Sugestões: não fiz essa parte no texto abaixo

adicionar subtítulos para cada método e métrica de acurácia.

Referências: adicionar referências teóricas, como livros ou artigos científicos.  
----------------------------------------------------------------------------------

Link Github: <https://github.com/valdineyatilio/ProjetoAplicado-II>

------------------------------------------------------------------------------------

1. Definição da Linguagem de Programação:   
A linguagem que será utilizada será python.  
-------------------------------------------------------------------------------------  
2. Análise Exploratória da Base de Dados:

O Algoritmo da análise exploratória da base de dados se encontra no arquivo AnaliseExploratoria.py e AnaliseExploratoria.ipynb como alternativas de visualização e execução dos códigos referente ao que foi requisitado a apresentar.

---------------------------------------------------------------------------

3. Tratamento da Base de Dados

O Algoritmo do tratamento da base de dados se encontra no arquivo TratamentoDeDados.py e TratamentoDeDados.ipynb como alternativas de visualização e execução dos códigos referente ao que foi requisitado a apresentar.

----------------------------------------------------------------------------------------------------

4. Definição e Descrição das Bases Teóricas dos Métodos

Redes Neurais Artificiais (RNA):

Teoria: As RNAs são inspiradas no funcionamento do cérebro humano e são compostas por camadas de neurônios artificiais. Elas são capazes de aprender padrões complexos a partir dos dados.

Aplicação: utilizando uma rede neural para prever os ganhos anuais mais altos com base em variáveis como assinantes, visualizações de vídeos e uploads.

Regressor MLP (Perceptron Multicamadas):

Teoria: O MLP é um tipo de rede neural feedforward que consiste em uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Ele utiliza a retropropagação para ajustar os pesos durante o treinamento.

Aplicação: Configuração do MLPRegressor com duas camadas ocultas (100 e 50 neurônios) e utilização do algoritmo de gradiente descendente estocástico (SGD) para a otimização.

Definição e descrição de como será calculada a acurácia

Erro Quadrático Médio (MSE):

Teoria: O MSE é uma métrica que mede a média dos quadrados dos erros, ou seja, a diferença entre os valores previstos e os valores reais. É uma medida comum de acurácia para modelos de regressão.

Aplicação: Cálculo do MSE para avaliar a performance do modelo de rede neural. Um MSE menor indica um modelo mais preciso.  
  
  
Definição e descrição das bases teóricas dos métodos

Análise de Variância (ANOVA):

Teoria: A ANOVA é uma técnica estatística usada para comparar as médias de três ou mais grupos para determinar se pelo menos um grupo é significativamente diferente dos outros. Ela se baseia na decomposição da variabilidade total em variabilidade entre grupos e variabilidade dentro dos grupos.

Aplicação: Utilizando a ANOVA para comparar as visualizações de vídeos entre diferentes categorias do YouTube. Isso ajuda a identificar se há diferenças significativas nas visualizações médias entre as categorias.

Definição e descrição de como será calculada a acurácia

Estatística F e Valor p:

Teoria: A estatística F é a razão entre a variabilidade entre os grupos e a variabilidade dentro dos grupos. Um valor F alto indica que há uma diferença significativa entre as médias dos grupos. O valor p é a probabilidade de obter um valor F tão extremo quanto o observado, assumindo que a hipótese nula (de que todas as médias são iguais) é verdadeira.

Aplicação: Calcular a estatística F e o valor p para determinar se há diferenças significativas nas visualizações de vídeos entre as categorias. Um valor p menor que um nível de significância (geralmente 0.05) indica que há uma diferença significativa.

Definição e descrição das bases teóricas dos métodos

Algoritmo Apriori:

Teoria: O algoritmo Apriori é um método clássico de mineração de dados usado para extrair padrões frequentes e regras de associação em grandes bases de dados. Ele funciona iterativamente, identificando conjuntos de itens frequentes e gerando regras de associação a partir desses conjuntos.

Aplicação: Utilizar o algoritmo Apriori para encontrar associações entre países e altos ganhos anuais no YouTube. Isso pode ajudar a identificar padrões interessantes sobre quais países têm maior probabilidade de gerar altos ganhos.

Regras de Associação:

Teoria: As regras de associação são usadas para descobrir relações interessantes entre variáveis em grandes bases de dados. Elas são expressas na forma “se A, então B”, onde A e B são conjuntos de itens. As métricas comuns para avaliar a força das regras incluem suporte, confiança e lift.

Aplicação: Utilizar a função association\_rules para gerar regras de associação a partir dos conjuntos de itens frequentes identificados pelo algoritmo Apriori. As regras geradas podem revelar insights sobre a relação entre o país de origem e a probabilidade de altos ganhos.

Definição e descrição de como será calculada a acurácia

Métricas de Avaliação (Lift, Suporte, Confiança):

Teoria:

Suporte: A proporção de registros no conjunto de dados que contém ambos os itens A e B.

Confiança: A proporção de registros que contém o item B entre aqueles que contêm o item A.

Lift: A razão entre a confiança da regra e a expectativa de encontrar o item B, dado que o item A está presente. Um lift maior que 1 indica uma associação positiva entre A e B.

Aplicação: utilizando essas métricas para avaliar a força das regras de associação geradas. Regras com alto suporte, confiança e lift são consideradas mais significativas.

5. Definição e Descrição da Acurácia

acuracia = accuracy\_score(y\_test, y\_pred)

Métricas para Avaliar a Acurácia

Métricas importantes:

Precisão (Precision): precisao = precision\_score(y\_test, y\_pred, average='weighted', zero\_division=1)

Recall (Sensibilidade): recall = recall\_score(y\_test, y\_pred, average='weighted', zero\_division=1)

F1-Score: f1 = f1\_score(y\_test, y\_pred, average='weighted', zero\_division=1)

AUC-ROC: O código verifica se há mais de uma classe em y\_test antes de calcular o AUC-ROC:

if len(set(y\_test)) > 1:

y\_test\_binarized = label\_binarize(y\_test, classes=classes)

y\_pred\_proba = modelo.predict\_proba(X\_test)

roc\_auc = roc\_auc\_score(y\_test\_binarized, y\_pred\_proba, multi\_class='ovr')

Processo de Validação

O código utiliza a técnica de divisão treino-teste para validar o modelo:

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

Conclusão

O código calcula a acurácia, utiliza métricas adicionais (precisão, recall, F1-score, AUC-ROC) e aplica uma técnica de validação (divisão treino-teste) para garantir a robustez dos resultados.