Data Science Code Challenge – BelugaDB

Candidata: Mariana Vieira Ribeiro Lopes

1. **Análise preliminar dos dados**
   Inicialmente, foi realizada uma breve análise dos dados obtidos, verificando-se o balanceamento das classes (para verificar se haviam exemplos suficientes de ambas), a porcentagem de valores nulos em cada coluna, distribuição dos valores das variáveis em relação à variável alvo e entropia das features.

2. **Redução de dimensionalidade**
   As informações obtidas na análise anterior serviram de base para a redução de dimensionalidade do problema. Embora o algoritmo de árvore de decisão utilizado já apresente uma seleção de atributos embutida, verificou-se a necessidade de reduzir a dimensionalidade do conjunto devido à binarização realizada nos dados: como haviam variáveis categóricas com um número muito grande de valores possíveis, ao serem binarizadas, gerariam um número muito grande de features. Ao observar que essas variáveis apresentavam também uma entropia alta, foi decidido pela exclusão das mesmas do conjunto de treinamento.

3. **Engenharia de features**
   A ‘fetaure_0’ que apresentava valores do tipo “*datetime*” foi fracionada em 3 novas features, mais informativas: dia da semana, mês e hora.

4. **Tratamento de valores faltantes**
   Após a pré-seleção de features, foi realizado um tratamento dos valores nulos nas features selecionadas, substituindo-se os valores nulos pela moda da série em questão, já que tratava-se de atributos categóricos.

5. **Binarização e tratamentos finais**
   Conforme mencionado anteriormente, aplicou-se uma técnica denominada binarização para adequar o formato da entrada para o algoritmo árvore de decisão.
   Além disso, fez-se uma manipulação nos dados a fim de garantir que as colunas o conjunto de treinamento incluíssem todas as colunas geradas pela binarização do conjunto de teste e vice versa.

6. **Treinamento e predição**
   Separou-se o conjunto de treinamento original em conjuntos de treinamento e teste, a fim de aplicar-se a técnica de validação *holdout*, com 33% dos dados destinados ao conjunto de teste e o restante para treinamento.
   Um algoritmo de árvore de decisão foi utilizado para treinar o modelo preditivo, que apresentou 78,73% de especificidade.
   Este modelo foi utilizado para classificar o conjunto de testes original e os resultados da predição foram salvos no arquivo 'id_predicao.csv'.