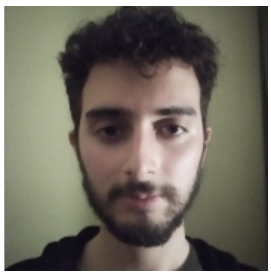
Universidade do Minho
Escola de Engenharia

Dados e Aprendizagem Automática
Grupo 11
2023/2024

João Paulo Peixoto Castro pg53929
Afonso Xavier Cardoso Marques pg53601
Renato André Machado Gomes pg54174
Vasco Rafael Barroso Gonçalves Rito a98728

Janeiro 2024




Índice

List of Figures

1 Introdução

O presente relatório enquadra-se na unidade curricular Dados e Aprendizagem Automática, na qual nos foi proposta a exploração, modelação e análise de dois datasets recorrendo a estratégias e algoritmos de machine learning.

O primeiro dataset, escolhido pelo grupo recorrendo à plataforma Kaggle, é focado na categorização de diamantes com base nos seus atributos, como tipo de corte, claridade, quilates e medidas de volume, tratando-se de um problema que se adapta bem tanto a algoritmos de regressão como de classificação. O segundo, fornecido pelos docentes, com o objetivo de ... o que constitui um problema de regressão.

O segundo dataset, fornecido pela equipa docente, é focado na previsão de injeção de energia na rede de acordo com vários fatores meteorológicos, tratando-se de um problema de classificação.

De modo a obtermos uma melhor compreensão, implementação e desenvolvimento dos datasets além de ajudar no planeamento dos mesmos, o grupo decidiu optar pela metodologia CRISP-DM. Esta metodologia é composta por seis etapas: estudar o negócio, estudar os dados, preparar os dados, modelar, avaliar e desenvolver.

2 Apresentação e Exploração dos Datasets

A primeira etapa para a concretização deste projeto consistiu no entendimento e exploração dos datasets, dado que estes podem conter dados incompletos, incoerentes ou errados. Esta etapa é importante para entender a preparação necessária para atingir o objetivo de cada um dos datasets além de compreender qual o método de avaliação do modelo mais apropriado.

Em seguida apresentamos os datasets, explicando cada atributo bem como os nodos utilizados para a exploração dos mesmos e o objetivo final pretendido.

2.1 Dataset Diamantes

Este dataset foi escolhido pelo grupo através da plataforma Kaggle. Efetivamente, este dataset foi usado com dois targets distintos. Inicialmente com o objetivo de prever o preço nos diamantes sendo este um problema de regressão. De seguida utilizamos como target a qualidade de corte de cada um sendo este um problema de classificação.

O dataset apresenta 10 atributos e 53940 entries. Dos 10 atributos 3 são categóricos e 7 são numéricos. Apresentamos em seguida os atributos iniciais:

- carat: peso em quilates do diamante;
- cut: qualidade do corte (Fair, Good, Very Good, Premium, Ideal);
- color: cor do diamante, de J(pior) a D(melhor);
- clarity: medida de quão claro é o diamante(I1 (pior), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (melhor));
- depth: valor da profundidade do diamante;
- table: faceta plana na superfície que se pode ver quando se olha para o diamante de cima;
- price: preço em dólar americano;
- x: comprimento em mm (0 - 10.74);
- y: largura em mm (0 - 58,9);
- z: altura em mm (0 - 31.8);

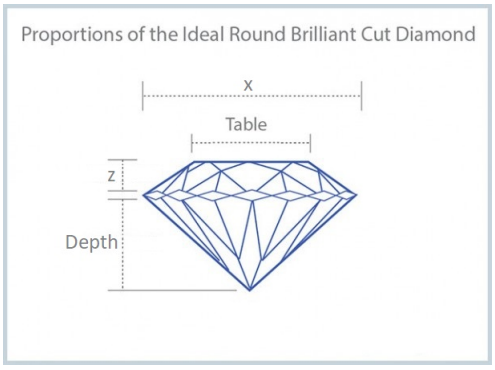


Figure 1: Medidas do diamante

Conseguimos obter uma ideia da distribuição de diamantes pelo atributo de qualidade de corte.

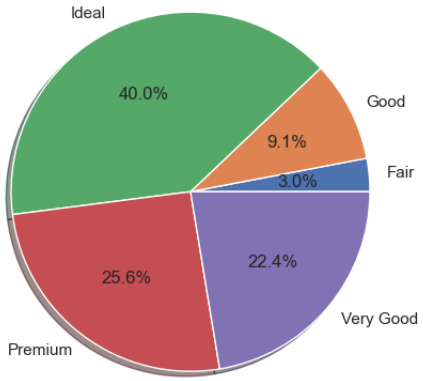


Figure 2: Distribuição por corte

A figura seguinte apresenta o output obtido após fazer a exploração das tendências estatísticas para os dados. Este permitiu obter informação sobre os valores externos das diversas features, bem como a sua média, desvio padrão e valores máximos e mínimos.

	carat	depth	table	price	x	y	z
count	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000
mean	0.797940	61.749405	57.457184	3932.799722	5.731157	5.734526	3.538734
std	0.474011	1.432621	2.234491	3989.439738	1.121761	1.142135	0.705699
min	0.200000	43.000000	43.000000	326.000000	0.000000	0.000000	0.000000
25%	0.400000	61.000000	56.000000	950.000000	4.710000	4.720000	2.910000
50%	0.700000	61.800000	57.000000	2401.000000	5.700000	5.710000	3.530000
75%	1.040000	62.500000	59.000000	5324.250000	6.540000	6.540000	4.040000
max	5.010000	79.000000	95.000000	18823.000000	10.740000	58.900000	31.800000

Figure 3: Tendências estatísticas

Através da análise de correlação apresentada na figura abaixo é possível medir a força e direção da associação entre duas variáveis, o que pode fornecer informação útil acerca das features que devem incorporar os modelos de aprendizagem automática. Tal como é possível verificar que existem features com um elevado grau de correlação (próximos de 1), tal como o par price-carat, o par price-x, price-y e price-z. É possível verificar também que o par table-depth tem uma baixa correlação, sendo esta correlação negativa. Importante salientar que este par é o único par que apresenta correlação negativa.

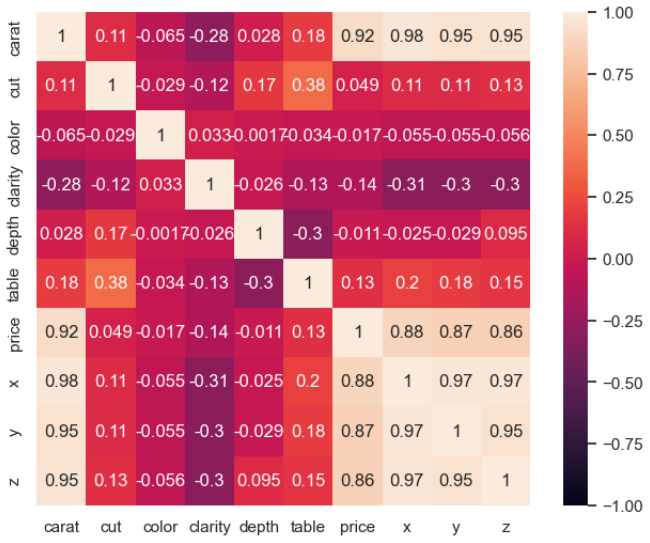


Figure 4: Correlação Linear

2.2 Dataset de Energia

O dataset de Energia é um dataset de series temporais, que foi fornecido para a análise e compreensão do consumo energético ao longo do tempo compreendido de 2021 a 2022. Este dataset contém 6 atributos em cada registo, cobrindo aspetos como o consumo de energia em diferentes horários e o auto-consumo. Com 2256 registos para 2021 e 8760 para 2022, o dataset oferece uma visão detalhada do uso de energia, permitindo análises temporais e de padrões de consumo.
O dataset é constituído pelos seguintes atributos:
- **Data**: Data do registo
- **Hora**: Hora do registo
- **Normal (kWh)**: Consumo de energia normal em quilowatts-hora
- **Horário Económico (kWh)**: Consumo de energia no horário económico em quilowatts-hora
- **Autoconsumo (kWh)**: Consumo de energia auto consumido em quilowatts-hora
- **Injeção na rede (kWh)**: Energia injetada na rede em quilowatts-hora

2.3 Dataset de Metereologia

O dataset de Produção Energética, que abrange os anos de 2021 e 2022, foi disponibilizado para permitir a análise das condições climáticas em diferentes períodos. Este dataset é composto por 15 atributos, que incluem informações como temperatura, pressão atmosférica, humidade e descrição do tempo. Este conjunto de dados possui 2928 registos para 2021 e 8760 para 2022, ele fornece dados detalhados que são fundamentais para estudos relacionados ao clima e às suas variações e métricas.
O dataset é constituído pelos seguintes atributos:

- **dt**: Timestamp do registo
- **dt_iso**: Data e hora do registo em formato ISO
- **city_name**: Nome da cidade
- **temp**: Temperatura atual em graus Celsius
- **feels_like**: Sensação térmica em graus Celsius
- **temp_min**: Temperatura mínima em graus Celsius
- **temp_max**: Temperatura máxima em graus Celsius
- **pressure**: Pressão atmosférica em hPa
- **sea_level**: Nível do mar
- **grnd_level**: Nível do solo
- **humidity**: Humidade relativa do ar em percentagem
- **wind_speed**: Velocidade do vento em m/s
- **rain_1h**: Quantidade de chuva em 1 hora em mm (dados faltantes)
- **clouds_all**: Cobertura de nuvens em percentagem
- **weather_description**: Descrição textual do tempo

3 Preparação dos Datasets

Após uma análise completa efetuada para cada um dos datasets, segue-se a preparação dos dados com o objetivo de eliminar qualquer informação não pretendida, converter dados para formatos mais amigáveis aos algoritmos usados e algum feature engineering.

Começamos por efetuar uma limpeza de dados, tratando dos missing values e outliers. De seguida é realizado um aumento da informação de acordo com o pretendido do problema. Deste modo, os datasets ficam prontos para a avaliação final do modelo desenhado. Apresentamos em seguida todo o tratamento de dados efetuado em cada um dos datasets:

3.1 Dataset Diamantes

3.1.1 Tratamento de dados global

- **Verificação de missing values e valores únicos**: De forma a ter a certeza que não havia valores por preencher ou atributos com valores únicos que pudessem ser eliminados.
- **Deteção de outliers**: Para depois na implementação dos modelos se proceder à sua devida remoção.
- **Conversão de dados categóricos em valores numéricos**: Para facilitar a aplicação dos vários modelos desenvolvidos e permitir a realização de futuras modificações.
- **Ajustes nos dados numéricos com fatorização**: Aplicação de técnicas de enconding e fatorização para tratar os dados. Decidimos manter apenas a fatorização.
- **Normalização dos dados numéricos**: Após ter tudo convertido para números e fatorizados, procedeu-se à normalização destes valores.

3.2 Dataset de Produção Energética e Sustentabilidade

3.2.1 Tratamento de dados global

- **Tratamento de NaN**: Definição de uma lista ‘allowed_nan‘, que contém valores a serem tratados como NaN durante a leitura dos dados usando ‘pd.read_csv‘.
- **Leitura dos Datasets de Treino**: Quatro conjuntos de dados de treino (dois de energia e dois de meteorologia) são lidos e concatenados horizontalmente para formar os conjuntos completos ‘energia‘ e ‘meteo‘.
- **Ajustes nos Dados de Energia**: A função ‘joinzero‘ é aplicada para garantir que os valores na coluna ’Hora’ tenham dois dígitos, já que após a leitura os zeros à esquerda desapareceram. Em seguida, as colunas ’Hora’ e ’Data’ são combinadas, e a coluna ’Hora’ é removida.
- **Ajustes nos Dados de Meteorologia**: A função ‘cleanTime‘ é aplicada para extrair a parte relevante da coluna ’dt_iso’, renomeada para ’Data’.
- **Junção dos Dados de Energia e Meteorologia**: Os conjuntos ‘energia‘ e ‘meteo‘ através de uma operação ‘inner join‘ são misturados através da coluna ’Data’, resultando no conjunto final chamado ‘energiPro‘.
- **Leitura dos Datasets de Teste**: Dois conjuntos de dados de teste (um de energia e outro de meteorologia) são lidos e ajustados de maneira semelhante aos conjuntos de treino. A mesclagem é realizada com base na coluna ’Data’, formando o conjunto de teste ‘testEnergiPro‘

4 Modelos Desenvolvidos

Finalizada a preparação de dados dos datasets, passamos para o desenvolvimento dos modelos de aprendizagem. Deste modo, conseguimos testar todo o tratamento que foi realizado e explicado na secção anterior, dando uma resposta ao problema proposto para cada um dos datasets.

De seguida iremos apresentar os algoritmos testados,bem como as suas características e os parâmetros de treino utilizados.

4.1 Dataset Diamantes

Todos os modelos foram realizados utilizando o tratamento de dados global mencionado anteriormente à exceção do Clustering K-Means.

4.1.1 Modelos de Regressão

Linear Regression: algoritmo utilizado para modelar a relação linear entre uma variável dependente (o target) e uma ou mais variáveis independentes (atributos), com o objetivo de fazer previsões ou entender a relação entre essas variáveis. O modelo assume uma relação linear, representada por uma linha reta, e procura encontrar os coeficientes que melhor se ajustam aos dados observados. Esses coeficientes são usados para fazer previsões com base nas novas entradas das variáveis independentes.

O primeiro passo tomado para este modelo foi de garantir que não havia valores de outliers e fazer a normalização de todos os valores nos atributos 'carat', 'depth', 'table', 'price', 'x' , 'y' e 'z'. Este tratamento foi usado nos modelos subsequentes. O target usado foi o atributo 'price'.

Obteu-se mm MAE (Erro Absoluto Médio) de aproximadamente 0.0321 o que indica que, em média, as previsões do modelo têm um erro absoluto de cerca de 0.0321 unidades em relação aos valores reais. O MSE (Erro Quadrático Médio) foi aproximadamente 0.00261 o que indica que o erro médio quadrático entre as previsões do modelo e os valores reais é de cerca de 0.00261. Tanto o MAE como o MSE são medidas de erro quadrático, logo valores menores são melhores. Neste caso, ambos os valores são bastante pequenos. A raiz quadrada do MSE (RMSE) é aproximadamente 0.0511 e fornece uma interpretação mais intuitiva do erro, na mesma unidade que a variável de resposta original. Um RMSE pequeno indica que o modelo está a fazer boas previsões em relação aos valores reais.

Finalmente O R^2 (Coeficiente de Determinação) é aproximadamente 0.8817. Isso significa que modelo explica cerca de 87.33% da variabilidade presente nos dados normalizados.

Decision Tree Regressor: Foi empregada uma Decision Tree Regressor como parte do processo de modelagem para prever valores numéricos da variável alvo 'price' com base nas seguintes features: 'carat', 'depth', 'table', 'x', 'y', 'z', 'cut', 'color' e 'clarity'.

Depois de implementado, e com tratamento igual ao modelo anterior, conseguiu obter-se resultados melhores que o modelo precendente (Linear Regression), com valores MAE = 0.0151, MSE = 0.001, RMSE = 0.0282, R^2 = 0.9639. É de notar que R^2 ficou mais próximo de 1, pelo que as previsões são mais precisas.

Redes Neuronais (Multilayer Perceptron): é modelo de aprendizagem inspirado no funcionamento do cérebro humano. O MLP consiste em camadas de neurónios interconectados, com uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Cada conexão entre neurónios tem um peso associado, e a rede aprende ajustando esses pesos durante o treino. A ativação de cada neurónio é determinada pela soma ponderada das entradas, passando por uma função de ativação não linear.

No nosso caso foi usada para descobrir quais os melhores parâmetros para aplicar a um modelo de KerasRegressor, tendo sido necessário remover completamente todos os atributos categóricos (que estavam convertidos para valores inteiros enquanto que o MLP só usa float). O target foi o atributo 'price' e o R^2 foi 0.865, pior que o modelo anterior.

Emsemble Learning (Gradient Boosting Regressor): é uma abordagem que combina vários modelos de aprendizagem fracos para formar um modelo robusto e mais poderoso. Aqui, os modelos são treinados sequencialmente, com cada novo modelo corrigindo os erros do modelo anterior. O modelo final é uma combinação ponderada dos modelos individuais, resultando em uma previsão mais precisa.

O target usado foi novamente 'price' com de $R^2 = 0.971$, o melhor resultado obtido de todos os modelos para problemas de regressão até ao momento.

4.1.2 Modelos de Classificação

Decision Tree Classifier: Partindo agora para alguns algoritmos de classificação, que se utilizam especialmente quando lidamos com atributos categóricos. Neste primeiro caso iremos aplicar um algoritmo com árvore de decisão de classificação.

Procedemos à remoção dos atributos 'color' e 'clarity' para evitar que o modelo ficasse demasiadamente bem treinado (overfitting). Usamos como target o atributo 'cut'. O valor de precisão obtido rondou os 70.75%.

Logistic Regression: foi aplicada uma técnica de Regressão Logística para modelar e prever a probabilidade da variável 'cut' com base nas seguintes features: 'carat', 'depth', 'table', 'x', 'y', 'z', 'color' e 'clarity'. O conjunto de dados foi dividido aleatoriamente em conjuntos de treino (70%) e teste (30%), e o modelo foi treinado utilizando média estocástica.

Os valores de precisão, recall e accuracy são todos iguais, podendo interpretar isso como um indicativo de que o modelo está a fazer previsões consistentes. No entanto, este modelo provou ser menos preciso do que o anterior, descendo a accuracy de 70% para 63%.

Clustering com K-Means: No modelo de clustering K-Means, foi inicialmente utilizada uma heurística para escolher um número apropriado de *clusters*, o método do cotovelo. Daí, foram utilizadas as *labels* 'carat', 'depth', 'table', 'price', 'y', 'z', surgindo um número de *clusters* a serem utilizados. Não foram usadas nenhuma *label* categórica nem o 'x' por apresentar elevada correlação tanto com 'y' como com 'z', o que poderia levar a redundância de informação.

Com base nesse número de *clusters* inicial, começou a desenvolver-se o algoritmo K-Means. Foram utilizados os dois *clusters* (que foram obtidos através do Elbow method), mantendo a *feature matrix* e utilizando como *label* a variável categórica 'cut'.

4.2 Dataset de Produção Energética e Sustentabilidade

Decision Tree: algoritmo utilizado para categorizar ou prever o valor de 'weather_description', aprendendo regras de decisão simples inferidas através de dados anteriores. Uma Decision Tree é um grafo hierarquizado (árvore) em que cada ramo representa a seleção entre um conjunto de alternativas e as folhas representam uma decisão.

Para este modelo tivemos primeiramente, que verificar as diferentes categorias presentes em 'weather_description'. Cada categoria textual em 'weather_description' é mapeada para um valor numérico. Por exemplo, 'sky is clear' é mapeado para 1, 'few clouds' para 2, e assim por diante. Essa conversão é crucial para permitir que o modelo de Decision Tree processe esses dados,já que ele requer entradas numéricas. A coluna 'Data' é transformada para um formato numérico dd-mm. Isso é feito para representar os dados de forma periódica, facilitando o reconhecimento de padrões temporais pelo modelo. Para lidar com potenciais valores nulos em 'rain_1h', realizamos uma estimativa desses valores com base na descrição meteorológica. Diferentes condições climáticas são associadas a estimativas de precipitação, como 0.15 para 'broken clouds'.

Random Forest: Um modelo Random Forest é um modelo de machine learning que opera construindo uma multitude de árvores de decisão durante o treinamento e produzindo a classe que é o modo das classes (classificação) ou a média das previsões (regressão) das árvores individuais. Este modelo é um tipo de ensemble learning, que é uma técnica que combina várias modelos de machine learning para criar um modelo mais poderoso e preciso.

Inicialmente a coluna 'Injeção na rede (kWh)' é convertida para valores numéricos usando um mapeamento, onde categorias como 'None', 'Low', 'Medium', 'High', 'Very High' são mapeadas para valores numéricos de 1 a 5. Utiliza-se uma abordagem semelhante à descrita anteriormente para estimar valores de 'rain_1h' com base na descrição meteorológica. Os dados são divididos em conjuntos de treino e teste, com 20% dos dados reservados para teste. O modelo de Random Forest é criado com os seguintes hyper parâmetros: 'max_features' como 'sqrt', 'min_samples_leaf' como 4, 'min_samples_split' como 2, e 'n_estimators' como 50. Este processo mostra como o modelo Random Forest é aplicado aos dados preparados, com foco na classificação da variável 'Injeção na rede (kWh)'. O modelo é treinado para entender as relações entre as variáveis e fazer previsões sobre a categoria de 'Injeção na rede (kWh)' para novos dados.

Clustering: Este algoritmo de aprendizagem não-supervisionada irá agrupar as observações do nosso dataset em grupos baseando-se na sua similiaridade de acordo com a relação humidade/temperatura. Estes atributos são contínuos e apresentam uma correlação de aproximadamente 0.5 e -0.5, respetivamente. O declive descreve como a humidade relativa do ar varia com as alterações na temperatura.

Após o particionamento (70 treino - 30 teste) iremos realizar o clustering usando o k-means e o k-medoids e procedemos à normalização dos dados, para evitar que haja uma atribuição desviada de uma observação que possua uma escala maior num dos clusters. Por outro lado, fizemos o mapping dos valores categóricos para numéricos. Como já sabemos de antemão o número de categorias da nossa label, podemos atribuir o número ideal de clusters.

Rede Neuronal MLP: Para a aplicação da rede neuronal foi realizado em primeiro lugar o scaling dos dados e a conversão de tipos para float. Após o particionamento hold-out, construímos uma árvore com a seguinte topologia:
- ReLu como função de ativação
- topologia sequencial
- 4 camadas (3 camadas intermédias - 8; 16; 32)

- MAE como função de loss
- Adam como optimizador
- ratio de aprendizagem 0.01
- MAE MSE como métricas

De seguida, recorremos ao GridSearch para encontrar a melhor função otimizadora entre 'SGD','RMSprop','Adagrad' e para além disso, foi efetuado o cross-validation com kfold=8. Para o modelo de KerasRegressor aplicamos um número de epochs 3 vezes superior ao número de atributos. Analisando a performance do modelo podemos averiguar que o valor da validação segue ainda que com algumas oscilações o valor de treino.

Support Vector Machine: O Support Vector Machine (SVM) é um modelo de aprendizado de máquina poderoso e versátil, usado principalmente para tarefas de classificação, mas também para regressão. Em sua forma mais simples, o SVM é utilizado para classificar dados linearmente separáveis, encontrando o hiperplano que melhor divide as classes de dados com a maior margem possível. O hiperplano é escolhido de forma a maximizar a distância entre as linhas de suporte mais próximas de cada classe. Estas linhas de suporte são determinadas pelos pontos de dados mais próximos do hiperplano e são chamados de vetores de suporte.

Inicialmente diversos atributos são normalizados utilizando o MinMaxScaler. Isso inclui a transformação das colunas 'Data', 'Normal (kWh)', 'Horário Económico (kWh)', 'Autoconsumo (kWh)', 'temp', 'pressure', 'humidity', 'wind_speed', 'rain_1h', 'clouds_all' e 'weather_description'. A coluna 'Injeção na rede (kWh)' é separada do dataset e usada como variável alvo (target), enquanto as outras colunas são usadas como recursos (features). Este dataset é dividido em conjuntos de treino (70%) e teste (30%). Este processo ilustra como o modelo SVM é aplicado e avaliado, focando-se na classificação da variável 'Injeção na rede (kWh)'. O SVM é um modelo poderoso, especialmente eficaz em espaços de alta dimensão, e é frequentemente utilizado em problemas de classificação. A otimização de hiperparâmetros, embora comentada, é uma etapa crucial para melhorar o desempenho do modelo.

Stacking Ensemble Learning : O Stacking Ensemble Learning é uma técnica avançada que combina as previsões de múltiplos modelos de machine learning para aumentar a precisão e eficácia na previsão. Funciona ao treinar diversos modelos base independentemente e usando as suas previsões como entradas para um modelo de meta-aprendizagem. Este modelo de segundo nível é treinado para otimizar e combinar as previsões dos modelos base, capitalizando assim as forças individuais de cada modelo. O Stacking é eficaz pela sua capacidade de sintetizar diferentes padrões e tendências capturadas por vários modelos, mas exige uma gestão cuidadosa para evitar overfitting e garantir a diversidade entre os modelos base.

5 Resultados Finais e Análise Crítica

Tendo todos os modelos de aprendizagem testados,apresentamos agora o algoritmo final escolhido com os resultados obtidos para ambos os datasets. Apresentamos ainda uma análise crítica a esses mesmos resultados.

5.1 Dataset Diamantes

Na seguinte tabela temos um sumário dos valores registados nos modelos de Machine Learning de regressão previamente explicados.

Modelos	**MAE**	**MSE**	**RMSE**	**R2̂**
Linear Regression	0.032	0.002	0.0511	0.881
Decision Tree Regressor	0.015	0.002	0.028	0.963
Redes Neuronais (Multilayer Perceptron)	0.032	0.003	0.056	0.858
Emsemble Learning (Gradient Boosting Regressor)	0.015	0.001	0.025	0.971

Table 1: Tabela 1 de resultados do dataset de Diamantes

Na seguinte tabela temos um sumário dos valores registados para a accuracy dos modelos de Machine Learning para classificação previamente explicados.

Modelos	**Accuracy**
Decision Tree Classifier	70.752%
Logistic Regression	63.304%
Clustering com K-Means	59%

Table 2: Tabela 2 de resultados do dataset de Diamantes

Após analisarmos os resultados obtidos é possível concluir que os algoritmos de classificação obtiveram, no geral, uma pior prestação (especialmente o clustering com K-Means) quando comparados com os algoritmos de regressão. Dentro do espetro da regressão, todos os modelos tiveram um boa prestação, com destaque para o modelo de Emsemble Learning com Gradient Boosting Regressor com um valor de R^2 de 0.971 e o modelo de Decision Tree Regressor a seguir com 0.963. Na classificação o melhor foi o modelo de Decision Tree Classifier com precisão de 70.752% seguido do Logistic Regression com 63.304%. Relativamente ao clustering k-means onde se obteve apenas 59%, um valor bastante mais baixo do que a expetativa, concluímos que este poderá ser devido ao facto de ser um algoritmo de aprendizagem não supervisionada e a preparação dos dados global poderá não ter sido a mais adequada.

5.2 Dataset Energia e Meteorologia

Na seguinte tabela temos um sumário dos valores registados para a accuracy dos modelos de Machine Learning previamente explicados.

Era esperado que o modelo Random Forest tivesse um desempenho superior ao da Decision Tree simples, devido à sua natureza de ensemble que geralmente fornece uma maior precisão, o que é confirmado com uma precisão de 86,0%, em comparação com 85,0% da Decision Tree. A leve diminuição na precisão da Decision Tree com pruning

Modelos	Accuracy
Decision Tree	85,0%
Decision Tree com prunnig	84,0%
Random Forest	86,0%
Clustering com KMeans	45,0%
Support Vector Machine	83,0%
Rede neural artificial - MLP	83.1%
Stacking Ensemble Learning	84,97%
Boosting Ensemble Learning	84.6%

Table 3: Tabela de resultados do dataset de Energia e Meteorologia

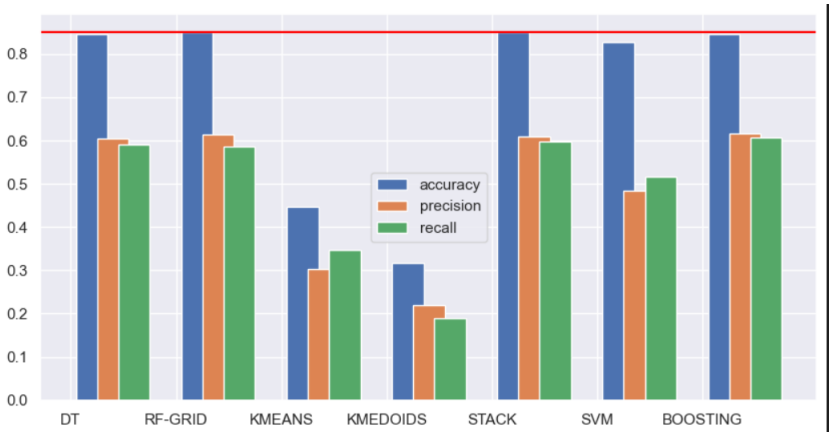


Figure 5: Métricas de avaliação dos modelos desenvolvidos

(84,0%) pode ser um reflexo da redução da complexidade do modelo, evitando overfitting mas perdendo alguma precisão. A técnica de Clustering com KMeans apresenta uma precisão significativamente mais baixa (45,0%), o que pode indicar uma inadequação para este tipo de tarefa de classificação. O desempenho do Support Vector Machine (83,0%) e da Rede Neural Artificial - MLP (79,5%) está em linha com o esperado, demonstrando competência, mas sem superar os métodos de ensemble. O Stacking Ensemble Learning, apesar de ser uma técnica sofisticada, atinge uma precisão de 84,97%, o que é competitivo, mas neste caso, não ultrapassa significativamente os modelos individuais como o Random Forest.

6 Sugestões e Recomendações

Por fim, consideramos ser necessário realizar uma análise dos resultados obtidos no contexto do problema de cada um dos datasets e aos modelos finais desenvolvidos. Deste modo,conseguimos dar algumas sugestões e recomendações sobre como obter os melhores resultados para os problemas em questão.

A mais óbvia das melhorias que podiam ser feitas são nos algoritmos de aprendizagem com clustering (KMeans) nas duas tarefas. Inicialmente, decidimos experimentar vários valores para clusters no KMeans mas não teve um impacto significativo nos resultados finais. De seguida, experimentamos fazer pequenas alterações no pré-processamento dos dados, mas os resultados ou tinham valores muito baixos ou não tinham valores nenhuns devido a erros provocados pelas alterações feitas. Decidimos que no final seria mais sensato manter os valores baixos mas "estáveis" que tínhamos obtido.

7 Conclusão

Dado por concluído o trabalho prático, consideramos importante realizar uma análise crítica, e ainda, realçar os pontos positivos e negativos do trabalho realizado.

A extensa preparação de dados elaborada em ambos os datasets e a boa organização dos modelos desenvolvidos salienta o bom entendimento dos problemas com o objetivo de melhorar o resultado final. O grupo ter optado por desenvolver diversos modelos para testagem de resultados de modo a obter os melhores resultados constitui um ponto positivo do nosso trabalho.

Por fim, o grupo considera que o trabalho realizado é bastante positivo, pois cumpre todos os requisitos propostos no enunciado.

8 Anexos

8.1 Dataset Diamantes

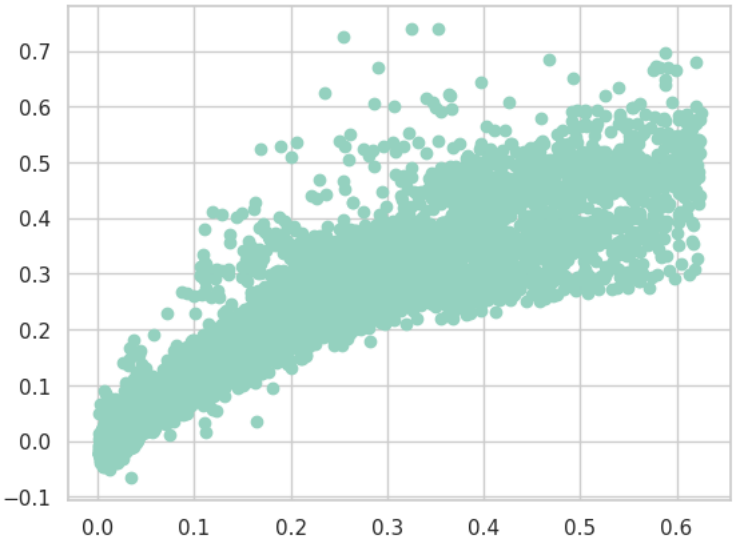


Figure 6: Previsões do Linear Regressor

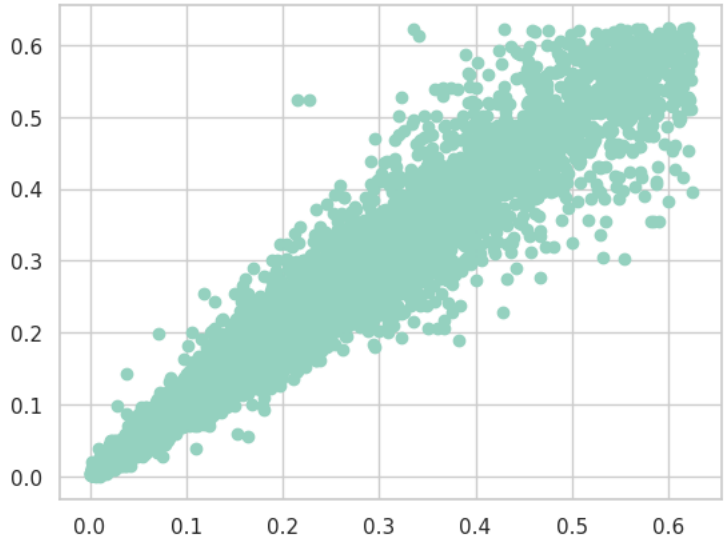


Figure 7: Previsões do Decision Tree Regressor

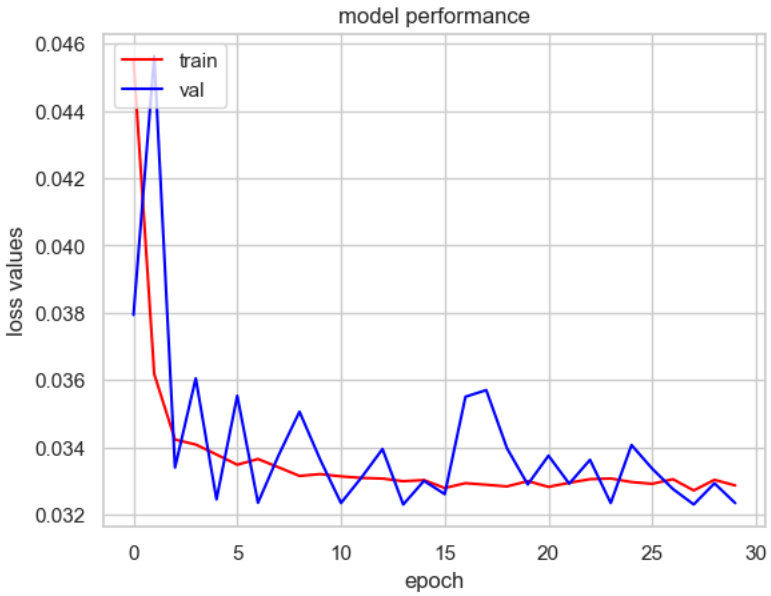


Figure 8: Valor de loss por epoch no MLP

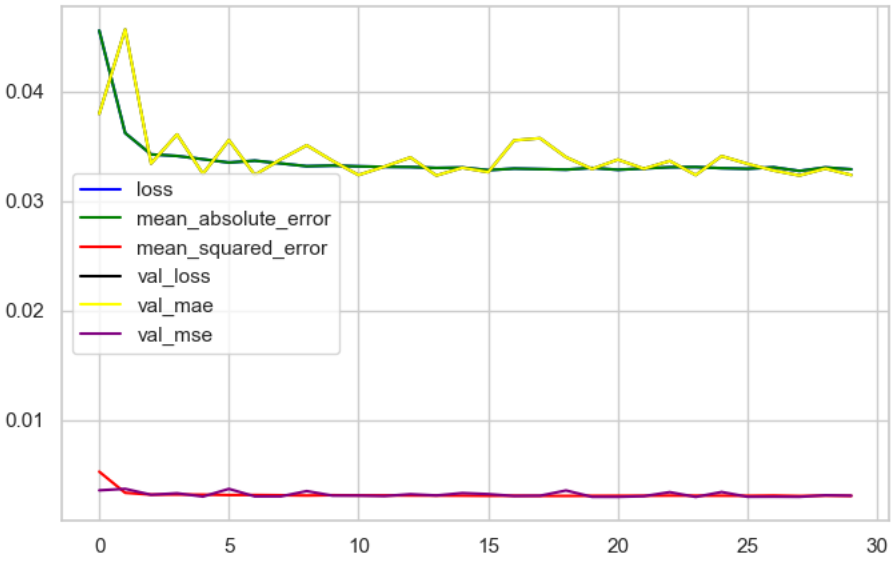


Figure 9: Variação de valores em MLP

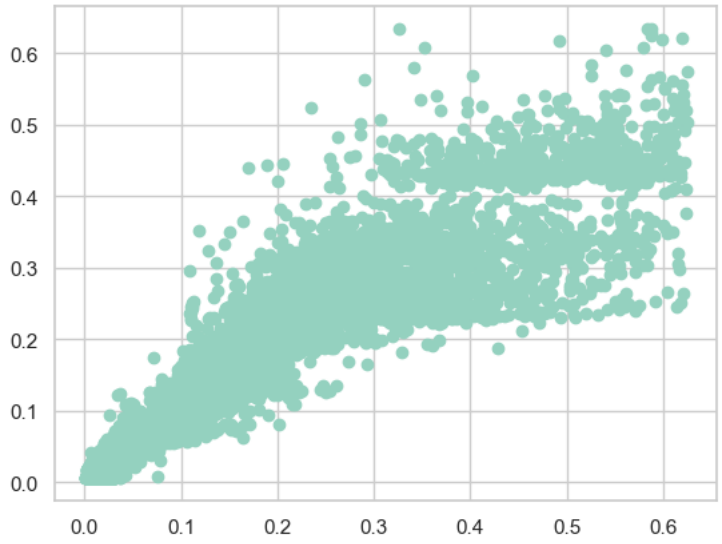


Figure 10: Previsões do MLP

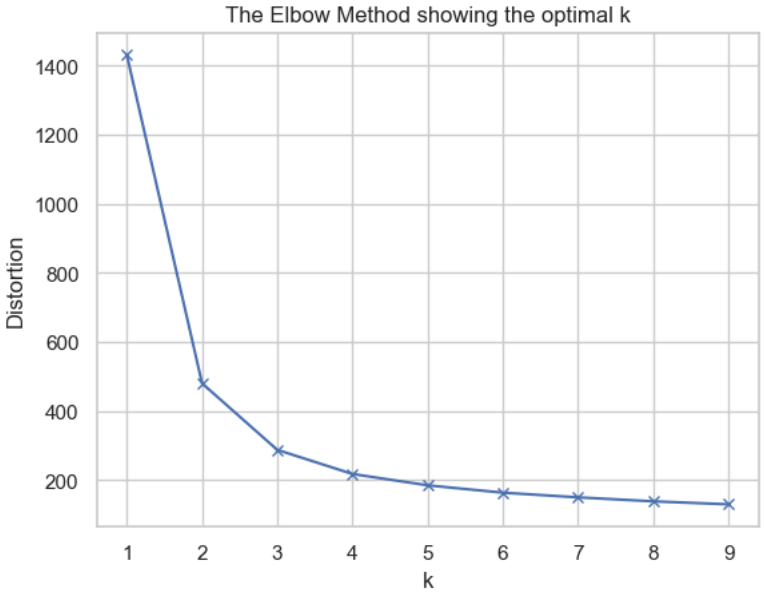


Figure 11: Elbow Score com o número ótimo de clusters

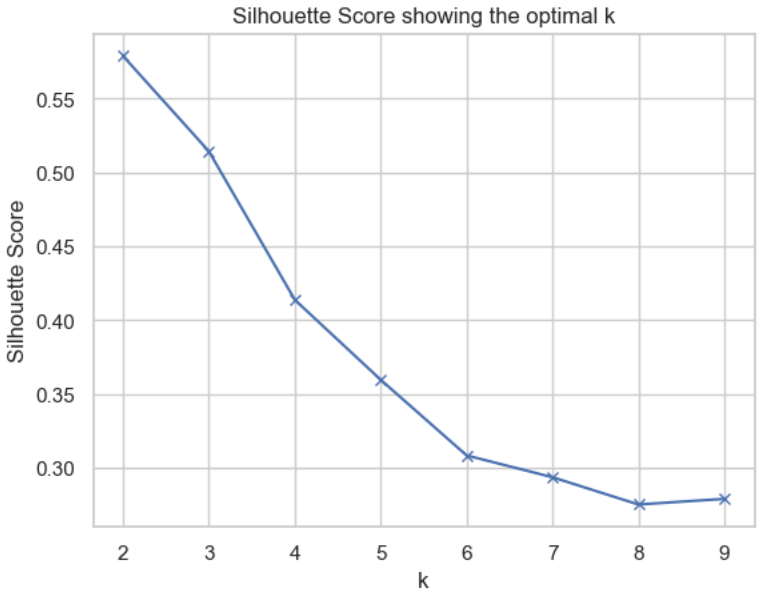


Figure 12: Silhouette Score com o número ótimo de clusters

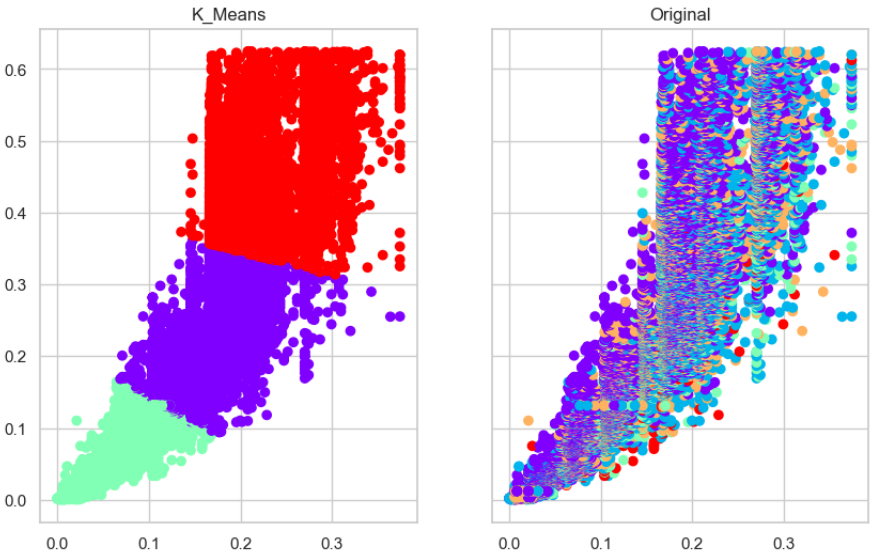


Figure 13: Resultado do K-Means

8.2 Dataset de Produção Energética e Sustentabilidade

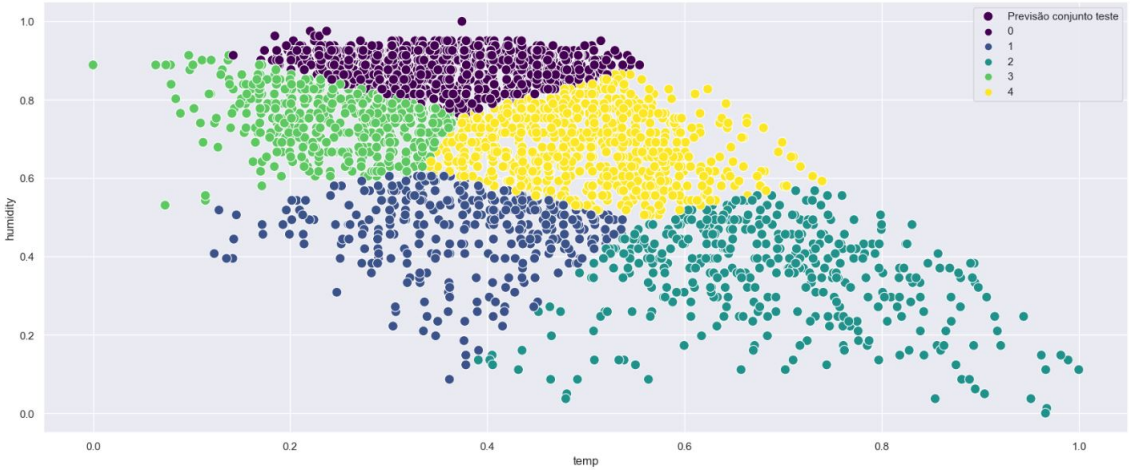


Figure 14: Previsão de Injeção na rede (KWh) clustering com K-Means baseando na relação Humidity/temp

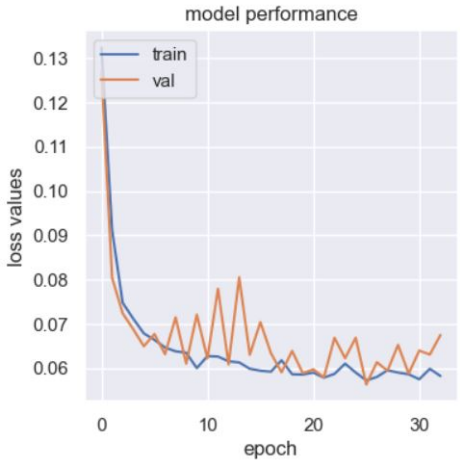


Figure 15: Performance do modelo MLP

Neste caso, a curva de treino demonstra um bom fit ainda que a curva de aprendizagem para validação loss revela alguns ruídos que não seguem tão bem a curva de treino. Algumas das soluções possíveis para achatar essa curva passariam por aumentar o número de observações para validação ou então ajustar o valor do k folds no processo de cross-validation.