.*

Foram utilizados os seguintes algoritmos para a etapa de classificação:

- *KNN (K-Nearest Neighbors)*
- *Random Forest*
- *SVM (Support Vector Machine)*
- *AdaBoostClassifier*



2. Observações

- *Desbalanceamento das Classes:*

As classes "water," "green_area," e "cloudy" possuem um número maior de imagens comparado à classe "desert," que tem 1131 imagens. Isso pode causar um viés no modelo, favorecendo as classes mais representadas.

- *Amostragem:*

Para lidar com o desbalanceamento, foi realizada uma amostragem de 500 imagens de cada classe. Essa abordagem buscou criar um conjunto de dados mais equilibrado para treinamento e avaliação.

- *Erros de Carregamento:*

Durante o carregamento das imagens, alguns erros foram encontrados e tratados. Essas falhas não impactaram significativamente o resultado final, mas foram importantes para garantir a integridade dos dados processados.

3. Resultados e Discussões

- *Amostragem e Pré- processamento*

A amostragem uniformizada garantiu que cada classe fosse igualmente representada no conjunto de dados de treino e teste, mitigando o impacto do desbalanceamento.

O pré-processamento incluiu o redimensionamento das imagens para 128x128 pixels e a conversão para arrays NumPy, seguido da extração de histogramas de cores como características.

- *Treinamento do Modelo:*

Um modelo SVM com kernel linear foi treinado utilizando os dados amostrados. O parâmetro class_weight='balanced' foi configurado para ajudar a lidar com o desbalanceamento das classes.

Acurácia: O modelo obteve uma acurácia de 83.75% no conjunto de teste, indicando um desempenho geral razoável.

- *Análise das Métricas de Desempenho:*

Precisão e Recall: As métricas variaram entre as classes. A classe "desert" apresentou valores mais baixos de precisão e recall em comparação com as outras classes, refletindo o desafio de aprender a partir de um número menor de exemplos.

Classificação Detalhada: O relatório de classificação revelou que, apesar da acurácia geral, o modelo teve dificuldades específicas com a classe "desert," o que pode ser atribuído ao menor número de amostras dessa classe.

- *Impacto do Desbalanceamento:*

O desbalanceamento de imagens impactou o desempenho do modelo, especialmente para a classe com menos imagens. A amostragem ajudou a equilibrar a representação das classes, mas ainda assim houve variação nas métricas de desempenho.

4. Resultados e discussão

- **Random Forest**
  - **acurácia:** *93.75%*

Class	Precision	Recall	F1-Score	Support
cloudy	0.95	0.97	0.96	100
desert	0.99	0.96	0.97	100
green_area	0.88	0.96	0.92	100
water	0.93	0.86	0.90	100
Accuracy			0.94	400
Macro Avg	0.94	0.94	0.94	400
Weighted Avg	0.94	0.94	0.94	400

Table 1: Random Forest Classification Report

- **KNN**
  - **acurácia:** *87.25%*

Class	Precision	Recall	F1-Score	Support
cloudy	0.92	0.90	0.91	100
desert	0.94	0.95	0.95	100
green_area	0.77	0.96	0.85	100
water	0.89	0.68	0.77	100
Accuracy			0.87	400
Macro Avg	0.88	0.87	0.87	400
Weighted Avg	0.88	0.87	0.87	400

Table 2: KNN Classification Report

- **SVM**
  - **acurácia:** *80.25%*

Class	Precision	Recall	F1-Score	Support
cloudy	0.90	0.84	0.87	100
desert	0.89	0.93	0.91	100
green_area	0.67	0.91	0.77	100
water	0.79	0.53	0.63	100
Accuracy			0.80	400
Macro Avg	0.81	0.80	0.80	400
Weighted Avg	0.81	0.80	0.80	400

Table 3: SVC Classification Report

- **AdaBoost**
  - **acurácia:** *80.75%*

Class	Precision	Recall	F1-Score	Support
cloudy	0.83	0.70	0.76	100
desert	0.78	0.88	0.83	100
green_area	0.75	0.99	0.85	100
water	0.93	0.66	0.77	100
Accuracy			0.81	400
Macro Avg	0.82	0.81	0.80	400
Weighted Avg	0.82	0.81	0.80	400

Table 4: AdaBoost Classification Report

Tabela com as acurácias:

Modelo	*Acurácia*
Random Forest	*93.75%*
KNN	*87.25%*
SVM	*80.25%*
Adaboost	*80.75%*

5. Discussão dos resultados

Random Forest foi o modelo com o melhor desempenho geral, alcançando uma acurácia de 93,75%. Isso sugere que o modelo foi mais eficaz na captura das características relevantes dos dados. O modelo apresentou equilíbrio em todas as classes, com destaque para as classes "desert" e "cloudy", que tiveram f1-scores acima de 96%. Esse resultado é esperado, já que Random Forest tende a ser robusto contra o overfitting e oferece boa capacidade de generalização. O KNN teve um desempenho inferior ao Random Forest, com uma acurácia de 87,25%. Ele foi eficiente na classificação da classe "desert" com um f1-score de 0.95, mas apresentou uma queda significativa de desempenho na classe "water", com um recall de 0.68, o que sugere que o modelo teve dificuldade em distinguir essa classe de outras, particularmente "green_area", que apresentou um recall de 0.96.

O SVC obteve uma acurácia de 80,25%, o menor entre os quatro modelos. Isso indica que o modelo teve dificuldades em capturar a separação das classes, particularmente na classe "water", com um f1-score de 0.63 e um recall de 0.53, o que sugere confusão significativa entre "water" e outras classes. O modelo foi mais eficiente na classificação das classes "cloudy" e "desert", mas ainda apresentou um desempenho geral inferior

aos outros métodos. O AdaBoost apresentou resultados semelhantes ao SVC, com uma acurácia de 80,75%. Embora tenha se saído melhor do que o SVC, especialmente na classe "green_area", com um recall impressionante de 0.99, ele teve desempenho inferior nas classes "cloudy" e "water", apresentando f1-scores mais baixos nessas categorias. Isso sugere que o AdaBoost foi capaz de capturar bem a variabilidade de algumas classes, mas não foi capaz de generalizar bem para todas elas.

6. Conclusões Finais:

Random Forest mostrou-se :o modelo mais robusto e eficiente, sugerindo que é o mais adequado para este conjunto de dados. Sua capacidade de lidar com conjuntos de dados desbalanceados e capturar melhor as características das classes foi evidente. O KNN apresentou um bom desempenho, mas seu comportamento altamente dependente da proximidade de amostras semelhantes pode ter levado a confusões entre classes com características visuais parecidas, como "water" e "green_area". Já os modelos SVC e AdaBoost tiveram desempenhos semelhantes, ambos apresentando dificuldade para lidar com classes mais desafiadoras como "water", o que indica que modelos mais complexos podem não ser os mais adequados para esse problema específico, especialmente com features de baixa dimensionalidade como histogramas de cores.

7. Próximos passos:

- *Coleta de Mais Dados:*

Objetivo: Obter mais imagens para a classe "desert" para melhorar a representatividade dessa classe no dataset.

Benefícios: Reduzirá o impacto do desbalanceamento e poderá melhorar a precisão e recall da classe "desert."

- *Aprimoramento do Modelo:*

Experimentação com Diferentes Modelos: Testar outros algoritmos de machine learning, como Random Forest e Redes Neurais, para comparar o desempenho.

Ajuste de Hiperparâmetros: Realizar uma busca de hiperparâmetros para otimizar o desempenho do modelo Random Forest ou outros algoritmos escolhidos.

- *Análise Adicional e Validação Cruzada:*

Validação Cruzada: Implementar validação cruzada para obter uma avaliação mais robusta do modelo e garantir que o desempenho seja consistente em diferentes subconjuntos dos dados.

Análise de Erros: Realizar uma análise detalhada dos erros de classificação para entender melhor onde o modelo falha e como isso pode ser corrigido.

- *Visualização e Interpretação dos Dados:*

Gráficos e Diagramas: Criar gráficos e diagramas para visualizar a distribuição das imagens, as métricas de desempenho por classe e o fluxo de dados através do modelo. Isso ajudará na interpretação dos resultados e na comunicação dos achados.

Este relatório fornece uma visão detalhada sobre o desbalanceamento de classes e a eficácia da abordagem de amostragem adotada, além de sugerir passos para aprimorar o modelo e o processo de classificação de imagens.

8. Como rodar os códigos:

Rodar o script principal: O código principal está em ***final.py*****.** Para rodá-lo, basta executar o seguinte comando no terminal: ***python final.py***

O script final.py fará a leitura das imagens nas pastas water, green_area, desert e cloudy, realizará a extração dos atributos, salvará os dados em CSVs e executará os modelos Random Forest, KNN, SVC e AdaBoost, gerando as métricas de classificação.