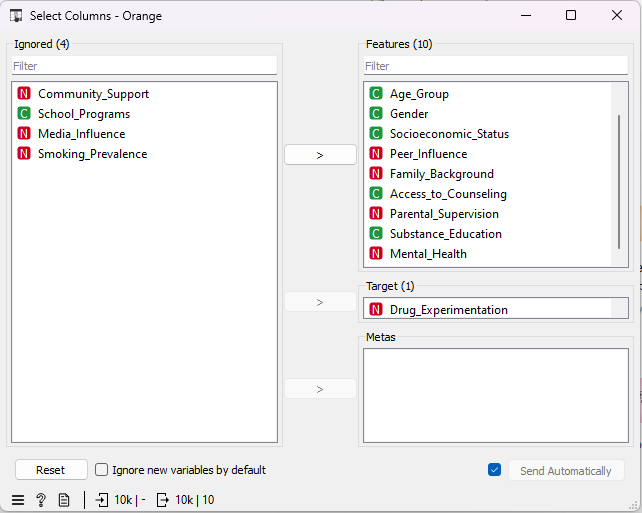
**Template para entrega do projeto da disciplina**

**Ciência de Dados e Inteligência Artificial**

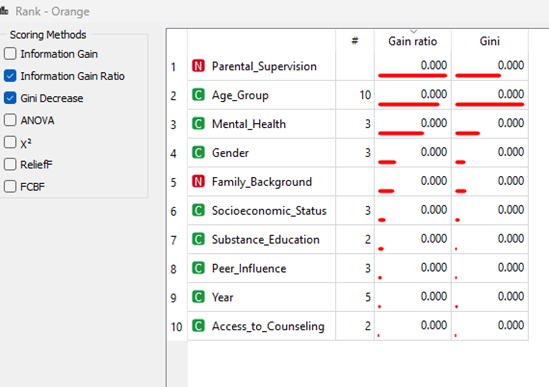
**Fase 2**

|  |  |
| --- | --- |
| **Nome do estudante** | **Nicole Saenger** |



|  |  |
| --- | --- |
| Exploração dos dados | |
| *Que tipo de experimentos você fez na exploração dos dados (verificação de outliers, cálculos de médias, dados inválidos etc.).* | |
| Em uma primeira análise, a fim de iniciar a etapa de entendimento e exploração dos dados, realizei a importação do arquivo CSV ‘youth\_smoking\_drug\_data\_10000\_rows\_expanded’ e adicionei o widget ‘Data Info’ a fim de verificar os detalhes do arquivo escolhido, visualizando o número total de linhas e colunas com o tipo de cada uma delas. Já nesta etapa, foi possível validar se havia dados faltantes, e pude constatar que o dataset já estava completo.    Em seguida, adicionei a ferramenta ‘Select Columns’ a fim de remover aquelas que eu identifiquei previamente que não fariam sentido a minha análise, mantendo somente 11 colunas, sendo 5 delas categóricas, 5 numéricas e uma delas atribui a coluna alvo (target).  Realizada esta etapa, adicionei um ‘Data Table’ para conferir se as colunas que eu havia removido não constavam mais, e interliguei com o widget ‘Edit Domain’ a fim de alterar o tipo de algumas colunas. Inicialmente, havia decidido realizar a reinterpretação apenas da coluna ‘year’, passando os valores numéricos para categorias de 5 anos, englobando 2020, 2021, 2022, 2023 e 2024. Posteriormente, após constatar necessidades dos algoritmos, retomei a ferramenta para realizar demais ajustes.  A fim de obter uma visualização mais estratégia em relação aos dados, adicionei cores e mais 3 ferramentas de visualização, ‘Box Plot’, ‘Feature Statistics’ e ‘Distribuitions’. Com o ‘Box Plot’, pude obter uma visualização mais estatística, comparando as categorias e verificado a existência de possíveis outliers.  Já com o ‘Feature Statistics’, pude obter uma visão geral em relação a todas as colunas do dataset, analisando métricas como média, moda, valores mínimos e máximas, taxa de dispersão e distribuição dos dados de maneira visual de acordo com as cores pré-estabelecidas referente a cada coluna.    Por fim, com o intuito de validar a distribuição dos dados e me certificar de que eles estavam balanceados, fiz uma análise utilizando o widget ‘Distribuition’, verificando através dos histogramas a frequência de cada um dos valores.    Após todas estas validações, constatei que o dataset possuía todos os valores devidamente preenchidos e de uma maneira muito bem distribuída, sem a presença de valores inesperados, dados extremos ou desbalanceados. | |
|  | |
| Escolha de, ao menos, três algoritmos de aprendizado para a modelagem |
| *Apresente os algoritmos utilizados e justifique a escolha.* |
| Após inúmeros testes e análises, escolhi utilizar os 3 algoritmos que me trouxeram melhores resultados: ‘kNN’, ‘Tree’ e ‘Random Forest’.  Em uma primeira análise, escolhi o ‘kNN’ visto que é um algoritmo simples e de aprendizado supervisionado, que faz previsões com base no parâmetro de vizinhos mais próximo que indicamos. Para realizar a previsão, ele mede a distância entre um ponto de teste e os demais, buscando os ‘k’ vizinhos mais próximos deste ponto e classificando-o com base na média dos valores vizinhos.  Como segundo algoritmo, dei prioridade para algum em que eu pudesse analisar de que maneira foi atingido o resultado. Levando em conta este fator, escolhi o ‘Tree’, visto que ele é considerado um algoritmo de ‘caixa aberta’, já que podemos analisar como foi feita a montagem da árvore, podendo compreender a relação entre as variáveis e a coluna alvo. O seu aprendizado baseia-se em dividir os dados em grupos resultantes de condições lógicas ou perguntas, construindo uma árvore hierárquica no qual cada nó representa uma condição.  Por fim, adicionei o ‘Random Forest’, que nada mais é do que inúmeras árvores de decisão, sendo que cada uma delas é treinada com um subconjunto diferente de dados, levando a diferentes respostas. A sua previsão é feita pela média destes produtos obtidos, trazendo um resultado mais preciso e funcionando bem para um conjunto de dados maior. Priorizei adicioná-lo a minha análise, pois como é um resultado de várias combinações, reduz o ‘overfitting’ que ocasionalmente pode ocorrer com a utilização de uma única árvore. |

|  |
| --- |
| Preparação dos dados de acordo com as características dos algoritmos de aprendizado escolhidos |
| *Descreva o processo realizado para essa etapa.* |
| Tendo em vista que na grande maioria dos algoritmos a ausência de dados, a má distribuição, outliers e valores fora de escala afetam negativamente o processo de aprendizado, retomei a analisar o dataset a fim de ajustar estes pontos.  Como no processo de exploração e análise dos dados eu já havia verificado a distribuição e integridade dos dados, não precisei retomar esta etapa. Contudo, precisei ajustar a escala de algumas colunas, padronizando-as para valores de [0, 1] utilizando o widget ‘Preprocess’ e escolhendo a opção ‘Normalize Features’.    Além de padronizar a escala das colunas numéricas, retomei ao widget ‘Edit Domain’ e transformei as colunas ‘Peer\_Influence’, ‘Mental\_Health’ e ‘Drug\_Experimentation’ em categóricas.  Em relação as colunas ‘Peer\_Influence’ e ‘Mental\_Health’, defini que os valores de 1 a 3 seriam classificados em escala como ‘Low’, valores de 4 a 6 como ‘Middle’ e o restante como ‘High’. Já no que se refere a coluna alvo ‘Drug\_Experimentation’, transformei-a para uma coluna categórica de ‘No’ para valores de 10 a 30, e ‘Yes’ para os demais, referindo-se à probabilidade do uso de substâncias.  Decidi por realizar estas transformações a fim de obter resultados mais precisos e pelo fato de conceder valores mais bem padronizados e compatíveis com as necessidades de cada um dos algoritmos. |



|  |
| --- |
| Execução dos experimentos de aprendizado e coleta das métricas |
| *Descreva o processo e os resultados obtidos.* |
| Na etapa de execução dos algoritmos, iniciei adicionando o widget ‘Rank’ ao ‘Data Table’ após o reprocessamento dos dados, com o intuito de averiguar as colunas mais relevantes e associadas à minha coluna alvo ‘Drug\_Experimentation’.  A partir deste resultado, selecionei a ferramenta ‘Select Columns’ e filtrei pelas colunas provenientes do resultado do ‘Rank’, a fim de melhorar a precisão dos modelos, reduzir o *overfitting*, melhorar a interpretação dos resultados e construir modelos mais rápidos. As colunas mantidas foram ‘Parental\_Supervision’, ‘Mental\_Health’, ‘Peer\_Influence’, ‘Age\_Group’, ‘Gender’, ‘Family\_Background’ e a coluna alvo ‘Drug\_Experimentation’.      Realizada a seleção das colunas, liguei ao novo ‘Data Table’ e selecionei novamente o ‘Feature Statistics’ e o ‘Box Plot’ com a intenção de revisitar os índices do dataset após as alterações feitas.  Com os dados ajustados e prontos para seguir com o processo, conectei o ‘Data Table’ ao widget de ‘Test and Score’, atribuindo uma porcentagem de dados para treino e outra para teste. Em seguida, ajustei os parâmetros, definindo o ‘Number of folds’ como 10, selecionando a opção ‘Stratified’ e ‘Test on test data’.    Posteriormente, acrescentei os 3 algoritmos escolhidos e liguei-os com o widget de ‘Test and Score’ e ao ‘Data Table’. O único que liguei somente ao ‘Test and Score’, foi o algoritmo ‘kNN’, visto que ele não requer um treinamento explícito.    Dando seguimento, iniciei os testes com os algoritmos, manipulando os parâmetros de cada um a fim de verificar quais me dariam melhores previsões. Como resultado, para o algoritmo de ‘**Random Forest**’ estabeleci 38 como o número de árvores, considerando 2 atributos para cada nó. Além disso, estabeleci como limite de profundidade máxima das árvores 14 níveis e o tamanho mínimo de um subconjunto como 8.    Já para o algoritmo ‘**Tree’**, selecionei a opção da árvore binária, ou seja, criar duas divisões em cada nó, defini o número 2 como o mínimo de instâncias que o nó de uma folha pode conter, estabeleci o tamanho mínimo de um subconjunto de dados para a divisão em um nó como sendo 4 e limitei a profundidade da árvore a 14 níveis também. Ademais, classifiquei o limite em percentual das instâncias a 95%.    Por fim, em relação aos parâmetros do ‘**kNN’**, defini 20 como o número de vizinhos (k) que o modelo iria considerar, como métrica escolhi a ‘Chebyshev’, que acredita que a maior diferença em uma única característica defini o resultado em relação aos vizinhos, e como parâmetro ‘Weight’, defini que os vizinhos mais próximos têm maior influência em comparação com os mais distantes.    Levando em conta os parâmetros estabelecidos, no widget ‘Test and Score’ obtive como resultado as seguintes métricas:    Antes de indicar qual dos 3 algoritmos melhores se destacou, vamos analisar cada uma das estatísticas e compará-las entre os 3 modelos.  Iniciando pelo índice de área sob a curva ROC (AUC), para os 3 casos obtivemos resultados bons, sendo que o ‘kNN’ chegou muito próximo do resultado máximo! Este índice compara a taxa de verdadeiros positivo scom a taxa de falsos positivos. Para os demais algoritmos, também obtivemos resultados promissores, mas um pouco menores em comparação com o kNN.   |  |  | | --- | --- | | Modelo | AUC | | kNN | 0.988 | | Random Forest | 0.876 | | Tree | 0.889 |   Analisando a taxa de ‘Classification Accuracy’ (CA), seguimos a mesma posição, com o ‘kNN’ trazendo o melhor resultado, chegando novamente em um índice próximo ao máximo. Já em relação ao ‘Random Forest’ e ao ‘Tree’, obtiveram resultados parecidos e medianos, porém consideravelmente bons.   |  |  | | --- | --- | | Modelo | CA | | kNN | 0.929 | | Random Forest | 0.735 | | Tree | 0.797 |   Quanto ao ‘F1 Score’, tivemos uma maior diferença entre os três algoritmos, sendo que, mais uma vez, o que melhor se destacou foi o ‘kNN’. Nesta situação, o que menos se destacou foi o ‘Random Forest’, apresentando um índice de 0.673. Através desta métrica, podemos observar a média entre os valores de ‘Precision’ e ‘Recall’.   |  |  | | --- | --- | | Modelo | F1 | | kNN | 0.930 | | Random Forest | 0.673 | | Tree | 0.798 |   Com o índice de Precision (Prec), também obtivemos bons resultados de uma maneira geral, medindo a proporção de instâncias classificadas como positivas verdadeiras.   |  |  | | --- | --- | | Modelo | Prec | | kNN | 0.937 | | Random Forest | 0.784 | | Tree | 0.799 |   Para a taxa de ‘Recall’, também conhecida como sensibilidade, mais uma vez obtivemos resultados positivos, com o ‘kNN’ trazendo melhores resultados e os demais resultados um pouco inferiores. Este índice calcula a proporção de instâncias positivas que foram identificadas corretamente.   |  |  | | --- | --- | | Modelo | Recall | | kNN | 0.929 | | Random Forest | 0.735 | | Tree | 0.797 |   Por último, possuímos o índice de MCC, que é uma métrica referente a todas as categorias de Positivos falsos e verdadeiros e de Negativos falsos e verdadeiros. Dentre as demais taxas analisadas anteriormente, esta é a que teve maior discrepância entre os modelos. Novamente, ‘kNN’ apresentou o índice mais alto entre os demais, porém inferior em comparação com as suas métricas anteriores. Já o algoritmo ‘Random Forest’ trouxe um resultado ruim, indicando 0.360. E como resultado mediano, o ‘Tree’ apresentou como taxa de 0.546.   |  |  | | --- | --- | | Modelo | MCC | | kNN | 0.852 | | Random Forest | 0.360 | | Tree | 0.546 | |

Logo, com base nos índices analisados, o que melhor se comportou em relação as métricas foi o algoritmo ‘kNN’, que manteve ótimos resultados para cada uma das relações. Quanto aos demais, também tivemos resultados positivos, porém com algumas variações. Dentre os dois, o algoritmo ‘Tree’ se saiu um pouco melhor, principalmente em relação as métricas de F1 e MCC, mas em comparação com os demais índices, ambos se mantiveram bem equivalentes.

Com o intuito de analisar os resultados obtidos de uma maneira mais visual, adicionei o widget ‘Pythagorean Forest’ ao algoritmo de ‘Random Forest’, podendo visualizar as múltiplas árvores geradas e compreender quais características foram mais relevantes para as tomadas de decisão.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente com confiança média

Neste sentido, para o algoritmo ‘Tree’, também utilizei uma ferramenta de visualização. Contudo, como o Orange não possibilitou ligar o widget do ‘Tree Viewer’ diretamente ao modelo, eu primeiro adicionei a opção ‘Save Model’ para salvá-lo e recarregá-lo no widget ‘Load Model’, para então conseguir conectar ao visualizador e obter a árvore de decisão, compreendendo a maneira que ela foi montada.

O visualizador ‘Tree Viewer’ exibe a hierarquia da árvore mostrando o nó superior ‘Root’, que neste caso é a feature ‘Parental\_Supervision’, e os demais nós internos, que vão se dividindo de acordo com os critérios estabelecidos. Desta maneira, podemos enxergar de forma clara quais atributos foram cruciais para as decisões e compreender os critérios utilizados.

Diagrama

Descrição gerada automaticamente com confiança média

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente Por fim, adicionei mais 2 widgets ao ‘Test and Score’ com o intuito de obter uma visão geral referente aos índices obtidos. Com a ‘Confusion Matrix’, pude avaliar o desempenho de cada um dos modelos verificando as taxas de cada um referente as classificações de positivos verdadeiros e falsos, e de negativos verdadeiros e falsos.

Ademais, analisando o widget de ‘ROC Analysis’ conseguimos identificar de maneira gráfica a taxa de positivos verdadeiros contra a taxa de positivos falsos, conseguindo comparar o desempenho e equilíbrio dos modelos.

Gráfico

Descrição gerada automaticamente

|  |
| --- |
| Relato dos experimentos e lições aprendidas |
| *Apresente uma reflexão acerca dos resultados obtidos com este projeto.* |
| Levando em conta a análise exploratória realizada e os resultados provenientes dos algoritmos, pude confirmar que os fatores externos como supervisão familiar, saúde mental, histórico familiar e influência de parceiros, estão diretamente ligados ao uso de substâncias. Os modelos treinados, como **Random Forest**, **Árvore de Decisão (Tree)** e **k-NN**, demonstraram uma relação consistente entre essas variáveis e a propensão ao uso de substâncias, mesmo com variações nos índices de desempenho.  Os resultados, embora apresentem limitações nos índices como AUC, CA e F1 Score, corroboram com a hipótese inicial de que esses fatores possuem peso significativo no comportamento relacionado ao consumo de substâncias. A supervisão familiar, por exemplo, mostrou-se uma das variáveis mais relevantes nas divisões dos modelos baseados em árvores, indicando que ambientes com baixa supervisão aumentam o risco. Além disso, questões relacionadas à saúde mental também se destacaram, com indivíduos apresentando maior tendência ao uso em cenários de instabilidade emocional. Desta forma, reforçando a importância de intervenções em saúde mental e de estratégias para fortalecer a supervisão e o suporte familiar como medidas preventivas.  Em conclusão, os resultados obtidos confirmam que fatores psicossociais e ambientais têm uma influência direta no uso de substâncias, destacando a necessidade de políticas públicas e ações específicas que abordem essas questões de forma integrada, visando a redução do consumo de substâncias em populações vulneráveis. |

|  |
| --- |
| LINK PARA O ARQUIVO DO PROJETO DO ORANGE E DOS DADOS UTILIZADOS |
| *Insira os links para os arquivos.* |
| <https://www.kaggle.com/datasets/waqi786/youth-smoking-and-drug-dataset> |