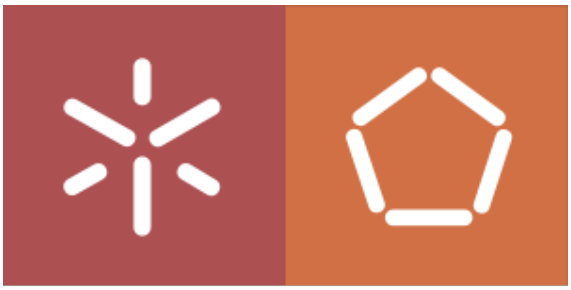

Universidade do Minho

Mestrado Engenharia Informática

Dados e Aprendizagem Automática (2023/24)

Grupo 17:
Diogo Costa PG53783
João Brito PG53944
Henrique Ribeiro A95323
Telmo Oliveira PG54247

Braga, 25 de junho de 2024

1 Introdução

O presente relatório tem como objetivo acompanhar o desenvolvimento do trabalho prático realizado ao longo do semestre, abordando dois principais grupos de análise. Um desses grupos será exclusivamente dedicado ao dataset da competição, composto por um conjunto de dados de treino e um conjunto de dados de teste fornecidos pelos docentes. O segundo grupo concentra-se no dataset escolhido pelo nosso grupo de trabalho (diabetes_data.csv).

O trabalho prático foi executado utilizando a linguagem de programação Python, que oferece diversas bibliotecas para implementação de algoritmos de machine learning. As secções deste relatório têm como objetivo apresentar e explicar o trabalho desenvolvido, destacando os datasets utilizados, as análises realizadas sobre eles e os modelos de machine learning criados. Será ainda fornecido um panorama geral das decisões tomadas pelo grupo ao longo do processo.

A metodologia adotada foi a CRISP-DM, escolhida por possibilitar a melhoria dos modelos em situações em que os resultados não são satisfatórios. No contexto desta metodologia, o primeiro passo consistiu em estudar o domínio do dataset e suas características. Em seguida, foi realizada a preparação mais adequada dos dados, visando remover *overfitting* e aprimorar o desempenho dos modelos. Ambos os conjuntos de dados foram divididos em conjuntos de treino e teste para o treinamento dos modelos. Finalmente, uma avaliação da qualidade dos resultados foi conduzida, reiniciando o processo se os melhores resultados não fossem alcançados. O trabalho foi considerado concluído quando o modelo atingiu um nível de desempenho satisfatório.

O conjunto de dados relacionado aos diabetes foi obtido por meio da plataforma Kaggle, onde também ocorreu a competição referente ao primeiro dataset. Após pesquisa de diversos conjuntos de dados para o trabalho, o nosso grupo selecionou o mencionado dataset devido ao interesse no tópico, possibilitando uma avaliação cuidada dos diferentes comportamentos e cuidados que implicam uma maior ou menor chance de desenvolver ou não diabetes. A escolha foi ainda motivada pela disponibilidade de informações detalhadas sobre as features que compõem o conjunto de dados.

2 Dataset Competição

2.1 Descrição do dataset

Na fase da competição foi nos dado 2 datasets de sobre a meteorologia e mais 2 de energia para treino então tivemos que juntar todos os datasets em um. Ao fazê-lo reparamos que num dataset de meteorologia estava incompleto, com este problema em frente decidimos que era melhor completa lo para termos um dataset mais robusto, este dados novos foram retirados de uma base de dados publica.

O dataset de energia possui 6 features e "alterar"linhas e o de meteorologia possui 15 features e "alterar"linhas, este dataset tem como target quantidade de energia elétrica injetada na rede elétrica (Injeção na rede (kWh)).

Esta fase ainda contem 1 datasets de treino que é a junção de uma dataset de energia e outro de meteorologia sem os resultados.

Features dos Datasets de Energia

- **Data**: o timestamp associado ao registo, ao dia;
- **Hora**: a hora associada ao registo;
- **Normal (kWh)**: quantidade de energia elétrica consumida, em kWh e proveniente da rede elétrica, num período considerado normal em ciclos bi-horário diários (horas fora de vazio);
- **Horário Económico (kWh)** : quantidade de energia elétrica consumida, em kWh e proveniente da rede elétrica, num período considerado económico em ciclos bi-horário diários (horas de vazio);
- **Auto consumo (kWh)**: quantidade de energia elétrica consumida, em kWh, proveniente dos painéis solares;

- **Injeção na rede (kWh)**: quantidade de energia elétrica injetada na rede elétrica, em kWh, proveniente dos painéis solares.

Features dos Datasets Meteorológicos

- **dt**: o timestamp associado ao registo;
- **dt_iso**: a data associada ao registo, ao segundo;
- **city_name**: o local em causa;
- **temp**: temperatura em ºC;
- **feels_like**: sensação térmica em ºC;
- **temp_min**: temperatura mínima sentida em ºC;
- **temp_max**: temperatura máxima sentida em ºC;
- **pressure**: pressão atmosférica sentida em atm;
- **sea_level**: pressão atmosférica sentida ao nível do mar em atm;
- **grnd_level**: pressão atmosférica sentida à altitude local em atm;
- **humidity**: humidade em percentagem;
- **wind_speed**: velocidade do vento em metros por segundo;
- **rain_1h**: valor médio de precipitação;
- **clouds_all**: nível de nebulosidade em percentagem;
- **weather_description**: avaliação qualitativa do estado do tempo.

2.2 Análise dos dados

Para realizar este trabalho, houve a necessidade de realizar um estudo geral do estado inicial dos dados, de modo a perceber o tratamento que seria necessário realizar para a utilização dos dados na criação de modelos.

A análise da correlação entre a energia gasta e as diferentes horas do dia oferece insights valiosos sobre os padrões de consumo ao longo do tempo. Ao explorar essa relação, podemos identificar potenciais influências das variações horárias nos níveis de consumo energético. Como podemos observar nas duas figuras abaixo, é notável que existe um maior injeção de energia entre 10 e as 14, e que entre as 19 e as 6 da manha a injeção é nula.

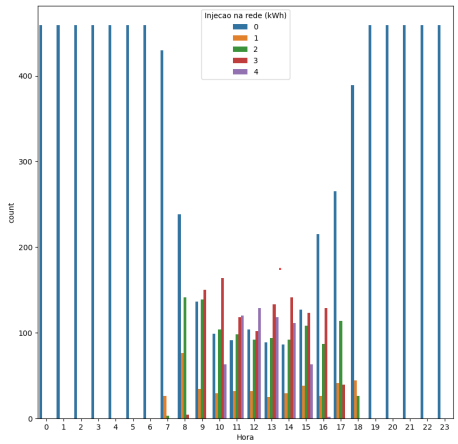

Figura 1: Grafico de barras dos vários tipos de injeção de rede por hora

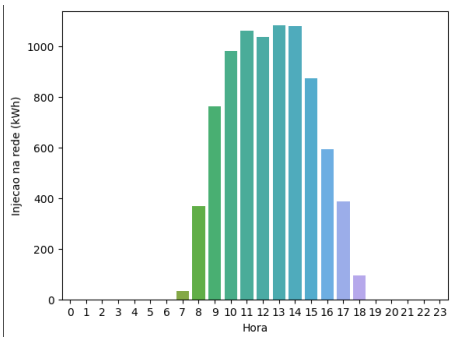

Figura 2: Gráfico de barras da injeção de rede(fez a transição de categórico para numérico) por hora

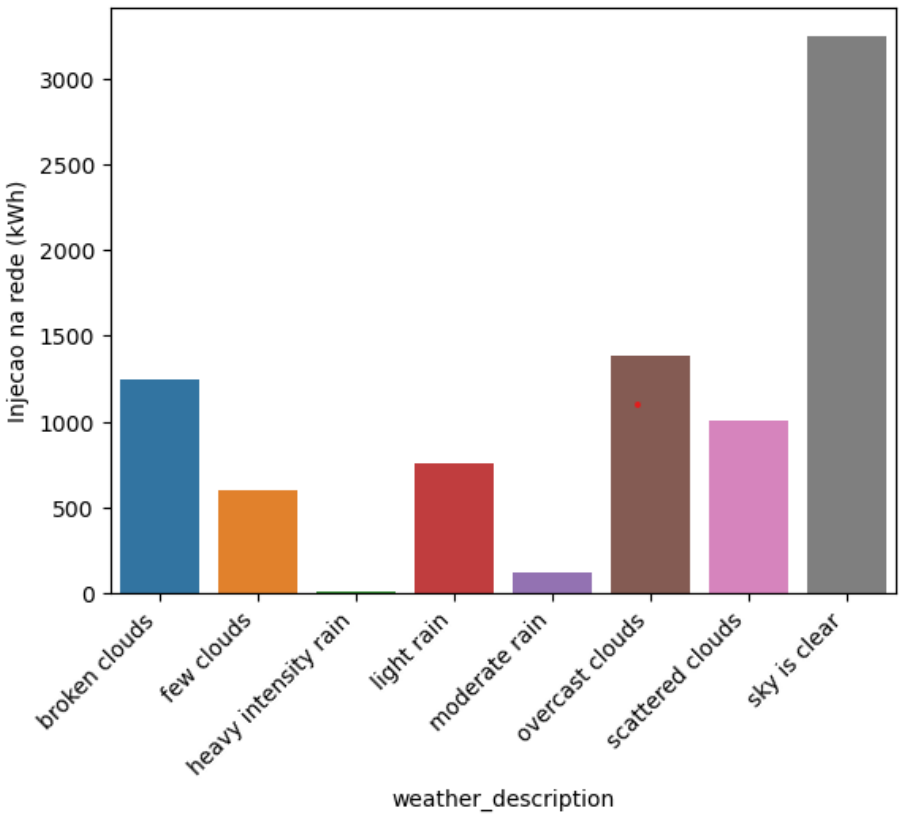

Figura 3: Gráfico de barras da injeção de rede(fez a transição de categórico para numérico) por descrição do tempo

O gráfico acima é bastante semelhante aos dois anteriores, só que aqui é analisada a correlação entre a "Injeção na rede"e "weather_description", onde é possível observar, como era esperado, uma melhor injeção de energia quando o céu está limpo e injeção de energia quase nula quando chove intensamente.

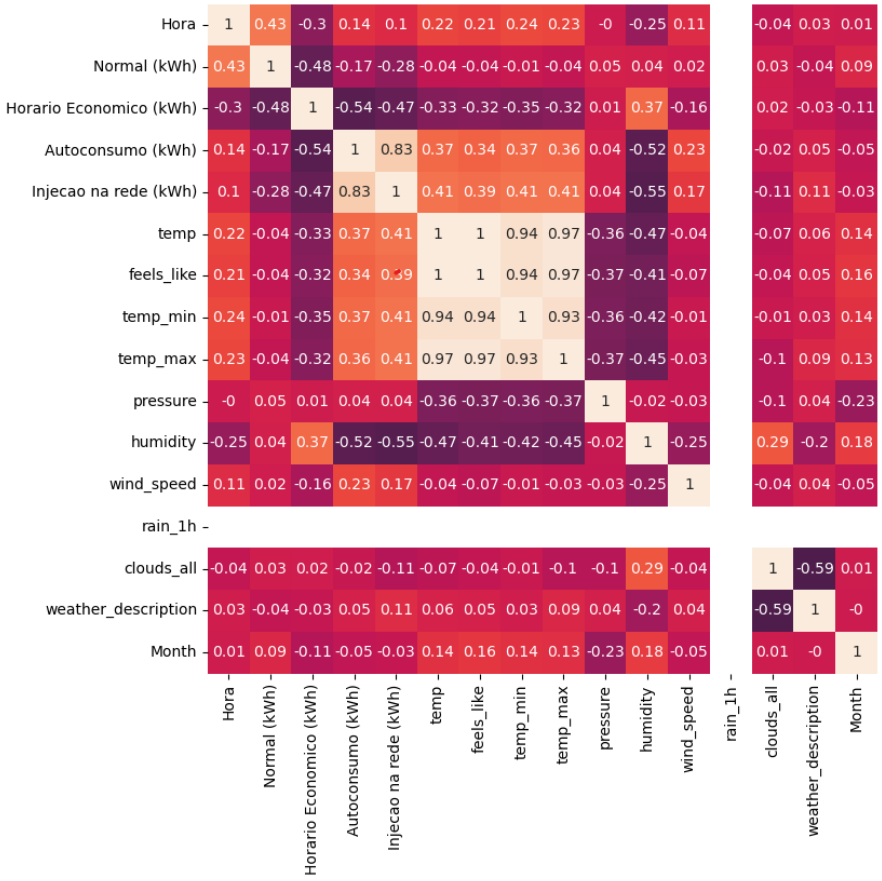

Figura 4: Matriz de correlação

Ao analisar uma matriz de correlação acima, é possível identificar padrões de relacionamento entre as variáveis, como por exemplo "Injeção de rede"e "Autoconsumo", as variáveis relacionadas com a temperatura também estão fortemente correlacionadas, e outras correlações notáveis como a "Hora"e "Injeção de rede", e "Humidity"e "Injeção de rede". Neste caso será útil analisar as variáveis mais correlacionadas com a variável "Injeção de rede"pois é esta variável que pretendemos prever.

A identificação de outliers é uma etapa crucial na análise de dados, destacando valores que se desviam significativamente do padrão geral do conjunto de dados. Esses pontos atípicos podem ter impactos significativos nas análises estatísticas e nos modelos de machine learning, distorcendo resultados e afetando a robustez das conclusões.

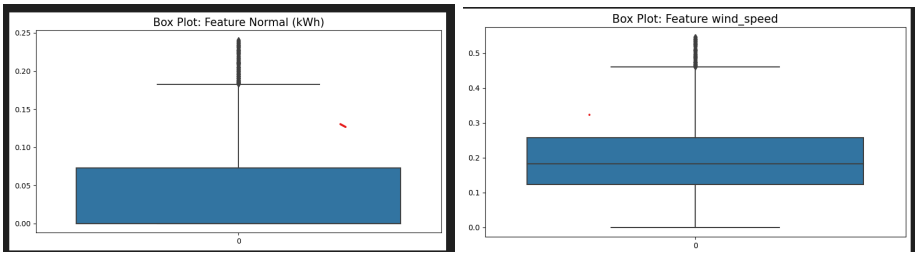

Figura 5: Gráfico de alguns outliers de duas features.

Acima podemos analisar dois gráficos Box Plot nos quais podemos identificar presença de outliers tanto no feature "Normal"como no feature"Wind_speed", onde os valores estão demasiado acima do normal.

2.3 Tratamento dos dados

No processo de preparação dos dados, realizamos uma série de etapas visando otimizar a qualidade e a utilidade das informações para as modelações subsequentes. As principais atividades desenvolvidas foram:

- **Transformação de Colunas Categóricas para Numéricas**

  Optamos por converter as colunas categóricas de 'Injeção na rede' e 'weather_description' em numéricas. Essa decisão foi fundamentada na necessidade de adequar a representação dos dados ao contexto analítico, permitindo uma interpretação mais precisa.

```
'Injecao na rede (kWh)': {
    'None': 0,
    'Low': 1,
    'Medium': 2,
    'High':3,
    'Very High':4
},
'weather_description': {
    'overcast clouds': 0,
    'scattered clouds': 1,
    'few clouds': 2,
    'sky is clear': 3,
    'broken clouds': 4,
    'light rain': 5,
    'moderate rain': 6,
    'heavy intensity rain':7
}
```

Figura 6: Json da configuração da transformação das Colunas Categóricas para Numéricas

- **Divisão de uma Coluna em Várias**

  No caso da 'Data', foi necessário desmembrar a coluna única que continha informações diversas em múltiplas colunas distintas como o 'Mes','Dia' e 'Ano'. Essa abordagem facilitou a manipulação e análise específica de cada componente, contribuindo para uma compreensão mais aprofundada dos dados.

- **Preenchimento de NaNs com 0**

  Adotamos a estratégia de preencher valores ausentes (NaNs) com zero no caso da precipitação por hora. Nos perante os NaNs dessa coluna tiramos a conclusão que quando não existe valor é porque não houve precipitação logo o '0'.

- **Eliminação de Colunas**

  Realizei a eliminação de colunas que não apresentavam relevância significativa para os objetivos da análise, visando simplificar o conjunto de dados e concentrar esforços nas variáveis mais pertinentes.Eliminamos 'city_name' pois era constante, o 'dt' pois o identificador da linha é desnecessario, e as colunas 'grnd_level','sea_level' pelo elvado numero de missing values.

- **Eliminação de Outliers**

Para garantir a robustez das análises estatísticas, identificamos os outliers que poderiam distorcer os resultados. Inicialmente eliminamos todos os outliers mas reparamos que a accuracy baixou quer com o nosso dataset de treino quer com o dataset de teste então como esta fase do trabalho é de competição achamos melhor deixar os outliers apesar que teoricamente não seja o mais correto, os modelos que tiveram uma pior accuracy deixamos estar essa eliminação pois achamos que teoricamente é mais correto e não interfere na competição.

Cada decisão tomada durante esse processo foi cuidadosamente avaliada com base no entendimento profundo do conjunto de dados e nos objetivos específicos da análise. O intuito foi assegurar que as transformações realizadas contribuíssem para aprimorar a qualidade dos dados, sem comprometer a integridade das informações.

2.4 Modelação

2.4.1 Árvores de Decisão e Random Forest

Para primeiro modelo de aprendizagem, utilizou-se um Decision Tree Classifier. Após treinamento e avaliação no conjunto de teste, a accuracy alcançada foi de aproximadamente 85,7%.

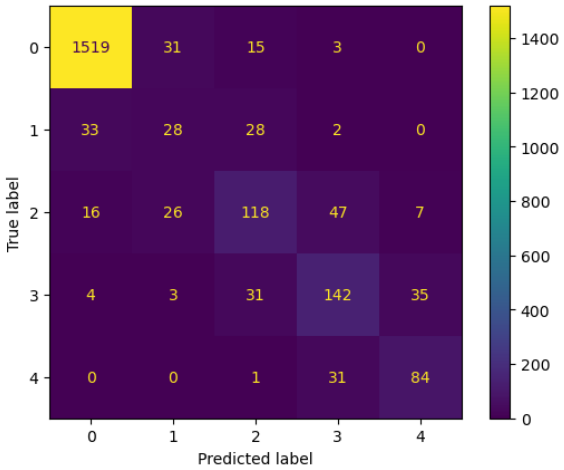

Figura 7: Confusion Matrix da Decision Tree

Em seguida, procurou-se melhorias no desempenho através de um modelo Random Forest Classifier. A otimização de hiperparâmetros resultou em um modelo aprimorado. Avaliando-o no conjunto de teste, observamos uma accuracy de cerca de 89,3%, indicando uma melhoria em relação ao modelo anterior.Posteriormente testamos com o dataset de teste no kaggle e deu-nos 89,6%

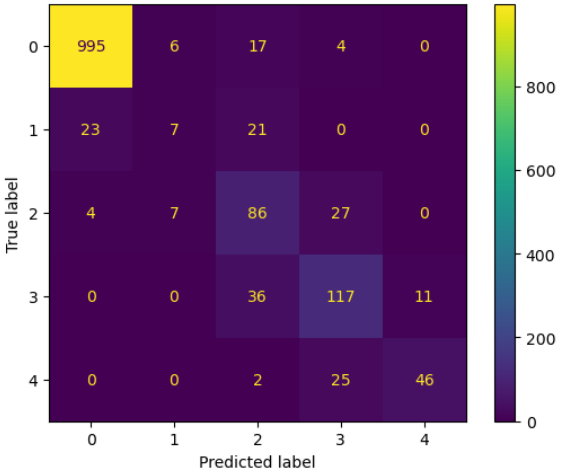

Figura 8: Confusion Matrix da Random Forest

Estes são os resultados obtidos pela Random Forest, sinalizando caminhos para ajustes e potenciais melhorias a serem exploradas.

2.4.2 Regressão Logística

Como segundo modelo de aprendizagem, empregamos três configurações diferentes para o Logistic Regression, nomeadamente newton-cg, lbfgs e liblinear. Cada configuração foi avaliada em termos de precisão, revocação e pontuação F1 no conjunto de teste.

Neste modelo o "newton-cg"foi o que acabou por ter uma boa precisão, nomeadamente 86%.

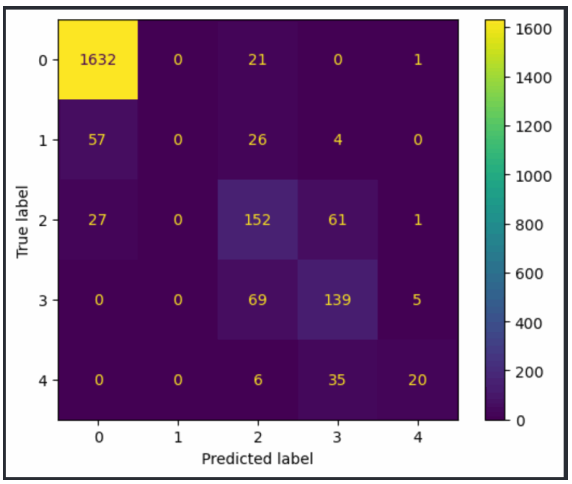

Figura 9: Confusion Matrix da Logistic Regression com o solver 'newton-cg'

Por fim, como o solver "Newton-cg"acabou por ter a maior precisão, entre os três solvers, foi o único que usamos no kaggle, apesar de atingir valores mais baixos neste último, ao nível dos 70%.

2.4.3 Redes Neuronais

Nesta etapa de construção do modelo, implementamos uma rede neural utilizando a biblioteca Keras para lidar com tarefas de classificação binária. A estrutura do modelo é composta por

camadas densas, e sua arquitetura é definida pelos parâmetros escolhidos, como a função de ativação, a taxa de aprendizado e o número de nós em cada camada.

Os dados foram submetidos a um processo de pré-processamento que envolveu o escalonamento dos recursos (MinMax Scaling) usando o MinMaxScaler da biblioteca scikit-learn. Esse processo garante que os dados estejam na faixa de 0 a 1, o que é importante para o treinamento eficiente da rede neural.

Os resultados finais revelaram uma precisão de aproximadamente 86.5% na fase de teste, acompanhada de métricas adicionais, como a matriz de confusão e pontuação F1. A análise dessas métricas proporciona uma compreensão abrangente do desempenho do modelo, especialmente em relação à sua capacidade de classificar instâncias positivas e negativas. Essas informações são essenciais para avaliar a eficácia da rede neural no contexto específico do projeto.

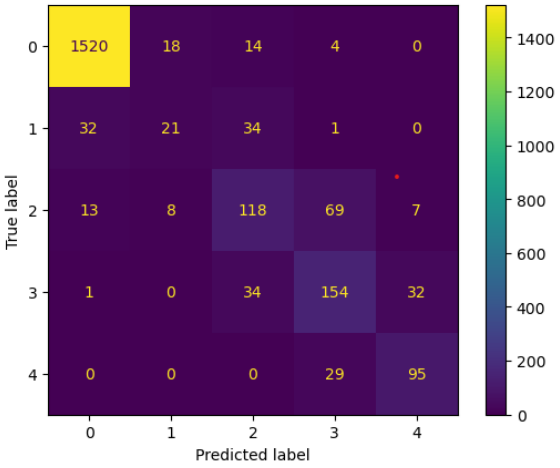

Figura 10: Confusion Matrix das Redes Neuronais

2.4.4 Support Vector Machine

Outro modelo de aprendizagem que resolvemos implementar foi o Support Vector Machine, que é um algoritmo que pretende encontrar um hiperplano ótimo para separar classes em um espaço dimensional, maximizando a margem entre elas.

Neste modelo usamos Cross Validation, nomeadamente k-fold, neste caso para k=10,para ajudar a fornecer uma estimativa mais estável do desempenho do modelo.

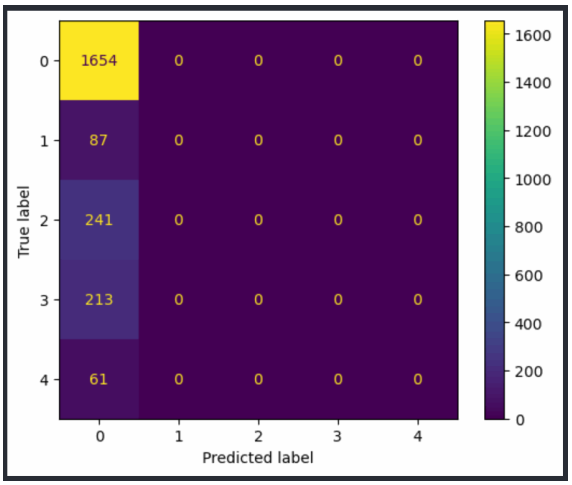

Figura 11: Confusion Matrix da Support Vector Machine'

Como resultado, acabamos por obter uma precisão de 86% no nosso dataset de treino, sendo que no Kaggle obtemos uma precisão de 84%.

2.4.5 XGBoost

Outra abordagem explorada foi a aplicação do algoritmo XGBoost Classifier, um modelo que foi escolhido pela sua versatilidade e eficácia para tarefas de classificação. Para otimizar a performance do modelo, realizamos procura aleatória de hiperparâmetros, para encontrar combinações eficientes num amplo espaço de possibilidades.

Para avaliarmos melhor o desempenho do modelo em diversas subdivisões dos dados de treino, empregamos a técnica de validação cruzada KFold.

Após a busca aleatória, identificamos a combinação mais eficaz de hiperparâmetros e procedemos ao treino do modelo otimizado. Ao aplicar essa versão do modelo ao conjunto de teste, conseguimos alcançar uma taxa de precisão de aproximadamente 89%.Ao analisar a eficácia do modelo, não nos limitamos à avaliação da precisão global. Também incorporamos diversas métricas, como o F1-score, o que nos permite obter uma compreensão mais abrangente do desempenho do modelo.

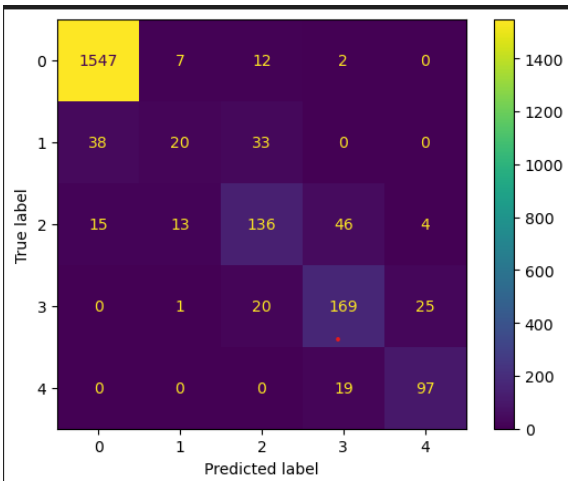

Figura 12: Confusion Matrix da XGboost

Em resumo, o XGBoost, após ser otimizado, apresentou um desempenho robusto e competitivo, destacando a sua elevada precisão neste dataset, que atingiu o maior valor entre todos os modelos.

2.5 Considerações Finais sobre a Tarefa de Decisão

Em resumo, conduzimos uma implementação abrangente utilizando diversos modelos, permitindo uma aprendizagem robusta e culminando em uma boa accuracy (89,6% ao utilizar o modelo XGBoost no dataset público e no privado 87,4%), que é um indicador crucial para avaliar o desempenho dos modelos. Notavelmente, optamos por uma abordagem mais conservadora na etapa de preparação de dados, uma vez que já havíamos alcançado uma boa precisão.

A decisão de minimizar as intervenções na preparação de dados foi tomada estrategicamente para evitar potenciais problemas de overfitting. Dada a qualidade alcançada nas métricas de avaliação, acredita-se que uma intervenção adicional na preparação dos dados poderia ter um impacto limitado ou até mesmo adverso no desempenho geral dos modelos.

Assim, a conclusão desta fase destaca não apenas o sucesso na obtenção de uma boa accuracy, mas também a tomada de decisões criteriosas em relação à preparação de dados, visando equilibrar a complexidade do modelo com a prevenção de possíveis sobreajustes. Este equilíbrio é fundamental para garantir que o modelo seja capaz de generalizar eficazmente para novos dados, consolidando, assim, sua utilidade e confiabilidade no contexto da tarefa de classificação binária em questão.

3 Dataset do Grupo

3.1 Descrição do dataset

O dataset escolhido para análise possui 70.682 linhas e é composto por 17 variáveis de características, além de uma variável alvo. Esta escolha foi feita a partir do Kagle, considerando sua relevância para compreender fatores de saúde e comportamento em uma amostra diversificada da população. **Features do Dataset:**

- **Age (Idade)**: Categorizada em 13 níveis, representando faixas etárias de 18 a 24 até 80 anos ou mais.
- **Sex (Gênero)**: Indicando o gênero do paciente (1 para masculino e 0 para feminino).
- **HighChol (Colesterol Alto)**: Binária (0 ou 1) indicando a presença de colesterol alto.
- **CholCheck (Verificação de Colesterol)**: Binária (0 ou 1) indicando se o paciente realizou verificação de colesterol nos últimos 5 anos.
- **BMI (Índice de Massa Corporal)**: Uma medida quantitativa do peso relativo à altura.
- fSmoker (Fumante): Binária (0 ou 1) indicando se o paciente fumou pelo menos 100 cigarros na vida.
- **HeartDiseaseorAttack (Doença Cardíaca ou Ataque Cardíaco)**: Binária (0 ou 1) indicando a presença de doença cardíaca ou ataque cardíaco.
- **PhysActivity (Atividade Física)**: Binária (0 ou 1) indicando se o paciente se envolveu em atividade física nos últimos 30 dias.
- **Fruits (Consumo de Frutas)**: Binária (0 ou 1) indicando se o paciente consome frutas uma ou mais vezes por dia.
- **Veggies (Consumo de Vegetais)**: Binária (0 ou 1) indicando se o paciente consome vegetais uma ou mais vezes por dia.

- **HvyAlcoholConsump (Consumo Elevaado de Álcool)**: Binária (0 ou 1) indicando consumo elevado de álcool.
- **GenHlth (Saúde Geral)**: Uma escala de 1 a 5, representando a autopercepção da saúde do paciente.
- **MentHlth (Saúde Mental)**: Número de dias de saúde mental prejudicada nos últimos 30 dias.
- **PhysHlth (Saúde Física)**: Número de dias de doença física nos últimos 30 dias.
- **DiffWalk (Dificuldade de Caminhar)**: Binária (0 ou 1) indicando se o paciente tem sérias dificuldades em caminhar ou subir escadas.
- **Stroke (Acidente Vascular Cerebral)**: Binária (0 ou 1) indicando se o paciente já teve um AVC.
- **HighBP (Pressão Alta)**: Binária (0 ou 1) indicando a presença de pressão alta.
- **Diabetes (Diabetes)**: Binária (0 ou 1) indicando a presença de diabetes.

Em todas as variáveis binárias, valor '0' indica a ausência da característica ou condição específica, enquanto o valor '1' indica a presença dessa característica ou condição.

3.2 Análise dos dados

Ao trabalhar com este conjunto de dados, a próxima etapa envolve a realização de uma análise inicial abrangente do dataset escolhido a qual será realizada no notebook *"data_exploration.ipynb"*. O objetivo é identificar e compreender o estado atual dos dados, visando determinar os tratamentos necessários para a preparação dos mesmos os, de modo a facilitar a construção de modelos de aprendizagem automática. Neste notebook, serão explorados aspetos como a distribuição de variáveis, a presença de *missing values*, estatísticas descritivas e visualizações pertinentes. Esta análise é fundamental para orientar as decisões sobre limpeza, transformação e seleção de características, contribuindo para o desenvolvimento eficaz de modelos de aprendizagem automática.

Inicialmente, na análise do dataset, apenas verificamos a contagem das pessoas que possuíam ou não a doença. E, de seguida, realizamos a identificação dos valores únicos de cada atributo, de maneira a verificar a presença de dados discrepantes (outliers) e entender a diversidade dos dados em cada variável. Trata-se de algo simples mas bastante importante na exploração inicial dos dados, fornecendo *insights* sobre a variabilidade e distribuição das informações contidas no conjunto de dados.

```
Age: [ 4. 12. 13. 11.  8.  1.  6.  3.  7. 10.  9.  5.  2.]
Sex: [1. 0.]
HighChol: [0. 1.]
CholCheck: [1. 0.]
BMI: [26. 28. 29. 18. 31. 32. 27. 24. 21. 58. 30. 20. 22. 38. 40. 25. 36. 47.
 19. 37. 41. 23. 34. 35. 42. 17. 33. 44. 15. 52. 69. 56. 45. 39. 92. 53.
 98. 50. 46. 79. 48. 16. 63. 72. 54. 49. 68. 43. 84. 73. 76. 55. 51. 75.
 57. 60. 12. 77. 82. 67. 71. 61. 14. 81. 59. 86. 13. 87. 65. 95. 89. 62.
 64. 66. 85. 70. 83. 80. 78. 74.]
Smoker: [0. 1.]
HeartDiseaseorAttack: [0. 1.]
PhysActivity: [1. 0.]
Fruits: [0. 1.]
Veggies: [1. 0.]
HvyAlcoholConsump: [0. 1.]
GenHlth: [3. 1. 2. 4. 5.]
MentHlth: [ 5.  0.  7.  3.  4.  2. 30. 20.  1. 15. 10. 25. 14. 28.  6. 29. 26. 12.
 16. 22. 13.  8.  9. 21. 18. 17. 27. 24. 23. 11. 19.]
PhysHlth: [30.  0. 10.  3.  6.  4. 15.  1.  2. 14.  7. 25. 21. 20.  5.  8. 22. 23.
 29. 12. 18. 28. 26. 24. 27. 11. 13. 16. 17.  9. 19.]
DiffWalk: [0. 1.]
Stroke: [0. 1.]
HighBP: [1. 0.]
Diabetes: [0. 1.]
```

Figura 13: Identificação dos valores únicos de cada atributo.

Após efetuadas estas pequenas consultas iniciais, procedemos para a análise da distribuição de cada atributo no dataset, recorrendo a **pie charts**, tal como se vê nas figuras seguintes.

Esta abordagem gráfica inicial oferece insights imediatos sobre a distribuição de cada variável no conjunto de dados, tal como mencionado anteriormente, preparando o terreno para análises mais aprofundadas.

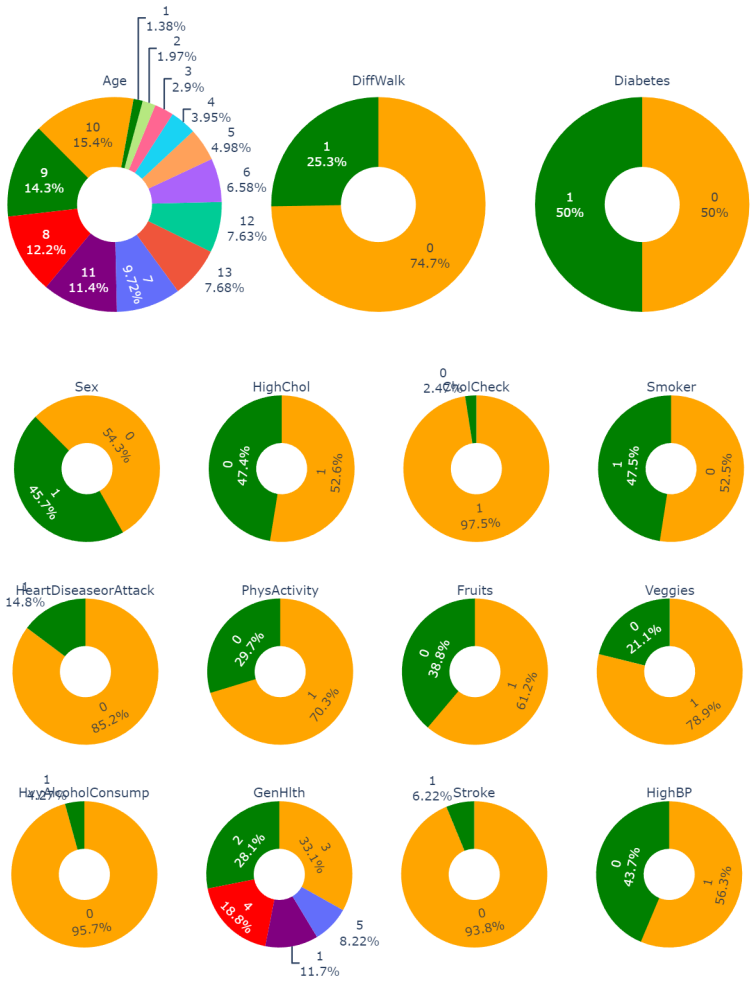

Figura 14: Distribuição dos diferentes atributos no dataset.

De seguida, decidimos analisar os atributos que se encontravam distribuídos por mais valores, os quais identificamos no começo da nossa análise, em histogramas. Desta forma foi possível identificar padrões e tendências nos dados.

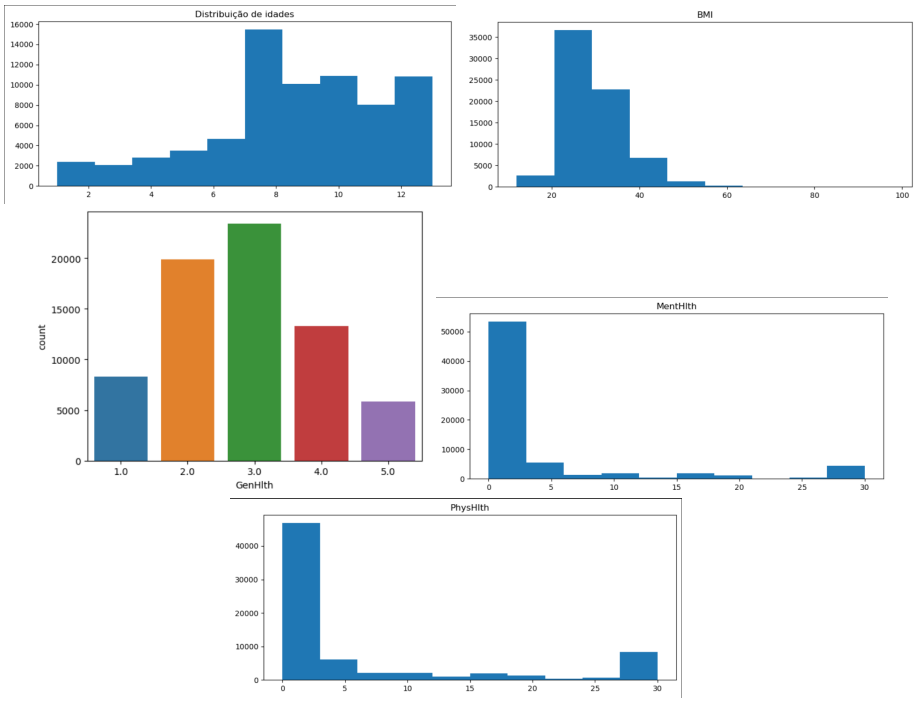

Figura 15: Gráficos representativos dos atributos não binários do dataset.

Decidimos ainda verificar a presença de *outliers* recorrendo ao uso de boxplots. O atributo que nos pareceu que poderia ter mais problemas neste sentido foi o **BMI** uma vez que na análise dos valores únicos este possuía uma grande variedade de valores. Assim, como se comprova na figura seguinte, de facto o atributo **BMI** possui bastantes outliers. A presença destes outliers em um conjunto de dados pode distorcer medidas estatísticas, como a média, e afetar a interpretação dos resultados, comprometendo a precisão das análises e a validade dos modelos estatísticos, sendo portanto importante a sua identificação.

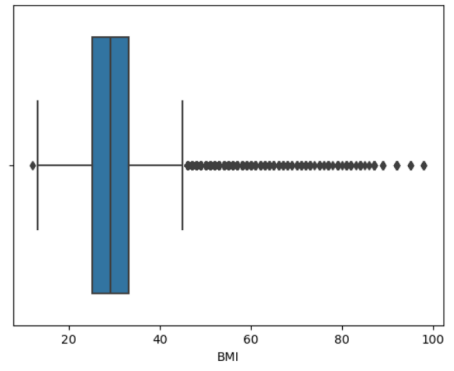

Figura 16: Boxplot relativo ao atributo BMI.

Por fim decidimos verificar a ligação presente entre os atributos binários do dataset e a presença ou não de diabetes. Utilizamos para tal vários *pie charts* onde comparamos os diferentes os atributos com diabetes e sem diabetes, sendo possível estabelecer uma ligação entre estes e o surgimento/presença da doença. Em seguida mostramos esses gráficos:

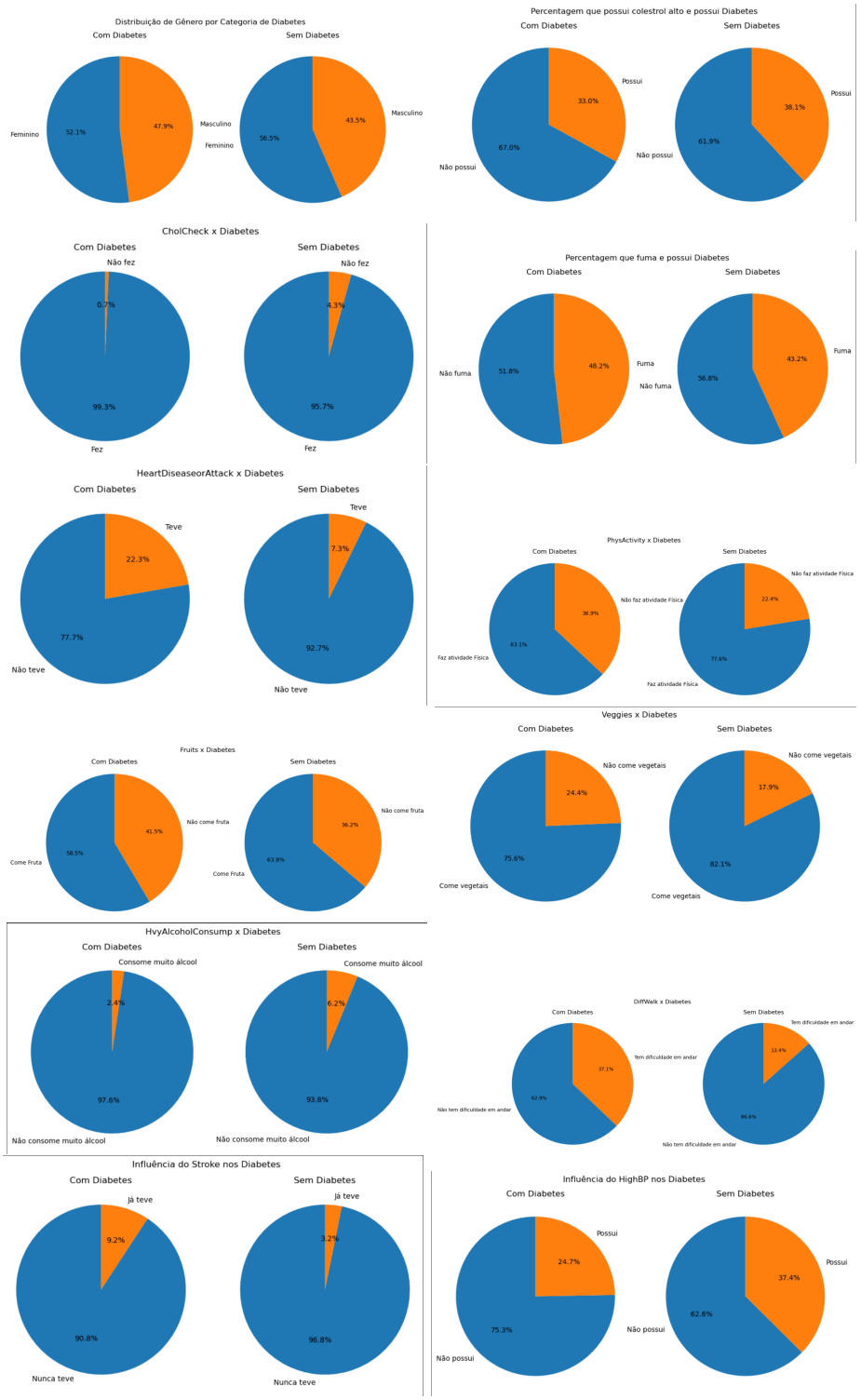

Figura 17: Análise a correlação dos atributos com os Diabetes.

Por último acrescentamos só mais um *pie chart* para verificar a distribuição da doença em estudo neste dataset e verificamos, tal como se comprova na imagem seguinte, de que existe igual número de pessoas com diabetes e sem neste dataset.

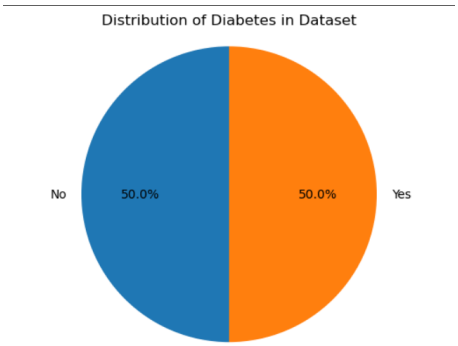

Figura 18: Distribuição de Diabetes pelo dataset.

3.3 Tratamento de dados

A partir da análise dos dados realizada anteriormente, o grupo decidiu, primeiramente, realizar a remoção de linhas duplicadas, de seguida, passou a realizar-se a transformação de algumas variáveis. As variáveis selecionadas para transformação foram :

- 'BMI' (Índice de Massa Corporal), em que se alterou os seus valores para variáveis discretas de 1 a 5, tendo em conta os intervalos fornecidos pela Organização Mundial da Saúde;
- 'MentHlth' (Dias de Saúde Mental Prejudicada), onde se alterou os seus valores para variáveis discretas de 1 a 5, onde 1 representa até 6 dias de saúde mental prejudicada, 2 representa 6 a 11 dias, 3 representa 12 a 17 dias, 4 representa 18 a 23 dias e 5 representa 24 a 30 dias;
- 'PhysHlth' (Dias de Doença Física), onde se alterou os seus valores para variáveis discretas de 1 a 5, onde 1 representa até 6 dias de saúde mental prejudicada, 2 representa 6 a 11 dias, 3 representa 12 a 17 dias, 4 representa 18 a 23 dias e 5 representa 24 a 30 dias;

Posteriormente, para garantir consistência e evitar desproporcionalidades entre as variáveis, aplicou-se a normalização de todos os dados utilizando a técnica de Min-Max Scaling.

Depois, realizou-se a análise de correlação, que revelou alguns padrões no conjunto de dados. Esses padrões fornecem insights iniciais sobre possíveis associações no conjunto de dados. Contudo, neste dataset o grupo não encontrou nenhuma correlação superior a 0.6, nem nenhuma com um valor negativo alto, o que indica que as features não se encontram fortemente correlacionadas. Este resultado sugere que as features do dataset não estão fortemente correlacionadas entre si, sendo esta observação é crucial para entender a independência e a diversidade das características presentes nos dados. Pelo que, não se realizou nenhuma alteração nos dados do dataset a partir desta análise.

De seguida, realizou-se a divisão do conjunto original em treinamento e teste, com 30% destinados ao conjunto de teste. Observou-se as contagens das classes da variável alvo 'Diabetes' nos conjuntos de treinamento e teste, constatando inicialmente um desbalanceamento.

Para lidar com esse desbalanceamento, realizamos um upsampling da classe minoritária ('Diabetes' == 0) no conjunto de treinamento. O objetivo foi equilibrar as classes, aumentando o número de instâncias da classe minoritária para 25.000.

Os conjuntos de treino foram, então, reconstruidos, combinando a classe maioritária original com a classe minoritária upsampled. Essa abordagem visa melhorar a capacidade do modelo de aprender padrões em ambas as classes.

Por fim, para facilitar análises futuras e a aplicação de modelos preditivos, exportamos os conjuntos de treinamento ("X_train", "y_train") e teste ("X_test", "y_test"). Além disso, o conjunto "pf"também foi exportado para referência, todos em formato CSV.

3.4 Modelação

3.4.1 Árvore de Decisão e Random Forest

Para primeiro modelo de aprendizado, utilizou-se um Decision Tree Classifier. Após treinamento e avaliação no conjunto de teste, a previsão alcançada foi de aproximadamente 66%.

Em seguida, procurou-se melhorias no desempenho através de um modelo Random Forest Classifier. A otimização de hiperparâmetros resultou em um modelo aprimorado. Avaliando-o no conjunto de teste, observamos uma accuracy de cerca de 72%, indicando uma melhoria em relação ao modelo anterior.

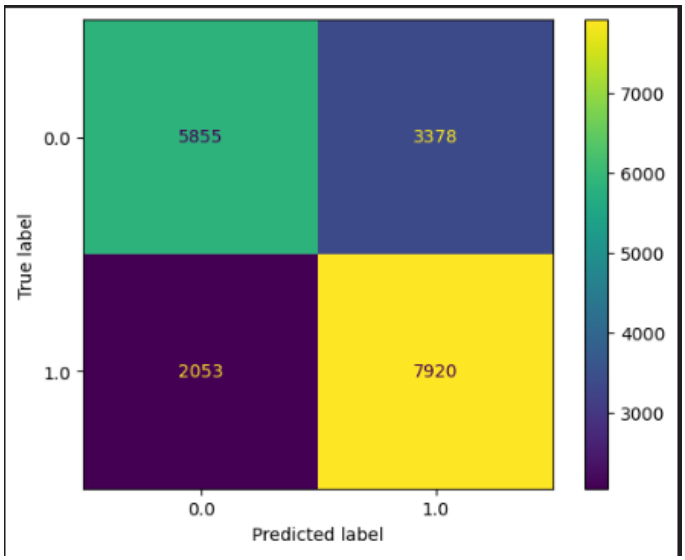

Figura 19: Confusion Matrix da Random Forest

Estes são os resultados obtidos pela Random Forest, sinalizando caminhos para ajustes e potenciais melhorias a serem exploradas.

3.4.2 Regressão Logística

Para segundo modelo de aprendizagem, empregamos três configurações diferentes para o Logistic Regression. Cada configuração foi avaliada em termos de precisão, revocação e pontuação F1 no conjunto de teste.

O Logistic Regression com o solver 'newton-cg' levou aproximadamente 0.69 segundos para treinar e alcançou uma accuracy de 74%.

Já o Logistic Regression com o solver 'lbfgs' teve um tempo de treinamento de cerca de 0.23 segundos, também resultando em uma accuracy de 74%.

Por fim, o Logistic Regression com o solver 'liblinear' teve um tempo de treinamento de aproximadamente 0.44 segundos, e a accuray foi novamente de 74%.

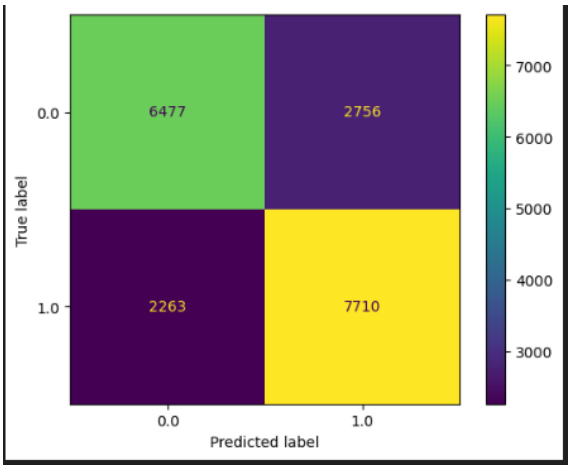

Figura 20: Confusion Matrix da Logistic Regression para os 3 solvers(ConFusion Matrix igual com os 3 solvers

Esses resultados destacam que, apesar da variação nos solvers, não houve diferenças significativas no desempenho do modelo. Essa análise fornece insights sobre o comportamento do Logistic Regression com diferentes configurações, orientando futuros ajustes no modelo.

3.4.3 Support Vector Machine

Decidimos adicionar também ao nosso projeto o método de Support Vector Machine (SVM) para tarefas de machine learning. Inicialmente, executamos uma validação cruzada de 10-fold para avaliar a robustez do modelo. Os resultados indicaram uma precisão média de 74%, com um desvio padrão de 0.01.

Posteriormente, foi utilizada a técnica de *Hold Out*, onde se treinou o modelo SVM com um conjunto de treino e avaliando-o com um conjunto de teste separado. Esta abordagem resultou em uma precisão de 74%. A matriz de confusão gerada para esta etapa forneceu insights sobre o desempenho do modelo em termos de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos.

Além disso, exploramos ainda a otimização de parâmetros por meio da técnica de *Grid Search*, que envolveu uma busca sistemática em um espaço pré-definido de hiperparâmetros. Novamente, a matriz de confusão foi utilizada para avaliar o desempenho do modelo com os parâmetros otimizados.

As duas matrizes de confusão adicionais, provenientes das etapas de *Hold Out* e *Grid search*, foram geradas para fornecer uma análise mais aprofundada da capacidade do modelo SVM em classificar corretamente as instâncias positivas e negativas. Essas matrizes de confusão oferecem uma visão mais detalhada do desempenho do modelo em diferentes cenários, contribuindo para a compreensão global da eficácia do SVM no contexto específico do seu projeto. Mostramos em seguida as duas matrizes geradas:

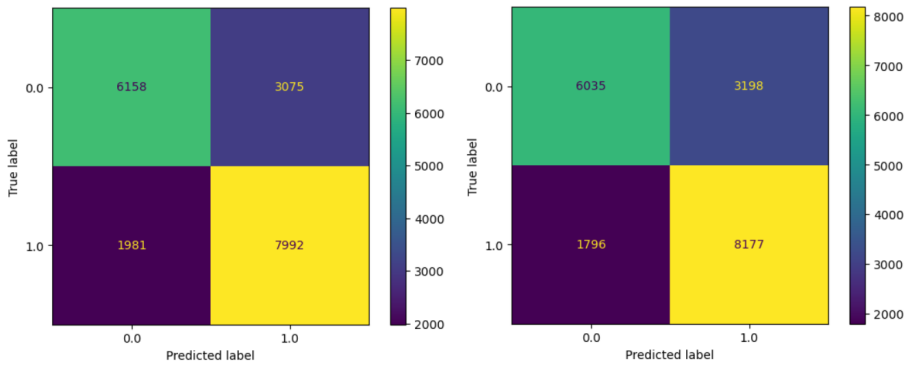

Figura 21: Matrizes de confusão geradas nas fases de Hold Out e Grid Search, respetivamente.

3.4.4 Redes Neuronais

Nesta fase de modelação, realizamos também a implementação de um modelo de rede neural utilizando Keras para tarefas de classificação binária. O modelo consiste em camadas densas com uma arquitetura específica definida pelos parâmetros, como função de ativação, taxa de aprendizado e número de nós por camada. O treinamento do modelo foi conduzido utilizando a técnica de validação cruzada K-Fold com 5 folds.

Foi realizada uma procura de hiperparâmetros utilizando Grid Search, focando na otimização da função de ativação, taxa de aprendizado e configuração de nós. O melhor conjunto de hiperparâmetros foi identificado, e o modelo foi treinado com esses valores otimizados. Durante o treino do modelo, foi utilizado um conjunto de validação para monitorizar o desempenho do modelo ao longo das épocas.

Os resultados finais mostraram uma precisão de aproximadamente 69.7% na fase de teste, acompanhada de métricas adicionais, como matriz de confusão, pontuação F1 e recall. A análise dessas métricas fornece uma compreensão abrangente do desempenho do modelo em termos da sua capacidade de classificar instâncias positivas e negativas. Estas informações são cruciais para avaliar a eficácia do modelo de rede neural no contexto específico do projeto.

3.4.5 XGboost

No contexto do desenvolvimento de modelos de aprendizagem, a última abordagem explorada foi a aplicação do algoritmo XGBoost Classifier, um modelo extremamente versátil e eficaz para tarefas de classificação. Para otimizar a performance do modelo, realizamos procura aleatória de hiperparâmetros, uma prática comum para encontrar combinações eficientes em um amplo espaço de possibilidades.

Os parâmetros ajustáveis incluíram taxas de aprendizagem, profundidade máxima da árvore, número de estimadores e outras características específicas do algoritmo. A validação cruzada, utilizando KFold, foi utilizada para avaliar o desempenho do modelo em diferentes subconjuntos dos dados de treino.

Após a procura aleatória, identificamos a melhor combinação de hiperparâmetros e treinamos o modelo otimizado. Ao aplicar esse modelo ao conjunto de teste, alcançamos uma accuracy em torno de 73%. Esse resultado sugere que o XGBoost pode ser uma ferramenta valiosa para tarefas de classificação neste contexto específico.

Ao avaliar a aplicação do modelo, não apenas observamos a accuracy global, mas também consideramos diversas métricas, como o F1-score. Essas métricas proporcionam uma compreensão mais abrangente do desempenho do modelo, especialmente em casos de desbalanceamento entre as classes.

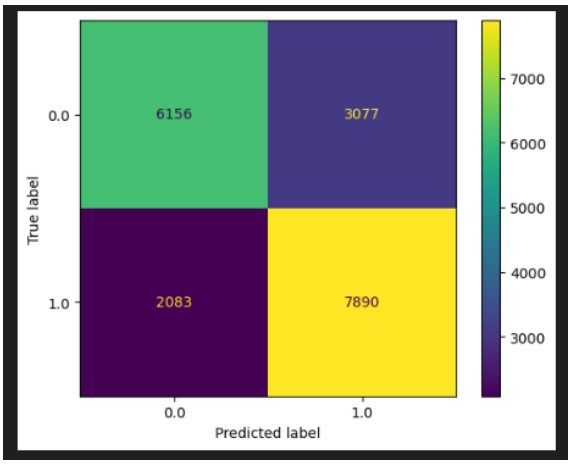

Figura 22: Confusion Matrix da XGboost

Em suma, o XGBoost, após otimização, demonstrou uma performance sólida e competitiva, indicando seu potencial para contribuir significativamente em tarefas de previsão relacionadas ao conjunto de dados em questão.

4 Conclusões sobre o trabalho realizado

Ao longo da realização deste projeto, encontrou-se algumas adversidades, desde a escolha do dataset para a tarefa Dataset Grupo, até ao desenvolvimento dos modelos. Contudo, o grupo considera que atingiu os objetivos estipulados para este projeto, criando diversos modelos com boa eficácia. Aqueles que não obtiveram alta eficácia possuem uma justificação plausível, evidenciando uma abordagem analítica.

Por fim, a realização deste projeto mostrou-se desafiante e permitiu consolidar diversos conteúdos abordados nas aulas de Dados e Aprendizagem Automática. Salientando ainda, que o conteúdo lecionado nas aulas práticas foi fundamental e serviu como alicerce para a maioria dos modelos de aprendizagem desenvolvidos. Isso destaca a importância prática das aulas, proporcionando uma base sólida para a implementação eficaz dos conceitos teóricos aprendidos nas aulas.