1. INTRODUÇÃO

O presente relatório apresenta o desenvolvimento e a avaliação de modelos de inteligência artificial aplicados a um problema de classificação, no qual o objetivo é prever se um usuário realizará a compra de uma casa num site imobiliário.

2. ANÁLISE EXPLORATÓRIA

Inicialmente, foi realizada uma limpeza dos dados, eliminando linhas com valores nulos e registros com valores negativos em colunas onde tais valores eram logicamente inválidos, como o tempo que o usuário permaneceu no site. Após essa etapa, o dataset foi reorganizado, garantindo que todas as informações fossem válidas e consistentes.

```
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Idade                190 non-null    float64
 1   Renda Anual (em $)   189 non-null    float64
 2   Gênero               193 non-null    object
 3   Tempo no Site (min)  200 non-null    float64
 4   Anúncio Clicado      190 non-null    object
 5   Compra (0 ou 1)      200 non-null    int64
dtypes: float64(3), int64(1), object(2)
memory usage: 9.5+ KB
```

Figura 1 - Informações do dataset original

```
RangeIndex: 165 entries, 0 to 164
Data columns (total 6 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Idade                165 non-null    float64
 1   Renda Anual (em $)   165 non-null    float64
 2   Gênero               165 non-null    object
 3   Tempo no Site (min)  165 non-null    float64
 4   Anúncio Clicado      165 non-null    object
 5   Compra (0 ou 1)      165 non-null    int64
dtypes: float64(3), int64(1), object(2)
memory usage: 7.9+ KB
```

Figura 2 - Informações do dataset após limpeza

Em seguida, verificou-se as distribuições das variáveis independentes e a associação destas com a variável alvo “compra”. Quanto à idade dos usuários, observamos que os dados estão relativamente bem distribuídos entre 18 e 60 anos.

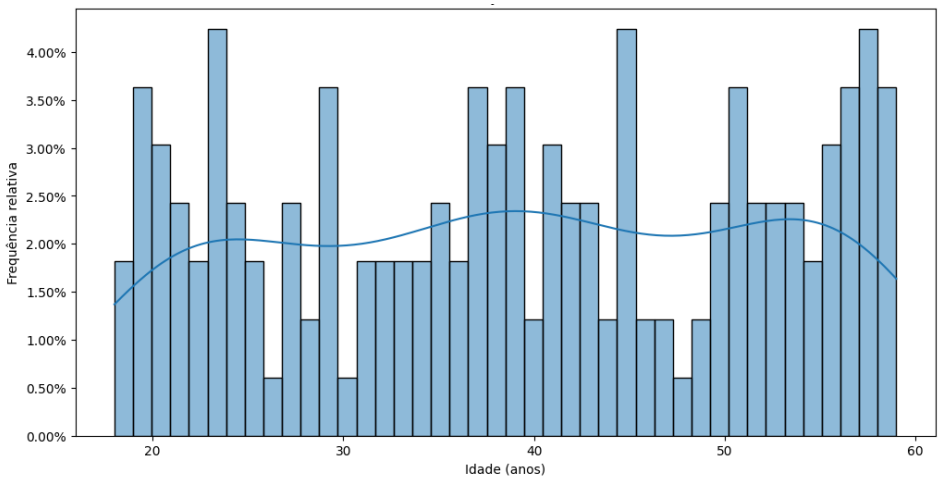

Figura 3 - Distribuição de idade

Entretanto, usuários com menos de 20 anos representam uma minoria e possuem menor propensão a adquirir uma casa. Especificamente para este grupo de indivíduos, a análise revelou que 85,71% deles não compram uma casa. Essa relação pode ser atribuída à falta de independência financeira ou maturidade, fatores geralmente associados a indivíduos mais jovens.

Quanto à renda anual dos usuários, a concentração é maior na faixa dos 30 a 50 mil dólares.

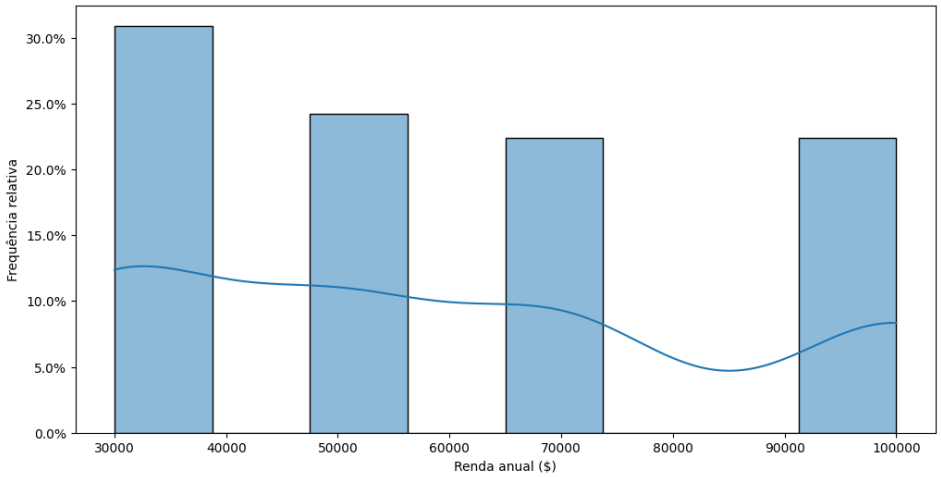

Figura 4 - Distribuição de renda anual

No entanto, foi possível identificar que a proporção de usuários que compram casas é inferior na faixa de renda anual acima de 70 mil dólares, revelando que esse fator não é tão determinante quanto se espera.

O tempo de uso do site contrasta fortemente com isso. A exploração deixa evidente que usuários com menos de 10 minutos de navegação raramente compram casas, o que talvez classifique a interação com o site como decisivo para a compra.

```
temp_df = df[df["Tempo no Site (min)"] ⩽ 10]
display(temp_df["Compra (0 ou 1)"].value_counts(normalize = True).map("{:.2%}".format))
```

✓ 0.0s

```
Compra (0 ou 1)
0    81.25%
1    18.75%
```

Figura 5 - Proporção de compra entre os usuários que passam menos de 10 minutos no site

As últimas duas variáveis (“gênero” e “anunciado clicado”) apresentam distribuições bem uniformes, mas não percebemos uma relação clara delas com a variável alvo. Tanto homens como mulheres, independentemente de verem ou não anúncios, compram casas em proporções similares. Este cenário pode ser um ponto de ajuste para a imobiliária, pois demonstra que as propagandas estão afastando os usuários ou não estão bem construídas no quesito de persuasão.

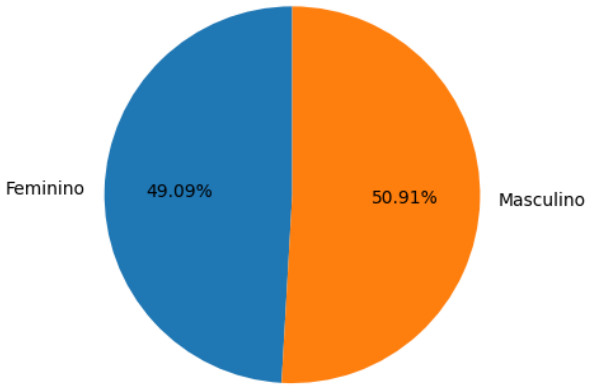

Figura 6 - Distribuição de gênero

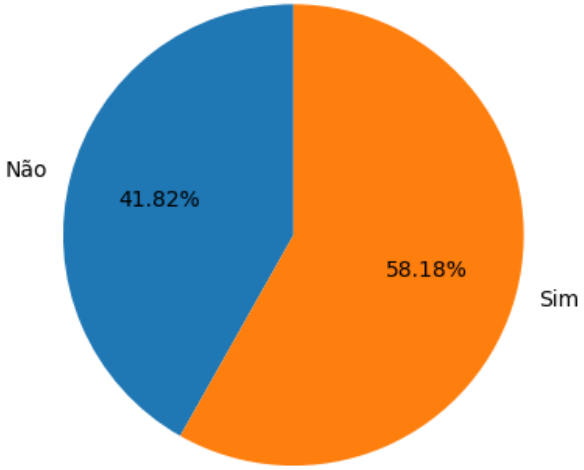

Figura 7 - Distribuição de anúncio clicado

Por fim, separando os usuários entre aqueles que compraram uma casa e aqueles que não, obtemos:

Compra (0 ou 1)	Idade	Renda Anual (em $)	Tempo no Site (min)
0	38.419643	60625.000000	17.331078
1	41.018868	57169.811321	19.396592

Figura 8 - Médias das variáveis numéricas dos dois grupos

Logo, pode-se esperar que as variáveis “idade” e “tempo no site” contribuam mais para as previsões feitas pelos algoritmos de IA. As demais variáveis, por terem associações menos explícitas com a “compra”, podem induzir erros durante o processo de treinamento, o que pode prejudicar a precisão dos modelos.

3. PRÉ-PROCESSAMENTO

Nesta etapa, usamos o LabelEncoder para transformar variáveis categóricas, como “gênero” e “anúncio clicado”, em valores numéricos. Algoritmos de machine learning, não conseguem processar diretamente variáveis de texto, por isso é necessário convertê-las para números. O LabelEncoder mapeia cada categoria para um número inteiro, permitindo que o modelo possa entender essas variáveis e usá-las para fazer previsões de forma eficiente.

	Idade	Renda Anual (em $)	Gênero	Tempo no Site (min)	Anúncio Clicado	Compra (0 ou 1)
0	29.0	30000.0	Feminino	5.741706	Não	0
1	58.0	50000.0	Feminino	21.885505	Sim	0
2	50.0	70000.0	Feminino	6.121339	Não	1
3	21.0	100000.0	Feminino	16.264925	Não	0
4	59.0	50000.0	Feminino	28.235667	Sim	1
...	...	...	...	...	...	...
160	34.0	70000.0	Feminino	9.338113	Sim	0
161	42.0	50000.0	Feminino	22.373777	Não	0
162	45.0	100000.0	Masculino	13.652493	Sim	0
163	54.0	30000.0	Feminino	25.562014	Não	1
164	18.0	50000.0	Masculino	26.550362	Não	0

Figura 9 - Valores do dataset original

	Idade	Renda Anual (em $)	Gênero	Tempo no Site (min)	Anúncio Clicado	Compra (0 ou 1)
0	29.0	30000.0	0	5.741706	0	0
1	58.0	50000.0	0	21.885505	1	0
2	50.0	70000.0	0	6.121339	0	1
3	21.0	100000.0	0	16.264925	0	0
4	59.0	50000.0	0	28.235667	1	1
...	...	...	...	...	...	...
160	34.0	70000.0	0	9.338113	1	0
161	42.0	50000.0	0	22.373777	0	0
162	45.0	100000.0	1	13.652493	1	0
163	54.0	30000.0	0	25.562014	0	1
164	18.0	50000.0	1	26.550362	0	0

Figura 10 - Valores do dataset após codificação das variáveis categóricas

Em seguida, realizamos a normalização das variáveis utilizando o StandardScaler. O objetivo dessa etapa é ajustar a escala das variáveis para que elas tenham uma distribuição com média 0 e desvio padrão 1. Se as variáveis tiverem escalas muito diferentes, como uma com valores na ordem de milhar e outra em torno de 1, a variável de maior escala pode ter mais influência no modelo, distorcendo os resultados. A normalização assegura que todas as variáveis contribuam igualmente para o treinamento do modelo.

Por fim, realizou-se a divisão dos dados em conjuntos de treinamento e teste utilizando a função train_test_split. Os dados foram divididos na proporção de 80% para treinamento e 20% para teste. Essa divisão é feita para garantir que o modelo seja treinado com uma parte dos dados e, em seguida, avaliado em dados independentes. Essa prática ajuda a prevenir o overfitting, um problema que ocorre quando o modelo aprende muito

bem os dados de treinamento, mas não consegue generalizar para novos dados. Ao separar os dados dessa forma, podemos garantir que a avaliação do desempenho do modelo seja justa e eficaz, refletindo sua capacidade de generalizar para dados desconhecidos.

4. CONSTRUÇÃO DOS MODELOS DE CLASSIFICAÇÃO

Foram escolhidos quatro modelos a serem aplicados no problema: Decision Tree, K-Nearest Neighbors, Logistic Regression e Random Forest. Tais modelos são algoritmos supervisionados de aprendizado, ou seja, requerem dados rotulados para treinamento (que é o caso do dataset em estudo) e fazem previsões com base em padrões aprendidos a partir desses dados. Esses modelos buscam identificar a relação entre as variáveis independentes e a variável alvo. Apesar dessa semelhança fundamental, cada modelo possui características únicas que influenciam seu desempenho e aplicação, e cada um apresenta vantagens e limitações dependendo da natureza dos dados e do problema em questão.

Decision Tree (DT) é um modelo simples que divide os dados em subgrupos com base em características específicas para prever o resultado. Cada nó na árvore representa uma decisão sobre uma variável, e as folhas indicam o resultado final. Ele é intuitivo e fácil de interpretar, mas, se muito profundo, tem sua capacidade de generalização prejudicada.

K-Nearest Neighbors (KNN) faz previsões com base nos K vizinhos mais próximos a um ponto de dados. O modelo classifica os dados de acordo com a maioria dos rótulos dos vizinhos mais próximos. Sua principal vantagem é a flexibilidade para capturar padrões complexos, mas pode ser computacionalmente caro com grandes volumes de dados, além de ser sensível à escala das variáveis, exigindo normalização.

Logistic Regression (LR) é usado para problemas de classificação binária, prevendo a probabilidade de um evento com base nas variáveis de entrada. Sua simplicidade e eficiência são vantagens, especialmente em problemas lineares, mas ela tem limitações em capturar relações não lineares complexas entre as variáveis, o que pode ser uma desvantagem em cenários mais complicados.

Random Forest (RF) combina várias árvores de decisão, reduzindo o overfitting e melhorando a capacidade de generalização. Ele lida bem com grandes conjuntos de dados e captura interações complexas entre variáveis. No entanto, sua complexidade torna a interpretação mais difícil em comparação com modelos mais simples.

Durante o treinamento de todos os modelos, utilizou-se validação cruzada e GridSearch para encontrar a combinação mais otimizada de hiperparâmetros, visando melhorar o desempenho e a generalização dos modelos. A validação cruzada ajuda a avaliar a performance do modelo de forma robusta, dividindo o conjunto de dados em várias partes e treinando o modelo em diferentes divisões.

O GridSearch executa uma busca por todas as combinações possíveis dentro de um intervalo predefinido de valores. Esta técnica foi selecionada porque o dataset do problema é relativamente pequeno, o que torna viável explorar várias combinações de forma eficiente. Embora o Random Search também seja uma alternativa válida, ele é mais utilizado quando o espaço de parâmetros é muito grande e o tempo de computação é mais restrito.

	param_n_estimators	param_max_depth	param_min_samples_split	param_min_samples_leaf	param_bootstrap
0	50	10	2	1	False
1	100	10	2	1	False
2	150	10	2	1	False
3	50	10	5	1	False
4	100	10	5	1	False
...	...	...	...	...	...
157	100	30	5	4	True
158	150	30	5	4	True
159	50	30	10	4	True
160	100	30	10	4	True
161	150	30	10	4	True

Figura 11 - Exemplo com os parâmetros avaliados durante o treinamento do modelo RF

5. RESULTADOS E CONCLUSÕES

Os resultados obtidos nos treinamentos fornecem uma visão geral do desempenho de cada abordagem na tarefa de prever se um usuário realizará ou não a compra de uma casa.

No caso do modelo de árvore de decisão, embora tenha apresentado uma precisão de validação cruzada razoável (62,8%), seu desempenho no conjunto de teste foi mais limitado, com uma precisão de 54,5%. Observa-se que a classe majoritária (compra = 0) foi favorecida, com uma precisão de 58% e um recall de 79%, enquanto a classe minoritária (compra = 1) obteve um recall de apenas 21%. Isso indica que o modelo não conseguiu capturar adequadamente os padrões associados à classe minoritária, o que é esperado em cenários de dados desbalanceados. Além disso, a matriz de confusão reflete que, na maioria das vezes, o modelo classificou erroneamente os compradores (compra = 1) como não compradores (compra = 0).

	precision	recall	f1-score	support
0	0.58	0.79	0.67	19
1	0.43	0.21	0.29	14
accuracy			0.55	33
macro avg	0.50	0.50	0.48	33
weighted avg	0.51	0.55	0.51	33

Figura 12 - Métricas de desempenho do modelo DT

O KNN obteve uma precisão de validação cruzada superior (69%) e uma precisão no conjunto de teste de 60,6%. O modelo mostrou um recall elevado para a classe 0 (89%) e um recall muito baixo para a classe 1 (21%), indicando novamente que o modelo apresenta dificuldades em lidar com a classe minoritária.

	precision	recall	f1-score	support
0	0.61	0.89	0.72	19
1	0.60	0.21	0.32	14
accuracy			0.61	33
macro avg	0.60	0.55	0.52	33
weighted avg	0.60	0.61	0.55	33

Figura 13 - Métricas de desempenho do modelo KNN

Para o modelo de regressão logística, os resultados mostram uma precisão no conjunto de teste semelhante à do KNN, com 60,6%. Entretanto, o balanceamento entre as métricas de precisão e recall para ambas as classes foi mais equilibrado do que nos modelos anteriores. Por exemplo, a classe 0 alcançou uma precisão de 65% e um recall de 68%, enquanto a classe 1 obteve valores menores, mas relativamente mais próximos (54% de precisão e 50% de recall). Isso sugere que a regressão logística conseguiu identificar alguns padrões relevantes, ainda que de forma limitada.

	precision	recall	f1-score	support
0	0.65	0.68	0.67	19
1	0.54	0.50	0.52	14
accuracy			0.61	33
macro avg	0.59	0.59	0.59	33
weighted avg	0.60	0.61	0.60	33

Figura 14 - Métricas de desempenho do modelo LR

Por fim, o modelo de floresta aleatória apresentou resultados similares aos do KNN e da regressão logística, com uma precisão no conjunto de teste de 60,6%. Assim como nos outros modelos, o recall da classe minoritária (compra = 1) foi muito baixo (14%), enquanto o da classe majoritária foi elevado (95%). Embora o modelo tenha a capacidade de explorar relações mais complexas entre as variáveis, ele parece ter enfrentado dificuldades devido ao desbalanceamento dos dados ou à limitada quantidade de informações no conjunto de treino.

	precision	recall	f1-score	support
0	0.60	0.95	0.73	19
1	0.67	0.14	0.24	14
accuracy			0.61	33
macro avg	0.63	0.55	0.48	33
weighted avg	0.63	0.61	0.52	33

Figura 15 - Métricas de desempenho do modelo RF

De forma geral, os quatro modelos apresentaram resultados modestos, com precisão e outras métricas oscilando em torno de 60%. Isso pode ser atribuído a vários fatores, incluindo o desbalanceamento da variável alvo, o tamanho limitado do dataset e a natureza das variáveis independentes.

No que diz respeito à influência das variáveis na decisão do modelo, o comportamento da árvore de decisão e da floresta aleatória atenderam às expectativas. Ambas as técnicas identificaram “tempo no site” como a variável mais relevante, seguida por “idade”. Esses resultados corroboram com a análise inicial dos dados.

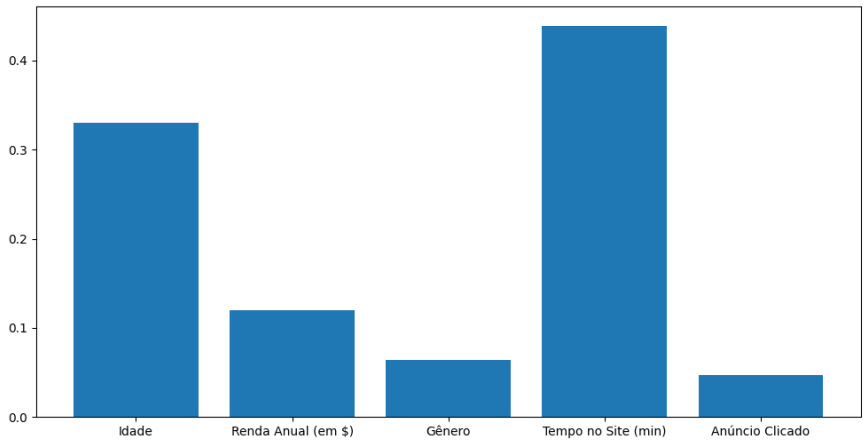

Figura 16 - Influência das variáveis na decisão do modelo DT

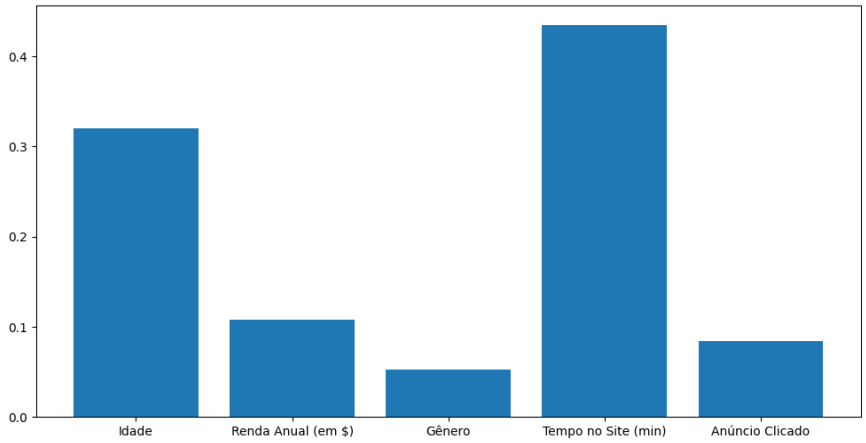

Figura 17 - Influência das variáveis na decisão do modelo RF

Por outro lado, os modelos KNN e de regressão logística apresentaram um comportamento divergente. No caso do KNN, o tempo no site também se destacou, mas,

curiosamente, o gênero apareceu como uma das variáveis mais determinantes. Esse comportamento pode ser justificado pela sensibilidade do KNN à distribuição local dos dados no espaço dimensional. Caso existam padrões regionais de gênero associados à compra de uma casa em pequenos subconjuntos dos dados, o algoritmo pode ter amplificado essa característica.

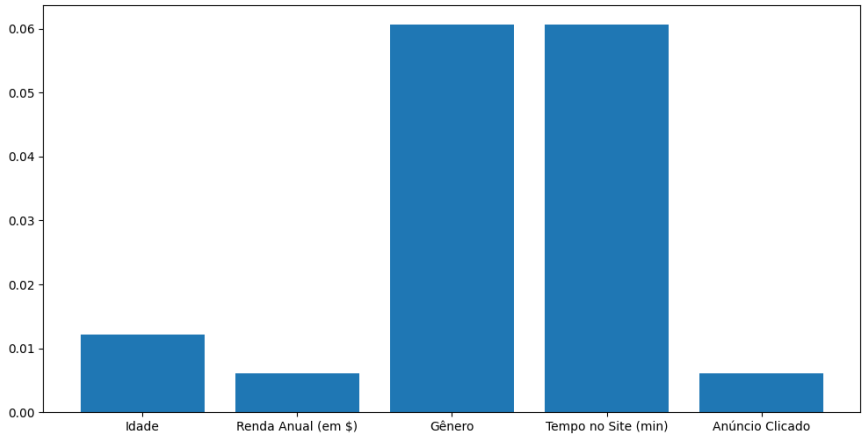

Figura 18 - Influência das variáveis na decisão do modelo KNN

No modelo LR, a variável mais relevante foi “anúncio clicado”, seguida por “tempo no site” e “renda anual”. A idade, que teve grande destaque na análise exploratória, recebeu uma menor importância relativa. Esse comportamento pode ser explicado pelas limitações intrínsecas do modelo de regressão logística, que assume relações lineares entre as variáveis preditoras e a variável alvo.

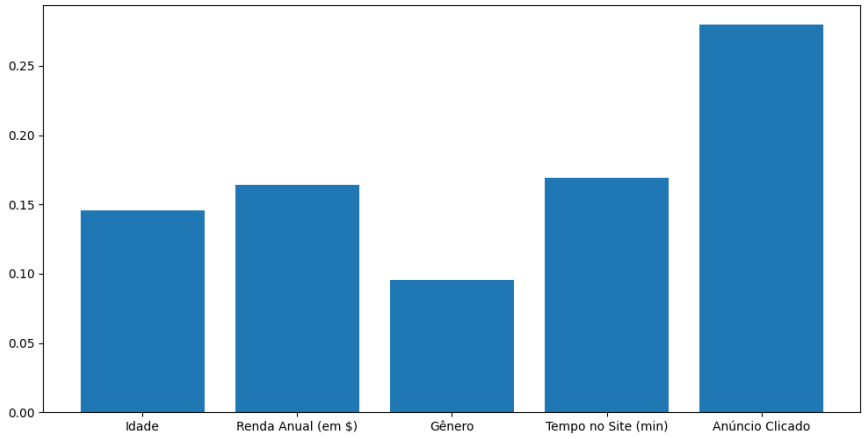

Figura 19 - Influência das variáveis na decisão do modelo LR

Como melhorias, a principal recomendação seria a ampliação do conjunto de dados, tanto em termos de quantidade quanto de diversidade, para permitir que os

modelos aprendam padrões mais representativos e generalizáveis. Coletar mais informações sobre os usuários e suas interações com o site pode ajudar a enriquecer as variáveis e melhorar a qualidade das previsões. Além disso, estratégias simples, como balanceamento do conjunto de dados, podem aumentar o recall da classe minoritária. Por fim, revisar as transformações aplicadas às variáveis para melhor adequá-las aos pressupostos dos modelos (como linearidade no caso da regressão logística) pode ajudar a melhorar o desempenho geral.