Universidade do Minho

Mestrado em Engenharia Informática

Relatório Final

Dados e Aprendizagem Automática

pg52669 Afonso Nogueira Ferreira
pg52685 Joana Catarina Oliveira Gomes
pg54144 Pedro Bogas Oliveira

GRUPO 38 - MEI

10 de janeiro de 2024

Conteúdo

1 Introdução

1.1 Contextualização

A unidade curricular de Dados e Aprendizagem Automática propôs um desafio abrangente e progressivo, dividido em duas fases distintas. O objetivo principal deste relatório é fornecer uma compreensão clara e abrangente das diferentes etapas do projeto, destacando os contextos específicos e objetivos associados a cada fase.

1.2 Objetivos

O presente trabalho prático visa sensibilizar e motivar os estudantes para a conceção e implementação de projetos no domínio de Machine Learning. Os objetivos principais compreendem a exploração e seleção criteriosa de datasets provenientes de fontes confiáveis e a conceção e otimização de modelos de Machine Learning.

Numa primeira fase, é dada mais ênfase na liberdade dos estudantes para poderem escolher o seu próprio dataset, reunirem a informação que acham importante, criarem os seus próprios objetivos e tentarem alcançá-los. Numa segunda fase, uma participação ativa numa competição Kaggle específica para a previsão da produção de energia solar.

É esperado que os estudantes desenvolvam capacidades de análise crítica dos resultados obtidos, interpretando-os no contexto do problema em questão. A elaboração de um relatório conciso, documentando todo o processo metodológico, desde a escolha dos datasets até à análise dos resultados, constitui uma componente crucial deste trabalho. Além disso, a avaliação por pares será conduzida para reconhecer e atribuir deltas que reflitam o contributo individual de cada membro do grupo. Este enfoque sistemático procura proporcionar uma experiência abrangente e fundamentada no âmbito da aprendizagem prática em Machine Learning.

2 Tarefa Dataset Grupo

2.1 Seleção e Exploração do Dataset

Globalmente, uma ampla variedade de frutos é cultivada, cada qual apresentando distintos tipos. Os elementos determinantes para a classificação do tipo de fruto residem nas características da sua aparência externa, tais como cor, comprimento, diâmetro e forma. A expressão externa das frutas assume um papel preponderante na determinação do tipo de fruto. A classificação da variedade de frutas com base na observação da sua aparência externa pode exigir perícia, sendo um procedimento moroso que requer esforço considerável. O propósito deste estudo consiste na classificação dos tipos de tâmaras, nomeadamente Barhee, Deglet Nour, Sukkary, Rotab Mozafati, Ruthana, Safawi e Sagai,

mediante a aplicação de distintos métodos de Machine Learning. Para atingir este objetivo, foram obtidas 898 imagens de sete tipos diferentes de tâmaras através do sistema de visão por computador (CVS). Por intermédio de técnicas de processamento de imagem, foram extraídas um total de 34 características, incluindo características morfológicas, forma e cor, destas imagens. Inicialmente, foram desenvolvidos modelos, que figuram entre os métodos de aprendizagem automática.

Name	Data Types	Default Task	Attribute Types	# Instances	# Attributes	Year	Download
Date Fruit Datasets	7 Class	Classification Clustering	Integer, Real	898	34	2021	Download 4964 downloaded

Figura 1: Descrição do dataset abordado

O conjunto de dados em questão compreende 35 atributos distintos (o atributo 0 também é uma coluna), cada um associado a uma instância específica. A seguir, são detalhados esses atributos, acompanhados de breves descrições:

1. AREA: Representa a área da forma.
2. PERIMETER: Indica o perímetro da forma.
3. MAJOR_AXIS: Refere-se ao comprimento do eixo principal da forma.
4. MINOR_AXIS: Denota o comprimento do eixo menor da forma.
5. ECCENTRICITY: Corresponde à excentricidade da forma.
6. EQDIASQ: Representa o diâmetro equivalente ao quadrado da forma.
7. SOLIDITY: Indica a solidez da forma.
8. CONVEX_AREA: Refere-se à área convexa da forma.
9. EXTENT: Corresponde à extensão da forma.
10. ASPECT_RATIO: Indica a razão de aspecto da forma.
11. ROUNDNESS: Refere-se ao arredondamento da forma.
12. COMPACTNESS: Denota a compacidade da forma.
13. SHAPEFACTOR_1: Representa o primeiro fator de forma.
14. SHAPEFACTOR_2: Corresponde ao segundo fator de forma.
15. SHAPEFACTOR_3: Indica o terceiro fator de forma.
16. SHAPEFACTOR_4: Refere-se ao quarto fator de forma.
17. MeanRR: Representa a média do canal vermelho (Red).

18. MeanRG: Denota a média do canal verde (Green).
19. MeanRB: Indica a média do canal azul (Blue).
20. StdDevRR: Corresponde ao desvio padrão do canal vermelho.
21. StdDevRG: Refere-se ao desvio padrão do canal verde.
22. StdDevRB: Denota o desvio padrão do canal azul.
23. SkewRR: Indica a inclinação do canal vermelho.
24. SkewRG: Corresponde à inclinação do canal verde.
25. SkewRB: Refere-se à inclinação do canal azul.
26. KurtosisRR: Representa a curtose do canal vermelho.
27. KurtosisRG: Denota a curtose do canal verde.
28. KurtosisRB: Indica a curtose do canal azul.
29. EntropyRR: Corresponde à entropia do canal vermelho.
30. EntropyRG: Refere-se à entropia do canal verde.
31. EntropyRB: Indica a entropia do canal azul.
32. ALLdaub4RR: Representa um recurso associado ao canal vermelho.
33. ALLdaub4RG: Denota um recurso associado ao canal verde.
34. ALLdaub4RB: Indica um recurso associado ao canal azul.
35. Class: Refere-se à classe/categoria à qual a amostra pertence (p.e. "BERHI").

Estes atributos englobam diversas características geométricas da forma, assim como estatísticas de cor provenientes dos canais RGB da imagem.

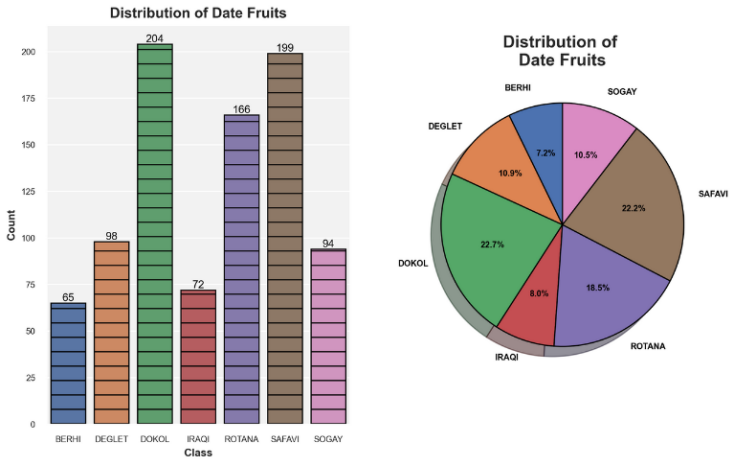

Figura 2: Análise dos dados através do Gráfico de Barras e do Gráfico de Pizza

A presente análise visa examinar a distribuição das classes na variável "Class", representada por dois tipos de gráficos distintos: um gráfico de barras e um gráfico de pizza. Destacam-se os resultados específicos para algumas classes relevantes, nomeadamente, BERHI (65 unidades, 7.2%), DEGLET (98 unidades, 10.9%), DOKOL (204 unidades, 22.7%), IRAQI (72 unidades, 8.0%), ROTANA (166 unidades, 18.5%), SAFAVI (199 unidades, 22.2%), e SOGAY (94 unidades, 10.5%). Estas informações evidenciam as discrepâncias visuais entre as representações gráficas, sendo que o gráfico de barras fornece uma perspetiva absoluta das quantidades, enquanto o gráfico de pizza destaca a distribuição percentual relativa. A adoção desses métodos gráficos complementares visa enriquecer a compreensão da distribuição das classes, oferecendo uma abordagem abrangente na interpretação dos dados.

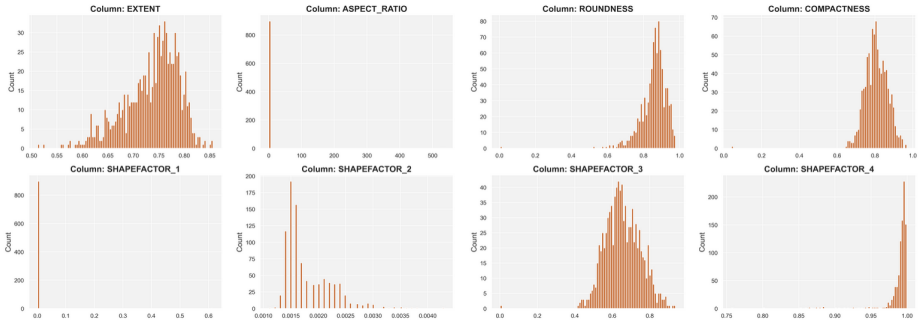

Figura 3: Análise de diversos dados através de Histogramas

Foram elaborados múltiplos gráficos similares à Figura 3 com o intuito de melhorar a visualização da distribuição, permitindo a identificação de padrões,

tendências e características relevantes. A presença de valores discrepantes (outliers) evidencia-se frequentemente em histogramas, simplificando a deteção de pontos atípicos que possam exercer um impacto significativo na análise. Através desta abordagem analítica, é possível obter uma percepção acerca da centralização dos dados, considerando medidas como a média e a mediana, assim como compreender a dispersão em torno desses valores centrais. A adaptação dos intervalos de dados não se revelou necessária, dado que estes já se encontravam adequados à análise empreendida. Sendo assim, a coluna original "Class" foi removida do DataFrame resultante. O objetivo principal desta transformação é permitir que os modelos que requerem entradas numéricas trabalhem com as classes categóricas presentes no conjunto de dados.

Logo de seguida, foram calculados os quartis Q1 (primeiro quartil) e Q3 (terceiro quartil) a partir dos dados do DataFrame, permitindo assim a determinação do intervalo interquartil (IQR). Posteriormente, utilizando a lógica do IQR, foram identificadas e removidas as linhas que continham valores considerados outliers, contribuindo para uma análise mais robusta e confiável dos dados. Consecutivamente, foi conduzida uma verificação rigorosa quanto à presença de valores nulos em todas as colunas do DataFrame resultante, evidenciando a integridade e consistência do conjunto de dados após removidos os outliers.

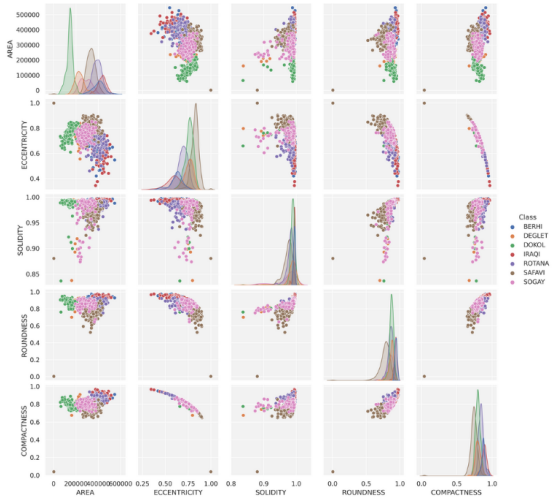

Figura 4: Análise dos dados através da função de visualização 'pairplot'

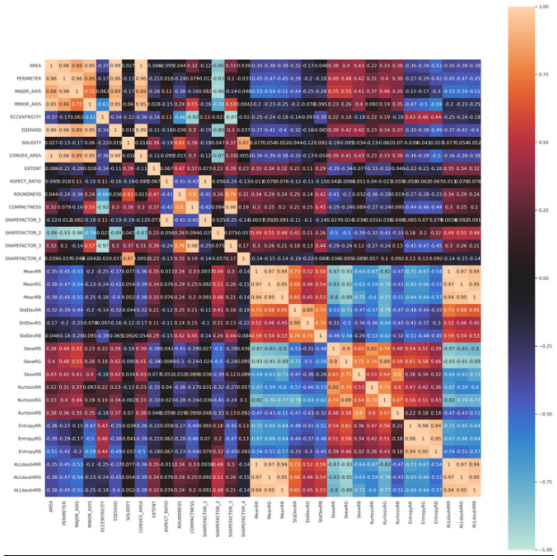

Figura 5: Análise dos dados através da Matriz de Correlação

Há um conjunto de colunas perfeitamente correlacionadas, portanto, decidiu-se excluir as seguintes colunas:
- EQDIASQ (perfeitamente correlacionada com AREA)
- CONVEX_AREA (igual a EQDIASQ)
- SHAPEFACTOR_3 (perfeitamente correlacionada com COMPACTNESS)
- ALLdaub4RR, ALLdaub4RG, ALLdaub4RB (perfeitamente correlacionadas com MeanRR, MeanRG, MeanRB, respectivamente)

Após as manipulações descritas, foi gerado e impresso um novo conjunto de dados modificado, denominado "fruitdatafinal.csv". Este conjunto de dados apresenta 588 entradas e 29 colunas, contendo informações relevantes, tais como: AREA, PERIMETER, MAJOR_AXIS, MINOR_AXIS, ECCENTRICITY, SOLIDITY, EXTENT, ASPECT_RATIO, ROUNDNESS, COMPACTNESS, SHAPEFACTOR_1, SHAPEFACTOR_2, SHAPEFACTOR_4, MeanRR, MeanRG, MeanRB, StdDevRR, StdDevRG, StdDevRB, SkewRR, SkewRG, SkewRB, KurtosisRR, KurtosisRG, KurtosisRB, EntropyRR, EntropyRB e Labeled Class.

Concluída a etapa de preparação dos dados, procedeu-se à configuração do conjunto de dados para a implementação e avaliação de um modelo. As variáveis independentes (X) foram estabelecidas, excluindo a coluna "Labeled Class". Posteriormente, efetuou-se a padronização das variáveis independentes através da aplicação da técnica Standard Scaler, com o intuito de assegurar que todas as características partilham a mesma escala. Subsequentemente, o conjunto de dados foi particionado em conjuntos distintos de treino e teste.

Adicionalmente, foi elaborado um gráfico de barras por meio da função sns.countplot, com o propósito de visualizar a distribuição das classes no conjunto de teste (y_test). Esta representação gráfica proporciona uma visualização da quantidade de instâncias pertencentes a cada classe, permitindo uma análise mais detalhada da distribuição dos dados no contexto do conjunto de teste.

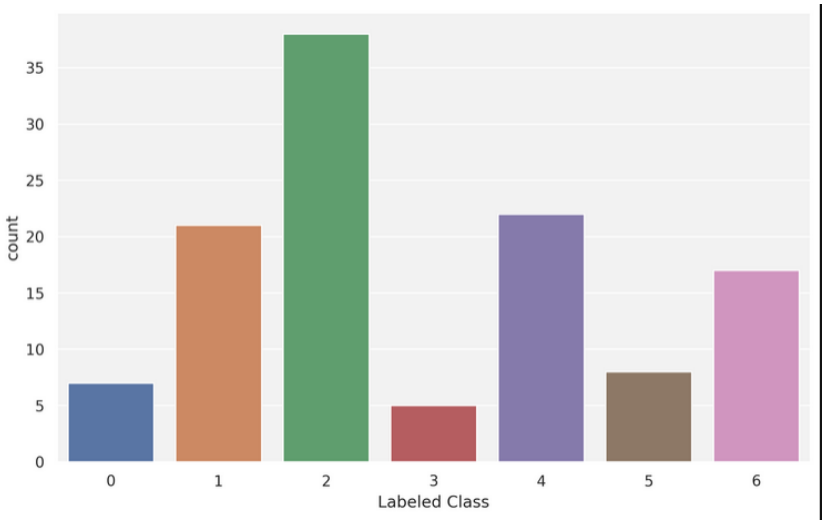

Figura 6: Análise das instâncias pertencentes a cada classe

2.2 Modelos de Machine Learning

Nesta fase foram criados diversos modelos de machine learning, de modo a conseguir encontrar qual o que melhor se aplicava ao dataset que o grupo possuía. Para cada um dos modelos seguiu-se a mesma fórmula. Os dados foram divididos em conjuntos de treino e teste. Feito isso, cada modelo foi treinado com os dados de treino e, de seguida, utilizado para realizar previsões sobre o conjunto de teste.

2.2.1 Regressão Linear

Inicialmente foi implementado um modelo de regressão linear utilizando a biblioteca scikit-learn, com o intuito de prever a variável de resposta com base nas variáveis independentes. A visualização gráfica, através do gráfico de dispersão, proporciona uma comparação entre os valores reais e os previstos, oferecendo uma perspetiva da performance do modelo. Os resultados obtidos nas métricas de avaliação, nomeadamente o Erro Absoluto Médio (MAE), o Erro Quadrático Médio (MSE) e o coeficiente de determinação (R-quadrado), são apresentados de forma sucinta. Este modelo apresenta uma precisão de 51.9%. Adicionalmente, o MAE de aproximadamente 0.999 e o MSE de 1.594 indicam que as

previsões apresentam um desvio médio absoluto e quadrático médio, respetivamente, relativamente aos valores reais. Estes valores demonstram que há uma grande margem para melhorias.

2.2.2 Regressão Logística

O modelo de Regressão Logística apresentou limitações na classificação, pois os resultados de Precision, recall e f1-score variam notavelmente entre as classes, destacando a necessidade de ajustes específicos para melhorar o desempenho em categorias específicas.

A precisão baixa de 48% sugere desafios do modelo em ajustar-se aos padrões presentes nos dados. Possíveis melhorias poderiam envolver ajustes nos parâmetros do modelo, exploração de técnicas de seleção de features e consideração de algoritmos mais complexos para otimizar o desempenho do modelo.

2.2.3 Decision Tree Classifier

O modelo Decision Tree Classifier foi implementado e avaliado, resultando na criação de uma árvore de decisão. A estrutura da árvore revela nós de decisão e características preponderantes para a classificação.

O desempenho do modelo, avaliado no conjunto de teste, exibiu uma precisão média de 80%. Observou-se variabilidade nas métricas de precisão para cada classe, indicando nuances na capacidade de discriminação entre as categorias.

Observando a matriz de confusão reparamos que a classe 4 alcançou uma precisão de 100%, destacando uma distinção clara nessa categoria. Porém, outras classes exibiram variabilidade, sugerindo a necessidade de ajustes específicos para otimizar a criação do modelo.

O F1-score médio de 0.79 reflete um equilíbrio satisfatório entre precisão e recall, embora aponte para áreas com potencial aperfeiçoamento.

Em suma, o modelo demonstra uma capacidade promissora de previsão, sugerindo potenciais benefícios mediante ajustes nos hiperparâmetros ou a exploração de abordagens suplementares para melhorias no desempenho.

2.2.4 Decision Tree com GridSearchCV

O modelo Decision Tree Classifier foi submetido a um processo de ajuste dos hiperparâmetros com GridSearchCV. Antes desse ajuste, a árvore de decisão original tinha uma profundidade máxima de 12 e 54 folhas.

O GridSearchCV foi configurado para explorar combinações de hiperparâmetros, incluindo critérios de divisão ('gini' e 'entropy') e profundidades máximas de 1, 2, 3, 4, 5, 6, 7, 8 e 10.

Este processo resultou num total de 90 ajustes. A saída do GridSearchCV mostra o tempo total para cada combinação, refletindo a eficiência do processo.

O modelo exibiu uma precisão média de 76%.

2.3 Decision Tree com Técnicas de Pruning

Este estudo abordou a otimização do desempenho do modelo de Árvore de Decisão por meio de duas estratégias de pruning: alteração da profundidade máxima (`max_depth`) e poda de complexidade de custo (`ccp_alpha`).

Alteração da Profundidade Máxima com `GridSearchCV`

Inicialmente, a árvore de decisão original tinha uma profundidade máxima de 12 e 54 folhas. A técnica de `GridSearchCV` foi utilizada para explorar diversos valores de `max_depth` (de 1 a 12), de modo a determinar a profundidade ótima. O resultado indicou que a melhor configuração foi `max_depth = 6`. A análise incluiu a visualização da árvore resultante, de profundidade máxima 6, e a avaliação de métricas de classificação.

A precisão global do modelo foi de 77%.

Poda de Complexidade de Custo (CCP) com `GridSearchCV`

A investigação da trajetória de poda (`cost_complexity_pruning_path`) foi realizada com o propósito de obter os valores de `ccp_alpha`. Diferentes valores de `ccp_alpha` foram explorados por meio do `GridSearchCV`, com o intuito de identificar o valor que maximiza o desempenho. A análise resultou num `ccp_alpha = 0.004449`. O modelo resultante, após a poda de CCP, foi avaliado através da matriz de confusão.

A matriz de confusão evidenciou o desempenho do modelo podado no conjunto de teste. Métricas de classificação indicaram uma precisão global de 77%, mantendo um equilíbrio satisfatório entre precisão e *recall* para diversas classes.

Conclusão sobre a técnica de Pruning

Ambas as estratégias demonstraram melhorias no desempenho em relação ao modelo original de Árvore de Decisão. A escolha entre tais abordagens pode depender das necessidades específicas do problema.

2.4 Random Forest

A presente análise salienta a otimização do desempenho de um modelo Random Forest por meio da técnica de ajuste de hiperparâmetros utilizando GridSearchCV. O objetivo é melhorar a capacidade de previsão do modelo, explorando diferentes combinações de hiperparâmetros.

Configuração Inicial do Modelo Random Forest

Inicialmente, o modelo de Random Forest foi configurado com 200 estimadores. Com intuito de otimizar o desempenho, foram selecionados os hiperparâmetros `criterion` (critério de divisão da árvore) e `max_depth` (profundidade máxima da árvore) para exploração.

Ajuste de Hiperparâmetros com GridSearchCV

A técnica de GridSearchCV foi utilizada para realizar uma busca no espaço de hiperparâmetros. Este processo envolveu a avaliação de diferentes combinações de critérios de divisão e profundidades máximas, utilizando validação cruzada para evitar overfitting.

Resultados Obtidos

O modelo Random Forest demonstrou uma capacidade melhorada de precisão em comparação com a sua configuração inicial. A análise das métricas de classificação revelou uma melhoria global na capacidade de previsão, destacando áreas específicas de força e identificando possíveis oportunidades para refinamentos adicionais.

A precisão global do modelo Random Forest otimizado foi de 0.86.

Além disso, a abordagem de ajuste de hiperparâmetros por meio do GridSearchCV ofereceu uma estratégia eficaz para melhorar o desempenho do modelo Random Forest. Os resultados obtidos contribuem significativamente para a compreensão do comportamento do modelo.

2.5 Análise dos Resultados dos Modelos de Machine Learning

A análise dos resultados dos modelos de Machine Learning adotados revela informações valiosas sobre o desempenho de cada abordagem na tarefa de classificação multiclasse. A tabela seguinte apresenta a precisão obtida por cada modelo.

Tabela 1: Precisão dos Modelos de Machine Learning

Modelo	**Precisão**
Regressão Logística	48%
Decision Tree Classifier	80%
Decision Tree com GridSearchCV	76%
Decision Tree com Pruning	77%
Random Forest	86%

Ao considerar os resultados obtidos pelos diversos modelos de Machine Learning na tarefa de classificação multiclasse, podem ser destacadas algumas conclusões.

A Regressão Logística apresentou uma precisão de 48%, indicando desafios significativos na adaptação aos padrões intrínsecos dos dados. As disparidades nas métricas de precisão, *recall* e *f1-score* entre as classes ressaltam a necessidade de melhorar específicos para melhorar sua performance.

O Decision Tree Classifier demonstrou robustez, alcançando uma precisão de 80%. A complexidade decisória evidenciada por essa abordagem sugere sua capacidade de capturar nuances nos dados. No entanto, variações nas métricas por classe indicam áreas potenciais de refinamento, especialmente para classes com desempenho inferior.

O processo de ajuste de hiperparâmetros utilizando GridSearchCV contribuiu para melhorar o Decision Tree, resultando em uma precisão média de 76%. O equilíbrio entre precisão e *recall* sugere uma melhoria geral na capacidade de generalização do modelo após otimização.

As estratégias de pruning aplicadas ao Decision Tree também se mostraram eficazes, mantendo um equilíbrio satisfatório entre precisão e *recall* e elevando a precisão para 77%. Essas abordagens oferecem insights valiosos sobre como mitigar a complexidade da árvore e melhorar seu desempenho.

O Random Forest, após o ajuste de hiperparâmetros, destacou-se como um dos modelos mais eficazes, atingindo uma precisão de 86%. Essa técnica de ensemble demonstrou uma capacidade robusta de previsão, evidenciando a importância da modificação de parâmetros.

Em resumo, a análise comparativa aponta que o modelo Random Forest se destaca como o mais eficaz entre os modelos avaliados, com desempenho notável. O Decision Tree também demonstrou um desempenho considerável, especialmente após a aplicação da técnica de pruning, ficando em uma posição relativamente próxima ao Random Forest.

2.5.1 Conclusão

Para concluir, os resultados obtidos neste estudo destacam a Regressão Logística como um ponto de partida na análise da tarefa de classificação multiclasse, apesar de enfrentar desafios na adaptação aos padrões intrínsecos dos dados. O Decision Tree Classifier exibiu robustez, atingindo uma precisão de 80%, com otimizações, como GridSearchCV e pruning, contribuindo para uma melhoria considerável. O Random Forest, como modelo de ensemble, destacou-se como a abordagem mais eficaz, alcançando uma precisão de 86% após modificação cuidadosa de hiperparâmetros, ressaltando a sua capacidade robusta em tarefas complexas de classificação multiclasse.

Em síntese, este estudo destaca a importância da seleção cuidadosa do modelo e da aplicação de estratégias de otimização para melhorar o desempenho. O Random Forest, em particular, demonstrou ser uma escolha promissora para tarefas similares, proporcionando um equilíbrio entre complexidade e eficácia na previsão. Estes resultados contribuem para o entendimento aprofundado do comportamento dos modelos, orientando decisões futuras em contextos similares de Machine Learning e classificação multiclasse.

3 Tarefa Dataset Competição

3.1 Análise do Conjunto de Dados da Competição

A primeira etapa do nosso trabalho envolveu a aquisição e exploração dos conjuntos de dados relacionados à competição. Três conjuntos de dados principais foram considerados para a análise: dados meteorológicos divididos em três períodos (setembro de 2021 a dezembro de 2021, janeiro de 2022 a dezembro de 2022 e janeiro de 2023 a abril de 2023) e dados de consumo de energia que acompanham esses períodos.

3.1.1 Dados de Consumo de Energia

Os conjuntos de dados de consumo de energia foram importados para os DataFrames *df_energia_1*, *df_energia_2*, e *df_energia_3*. Foram realizadas análises para identificar valores nulos usando mapas de calor.

Além disso, os DataFrames foram ajustados para melhor manipulação. No caso dos DataFrames *df_energia_1* e *df_energia_2*, as colunas 'Data' e 'Hora' foram renomeadas para 'date' e 'hour', respectivamente. No DataFrame *df_energia_3*, a coluna 'Injecao na rede (kWh)' foi adicionada para completar a correspondência com os outros conjuntos de dados.

As análises visuais fornecem insights preliminares sobre a presença de dados ausentes em alguns períodos. O próximo passo envolverá uma exploração mais aprofundada desses conjuntos de dados, incluindo estatísticas descritivas, análises temporais e correlações entre as variáveis meteorológicas e o consumo de energia.

3.1.2 Dados Meteorológicos

Procedemos à importação e exploração dos conjuntos de dados meteorológicos correspondentes a diferentes períodos: setembro de 2021 a dezembro de 2021 (`df_meteo_1`), janeiro de 2022 a dezembro de 2022 (`df_meteo_2`), e janeiro de 2023 a abril de 2023 (`df_meteo_3`).

Pré-Processamento e Visualização

Empregamos mapas de calor para identificar padrões de valores ausentes em cada conjunto de dados meteorológicos, oferecendo uma visão gráfica da distribuição da lacuna de dados. Este procedimento foi seguido por etapas de pré-processamento, durante as quais convertemos a coluna `'dt_iso'` para o formato de data e hora apropriado. Subsequentemente, criamos colunas adicionais para `'data'` e `'hora'`, reordenando o conjunto de dados para maior coesão. Colunas consideradas dispensáveis (`'sea_level'`, `'grnd_level'`, `'city_name'`, e `'weather_description'`) foram eliminadas para simplificação e foco nas variáveis relevantes.

A coluna `'rain_1h'` foi tratada quanto a valores nulos, sendo substituídos por zero, viabilizando uma manipulação mais eficiente desses dados.

Estatísticas Descritivas
Apresentamos as dimensões temporais dos conjuntos de dados meteorológicos:

- `N° de entradas em df_meteo_1`: 2928 entradas
- `N° de entradas em df_meteo_2`: 8760 entradas
- `N° de entradas em df_meteo_3`: 1752 entradas

Essas estatísticas fornecem uma compreensão da extensão temporal e da frequência horária dos registros meteorológicos em cada período.

3.1.3 Dados Meteorológicos do OpenMeteo

Realizamos uma análise sistemática dos conjuntos de dados `df_open_1`, `df_open_2` e `df_open_3`, informações essas que foram adquiridas no site `open-meteo.com` com o intuito de complementar os dados meteorológicos disponíveis. A escolha dos atributos específicos levou em consideração variáveis pertinentes para análises subsequentes, destacando-se 'temp', 'humidity', 'rain_1h', 'clouds_all', 'wind_speed', 'time', 'direct_radiation (W/m^2)', e 'diffuse_radiation (W/m^2)'.

Pré-Processamento e Transformação

O conjunto `df_open_1` abrange informações sobre cobertura de nuvens, radiação solar e outras variáveis meteorológicas para o período entre janeiro de 2021 e dezembro de 2022. Após as transformações, a estrutura do conjunto foi ajustada para conter as colunas 'date' e 'hour' no início, seguidas pelos atributos escolhidos.

O conjunto `df_open_1` possui 11.016 entradas, com variáveis como 'temp', 'humidity', 'rain_1h', 'clouds_all', 'wind_speed', 'direct_radiation (W/m^2)', e 'diffuse_radiation (W/m^2)'.

O conjunto `df_open_2` oferece dados semelhantes ao `df_open_1`, abordando apenas o período de janeiro a abril de 2023. As mesmas etapas de pré-processamento foram aplicadas para uniformizar a estrutura dos conjuntos de dados.

O conjunto `df_open_2` contém 2.256 entradas, com variáveis como 'temp', 'humidity', 'rain_1h', 'clouds_all', 'wind_speed', 'direct_radiation (W/m^2)', e 'diffuse_radiation (W/m^2)'.

O conjunto `df_open_3` concentra-se em variáveis meteorológicas em falta no dataset `df_meteo_3`, incluindo temperatura, humidade relativa, pressão atmosférica e velocidade do vento. As transformações seguiram os mesmos padrões aplicados aos conjuntos anteriores.

O conjunto `df_open_3` é composto por 504 entradas, destacando variáveis como 'temp', 'humidity', 'rain_1h', 'clouds_all', 'wind_speed', 'direct_radiation (W/m^2)', e 'diffuse_radiation (W/m^2)'.

3.2 Fusão/Merge dos Datasets

Após o processo de análise de dados, decidimos unir os datasets referentes aos anos de 2021, 2022 e 2023. A motivação principal para esta fusão foi criar uma visão mais abrangente e coesa dos padrões temporais e tendências associadas ao consumo de energia.

Procedimento:

Inicialmente, identificamos a necessidade de dividir a coluna 'time' em duas, 'date' e 'hour', para melhor representação temporal. Isso tornou mais claro o entendimento das informações e facilitou análises subsequentes.

Em seguida, realizamos a fusão dos conjuntos de dados de 2021 e 2022, criando um DataFrame consolidado ('df_merged_12'). Posteriormente, percebemos a importância de incluir dados do ano de 2023 para uma análise mais completa. Então, realizamos uma fusão adicional com dados específicos do ano de 2023, garantindo que o conjunto de dados resultante ('df_merged_3') incorporasse informações precisas e atualizadas.

Resultados Obtidos:

O DataFrame consolidado agora abrange os anos de 2021, 2022 e 2023, proporcionando uma visão do comportamento de consumo de energia ao longo desse período. A representação clara das datas e horas facilita a análise temporal e a identificação de padrões sazonais. Isto fortalecerá as análises futuras, pois a coleta e e fusão destes dados são fundamentais para embasar decisões e estratégias.

3.2.1 Substituição de Valores

Na fase seguinte, foi realizada a substituição de valores na variável 'Injecao na rede (kWh)'. O objetivo foi converter as categorias textuais em valores numéricos, facilitando o processo de treinamento do modelo. Isso envolveu a criação de um mapa de substituição, transformando categorias como 'None', 'Low', 'Medium', 'High' e 'Very High' em valores de 0 a 4, respectivamente.

3.2.2 Label Encoding

Além da abordagem de substituição de valores, optou-se por aplicar a técnica de 'Label Encoding' à variável 'Injecao na rede (kWh)'. Essa técnica consiste em transformar categorias em valores numéricos inteiros únicos, proporcionando uma representação mais eficiente para o treinamento do modelo.

3.3 Treino e Teste

O dataset foi particionado em conjuntos de treino e teste, uma prática essencial para avaliar a eficácia do modelo em dados não previamente observados. O conjunto de treino, composto por 60% dos dados, desempenha o papel de treinar o modelo, enquanto o conjunto de teste, representado pelos restantes 40%, é reservado para avaliar a capacidade de generalização do modelo em face de novos dados.

3.4 Modelos de Machine Learning para a Competição

3.4.1 Decision Tree Classifier

Inicialmente, empregamos um Decision Tree Classifier padrão sem ajuste de hiperparâmetros. Este modelo foi treinado no conjunto de treinamento e avaliado no conjunto de teste, resultando em uma accuracy de 86.2%.

3.4.2 Decision Tree com Grid Search:

Dada a complexidade do Decision Tree, decidimos realizar uma otimização de hiperparâmetros usando Grid Search. Especificamente, exploramos diferentes valores para os hiperparâmetros 'criterion' (critério de divisão) e 'max_depth' (profundidade máxima da árvore). Este processo envolveu treinar e avaliar o modelo para todas as combinações possíveis desses hiperparâmetros. O Grid Search revelou que a melhor combinação de hiperparâmetros para o Decision Tree Classifier foi 'criterion=entropy' e 'max_depth=7'. O modelo resultante, após a otimização, apresentou uma accuracy de 87.5% no conjunto de teste. Esta versão optimizada mostrou uma melhorar nas métricas de precisão, recall e f1-score para cada classe.

3.4.3 Decision Tree Classifier com Otimização de Profundidade Máxima

Exploramos a influência da profundidade máxima (*max_depth*) na performance do Decision Tree Classifier por meio de Grid Search. Inicialmente, identificamos a profundidade máxima padrão da árvore como 20. Utilizamos um conjunto de hiperparâmetros variando a profundidade máxima de 1 a 20.

O Grid Search revelou que a melhor profundidade máxima para maximizar o desempenho do modelo foi encontrada em 7. A árvore de decisão resultante, com essa profundidade otimizada, foi avaliada no conjunto de teste, onde alcançou uma accuracy de 88.11%.

O modelo otimizado com a profundidade máxima indicada mostrou um desempenho sólido, mantendo a precisão em níveis competitivos. A análise detalhada das métricas de classificação ressalta a capacidade do Grid Search em identificar parâmetros que resultam em melhor generalização do modelo.

3.4.4 Decision Tree Classifier com Cost-Complexity Pruning

Decidimos incluir a aplicação do Cost-Complexity Pruning (CCP) para fazer uma nova optimização. Inicialmente, calculamos os valores de alfa de CCP com o conjunto de treinamento. Este conjunto de alfas representa os trade-offs entre a complexidade da árvore e o seu desempenho nos dados de treinamento. A busca pelo melhor alfa foi realizada através do Grid Search e foi revelado que o alfa mais eficaz para o pruning foi encontrado em 0.0012. A árvore de decisão resultante, após a aplicação deste alfa otimizado, alcançou uma accuracy de 87.13% no conjunto de teste. Além disso evidenciou a eficácia do pruning na generalização do modelo e controle do overfitting.

3.4.5 Decision Tree Base com Bagging Classifier

Utilizamos um Bagging Classifier com uma árvore de decisão como estimador base. O processo incluiu a aplicação do Grid Search para otimização do hiperparâmetro 'n_estimators', o número de estimadores a serem utilizados no ensemble. A estratégia de validação adotada foi o Stratified Shuffle Split com 10 splits e um tamanho de teste de 40. O melhor resultado do Grid Search foi obtido com 'n_estimators=160', resultando em uma accuracy de 88.36% no conjunto de teste.

O Bagging Classifier demonstrou uma eficácia semelhante à árvore de decisão otimizada, com uma accuracy ligeiramente superior. A abordagem de ensemble proporcionou melhorias notáveis nas métricas de classificação, indicando a robustez e estabilidade do modelo.

3.4.6 Random Forest Classifier

Utilizamos um Random Forest Classifier com hiperparâmetros ajustados para melhor desempenho. O modelo foi treinado no conjunto de treinamento e avaliado no conjunto de teste, resultando em uma accuracy de 88.52%.

O Random Forest Classifier demonstrou uma accuracy superior em comparação com o Bagging Classifier com Decision Tree Base, alcançando um valor de 88.52%. Além disso, o modelo apresentou melhorias nas métricas de precisão, recall e f1-score para várias classes, destacando sua eficácia na classificação precisa das classes. O ajuste fino dos hiperparâmetros, como a profundidade máxima da árvore (max_depth) e o uso de bootstrap, contribuiu para a melhoria do desempenho geral do modelo.

3.4.7 Gradient Boosting Classifier

O Gradient Boosting Classifier foi empregado com os seguintes hiperparâmetros: n_estimators=100, learning_rate=1.0, max_depth=1 e random_state=2024. Após treinamento e avaliação no conjunto de teste, o modelo alcançou uma accuracy de 84.68%.

O Gradient Boosting Classifier exibiu uma precisão de 84.68%, demonstrando uma performance sólida na tarefa de classificação. Entretanto, as métricas

variam entre as classes, destacando a necessidade de um ajuste mais detalhado dos hiperparâmetros para otimizar o desempenho específico de cada classe.

3.4.8 Stacking Classifier

Para o Stacking Classifier, utilizamos uma combinação de Decision Tree e Random Forest como estimadores, com um final estimator sendo um Logistic Regression. Após treinamento e avaliação no conjunto de teste, o modelo atingiu uma accuracy de 88.72%.

O Stacking Classifier apresentou uma precisão de 88.72%, indicando um desempenho robusto na tarefa de classificação. As métricas de precisão, recall e f1-score para cada classe mostram melhorias notáveis em comparação com modelos individuais, destacando a eficácia da abordagem de ensemble para melhorar a generalização e o desempenho geral do modelo.

3.5 Análise dos Modelos de Machine Learning

Ao considerar globalmente o desempenho dos modelos de Machine Learning empregados na competição, observamos que o Stacking Classifier se destacou como o modelo mais eficaz, alcançando uma precisão de 88.72%. Esta abordagem de ensemble, que combinou Decision Tree e Random Forest como estimadores, seguidos por um Logistic Regression como final estimator, demonstrou uma capacidade robusta de generalização e precisão na tarefa de classificação multiclasse.

O Random Forest Classifier também apresentou uma performance notável, atingindo uma precisão de 88.52%. Esse modelo se beneficiou do ajuste fino de hiperparâmetros, incluindo a profundidade máxima da árvore (max_depth) e a utilização de bootstrap, destacando a importância da otimização para melhorar o desempenho global.

Outros modelos, como Bagging Classifier, Decision Tree com otimização de profundidade máxima e Decision Tree com Cost-Complexity Pruning, também exibiram resultados sólidos, superando o Decision Tree base. Cada um desses modelos reflete estratégias específicas, como a redução de overfitting, controle da complexidade da árvore e diversificação por meio de ensemble.

No entanto, é crucial ressaltar que a escolha do melhor modelo depende não apenas da precisão, mas também das considerações práticas, como interpretabilidade, tempo de treinamento e implementação. Cada modelo tem suas vantagens e limitações, e a seleção final deve levar em conta esses aspectos, além da métrica de desempenho.

Em resumo, o Stacking Classifier e o Random Forest Classifier se destacaram como os modelos mais promissores para a tarefa em questão, fornecendo resultados sólidos e consistentes. A análise detalhada das características individuais de cada modelo e a ponderação dos trade-offs associados são essenciais para a escolha do modelo mais adequado para a aplicação específica.

3.6 Resultados e Posicionamento na Competição

Na competição realizada na plataforma Kaggle, a equipe obteve um desempenho notável com um score de 0.87341, conquistando a 17ª posição entre 52 grupos participantes. A competição visava a previsão precisa da quantidade de energia elétrica gerada por painéis solares e injetada na rede elétrica a cada hora do dia. Essa tarefa envolveu o desenvolvimento de modelos de Machine Learning baseados em atributos como dados meteorológicos, informações geográficas e histórico de gasto e produção energética.

A métrica utilizada para avaliação na competição foi a accuracy, representando a percentagem de previsões corretas. A performance alcançada pela equipa, com um score de 87.341%.

3.7 Melhorias

- **Exploração de Outros Algoritmos:** Experimentar com uma variedade mais ampla de algoritmos de Machine Learning pode fornecer insights valiosos. A inclusão de modelos como Support Vector Machines (SVM) ou Redes Neurais pode ser explorada para avaliar seu desempenho em comparação com os modelos já utilizados.

- **Ajuste Fino dos Hiperparâmetros:** Realizar uma busca mais abrangente de hiperparâmetros pode levar a melhorias adicionais nos modelos. A otimização contínua de parâmetros, incluindo taxas de aprendizado, profundidades de árvores e número de estimadores, pode ser benéfica para melhorar a precisão das previsões.

4 Conclusão

A participação na competição de previsão de energia solar envolveu a aplicação de diversos modelos de Machine Learning para otimizar a estimativa da produção de energia por painéis solares. Exploramos modelos individuais, como Decision Trees e Gradient Boosting, e técnicas de ensemble, como Bagging e Stacking.

Os resultados refletem a complexidade do problema. A otimização de hiperparâmetros e o uso de técnicas de ensemble foram cruciais para melhorar o desempenho dos modelos. O Stacking Classifier destacou-se, combinando as Decision Trees e Random Forests, demonstrando precisão superior.

Na competição Kaggle, alcançamos um score de 0.87341, ficando em 17º lugar entre 52 grupos. Este desempenho destaca a eficácia das estratégias, mas sugere áreas de melhoria, como explorar outros algoritmos e buscar mais hiperparâmetros.

Em suma, este projeto contribuiu para a competição e enfatizou a importância do Machine Learning em problemas do mundo real, como previsão de energia solar. A aprendizagem contínua e experimentação são fundamentais para enfrentar desafios em domínios dinâmicos como o da energia renovável.