# Uma Análise de desempenho de Classificadores de Dados de Acidentes em indústrias

**Alunos: Luís Antonio de Almeida Rodriguez- MsC, Raoni Avilez Fiedler- Esp, Odair Oliveira de Sá-MsC**

**Resumo – O trabalho relatado neste documento versa sobre um conjunto de dados tabulares a respeito de acidentes de trabalho que precisa ser analisado. As questões de relevância foram mostradas no enunciado do trabalho e, para respondê-las, a equipe usou técnicas de Machine Learning aprendidas nas aulas, tais como a Preparação dos Dados, a construção de três classificadores *Naive Bayes* e um classificador Bayesiano que tomou como base a estrutura dos dados fornecidos. Dois classificadores foram comparados em seus desempenhos e foram observados resultados que mostraram maior relevância de variáveis para os processos decisórios e outras respostas às questões de relevância.**

**Palavras-chave – Classificador Bayes Ingênuo, Classificador Bayesiano Estruturado, Acidentes de Trabalho.**

## 1. INTRODUÇÃO

Classificação é o processo de predição de um conjunto de Classes, dado um conjunto de *"data points"*. As classes também podem ser chamadas de alvos ou categorias e a modelagem da classificação preditiva é a ação de tornar mais próxima uma função de mapeamento (f) considerando variáveis de entrada (x) e de saída (y), descritas em um *dataset* tal qual utilizado neste trabalho.

O foco deste trabalho é realizar a classificação dos dados fornecidos de forma a responder questões de relevância (QR) descritas no enunciado, tais como padrões relevantes nos dados, variáveis categóricas e outras.

## 2. DESCRIÇÃO DO PROCESSO E RESULTADOS OBTIDOS

O trabalho foi dividido de acordo com a estrutura proposta nas QR. Elas foram tomadas como metas e divididas em pequenas tarefas que geraram entregáveis:

- Descrição dos classificadores e do *dataset*
- Dados e resultados da comparação
- Melhorias possíveis para o classificador
- Padrões relevantes existentes nos dados

Os resultados obtidos foram: - um *dataset* preparado para ser submetido aos algoritmos e a implementação em python de dois classificadores, um deles com base no modelo *Naive Bayes* e um deles estruturado, com base nos dados fornecidos. Foram produzidos artefatos que mostraram a análise feita pela equipe através de gráficos e há uma Seção destinada à comparação entre dois dos classificadores, nos termos solicitados nas QR.

### 2.1. DESCRIÇÃO DOS CLASSIFICADORES E DO DATASET

O dataset foi fornecido de forma tabular, no formato *.csv,* e contém a descrição de vários itens relativos a acidentes de trabalho documentados, linha por linha, sendo cada uma delas um deles. Com alguns poucos comandos do Python3 executados no colab, temos o resultado da Figura 1. Após essa visualização, a equipe iniciou as análises e a Preparação dos Dados.

| | Data | Countries | Local | Industry Sector | Accident Level | Potential Accident Level | Genre | Employee ou Terceiro | Risco Critico |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 2016-01-01 00:00:00 | Country_01 | Local_01 | Mining | I | IV | Male | Third Party | Pressed |
| 1 | 2016-01-02 00:00:00 | Country_02 | Local_02 | Mining | I | IV | Male | Employee | Pressurized Systems |
| 2 | 2016-01-06 00:00:00 | Country_01 | Local_03 | Mining | I | III | Male | Third Party (Remote) | Manual Tools |
| 3 | 2016-01-08 00:00:00 | Country_01 | Local_04 | Mining | I | I | Male | Third Party | Others |
| 4 | 2016-01-10 00:00:00 | Country_01 | Local_04 | Mining | IV | IV | Male | Third Party | Others |

Figura 1 - O dataset

Foram encontradas nove colunas e 439 linhas. A Figura 1 mostra o comando python e o código-fonte pode ser visto em sua totalidade no **Anexo 1 - Implementação**.

Iniciando a Preparação dos Dados, foi verificada a completude do preenchimento de cada célula 'X,Y' e percebeu-se, conforme a Figura 2, que não havia dados nulos no dataset fornecido.

```
Data                      0
Countries                 0
Local                     0
Industry Sector           0
Accident Level            0
Potential Accident Level  0
Genre                     0
Employee ou Terceiro      0
Risco Critico             0
dtype: int64
```

Figura 2 - Verificação de completude

Após a verificação de completude, foi feita uma análise para saber quais eram os dados presentes em cada coluna, quantos tipos diferentes eram e quantas ocorrências de cada um havia. Esse procedimento foi implementado para cada uma das colunas do dataset, conforme a Figura 3.

```
Local_03    90
Local_05    59
Local_06    58
Local_01    57
Local_04    56
Local_10    44
Local_08    29
Local_02    24
Local_07    14
Local_12     4
Local_09     2
Local_11     2
Name: Local, dtype: int64
```

Figura 3 - Dados por coluna (Local)

Em algumas colunas foi possível perceber que os dados não estavam balanceados, poderia haver alguns problemas, por exemplo, na classificação dos Níveis de acidente, pois, os mais baixos têm pouca amostragem. A Figura 4 mostra isso.

```
I      328
II      40
III     31
IV      31
V        9
Name: Accident Level, dtype: int64
```

Figura 4 - Dados desbalanceados

A coluna *Risco Crítico* apresentou um grande número de diferentes tipos de registros. A melhor ação naquele momento foi analisar se existem alguns desses registros com significado semelhante, ou, se podem ser agrupados por uma categorização mais genérica. Se os significados semânticos dos mesmos forem próximos, isso é bom porque determina um número menor de "tipos" e uma amostragem maior para o agrupamento.

Ao listar a coluna *Risco Crítico*, é possível, de imediato, perceber que existem dois tipos "*Not Applicable*" e "*/Not Applicable*" que já podem ser agrupados em somente um. A Figura 5 mostra a diferença da coluna *Risco Crítico* antes e depois. Após o agrupamento, a coluna diminuiu de 34 itens para 23 itens.

Para uma melhor visualização dos dados, a coluna *Data* foi separada em quatro outras colunas de interesse: *day, month, year e Week_day.* Além disso, foi feita uma reorganização das colunas, deixando '*Year, Month, Day e Week_Day* à esquerda e a coluna *Accident_level* no final, foi

considerada de relevância e colocada à direita. Isso pode ser visto na Figura 6.

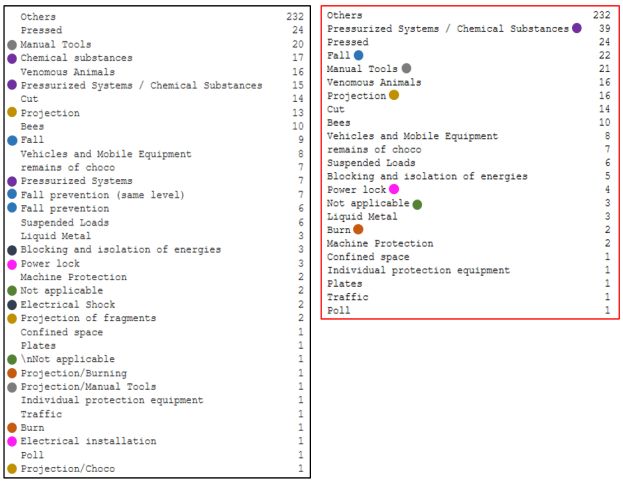


Figura 5 - Coluna *Risco Crítico*

Foram implementados também vários agrupamentos de tipos por significado, onde a equipe, em consenso, acreditou ter feito as juntadas por semelhança, o que pode ser visto na Figura 7.

| | Year | Month | Day | Week_day | Countries | Local | Industry Sector | Potential Accident Level | Genre | Employee ou Terceiro | Risco Critico | Accident Level |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2016 | 1 | 1 | 4 | Country_01 | Local_01 | Mining | IV | Male | Third Party | Pressed | I |
| 1 | 2016 | 1 | 2 | 5 | Country_02 | Local_02 | Mining | IV | Male | Employee | Pressurized Systems / Chemical Substances | I |
| 2 | 2016 | 1 | 6 | 2 | Country_01 | Local_03 | Mining | III | Male | Third Party (Remote) | Manual Tools | I |
| 3 | 2016 | 1 | 8 | 4 | Country_01 | Local_04 | Mining | I | Male | Third Party | Others | I |
| 4 | 2016 | 1 | 10 | 6 | Country_01 | Local_04 | Mining | IV | Male | Third Party | Others | IV |

Figura 6 - Reorganização

Com relação às colunas *Potential Accident Level* e *Accident Level*, os algarismos romanos foram trocados para Arábicos por uma questão de melhor visualização.

Foi feita uma categorização das colunas de dados e o resultado pode ser visto na Figura 7. Na Figura 8 é mostrada a divisão da coluna Data para verificar a relevância de cada parte individual dessa divisão, ou, dia, dia da semana, mês e ano.

A coluna *Week_day* representa os dias da semana de segunda-feira a domingo com valores entre "0" a "6", relativos aos dias, como na Figura 9.

| | Data | Countries | Local | Industry Sector | Accident Level | Potential Accident Level | Genre | Employee ou Terceiro | Risco Critico |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 01-01-16 0:00:00 | Country_01 | Local_01 | Mining | I | IV | Male | Third Party | Pressed |
| 1 | 02-01-16 0:00:00 | Country_02 | Local_02 | Mining | I | IV | Male | Employee | Pressurized Systems |
| 2 | 06-01-16 0:00:00 | Country_01 | Local_03 | Mining | I | III | Male | Third Party (Remote) | Manual Tools |
| 3 | 08-01-16 0:00:00 | Country_01 | Local_04 | Mining | I | I | Male | Third Party | Others |
| 4 | 10-01-16 0:00:00 | Country_01 | Local_04 | Mining | IV | IV | Male | Third Party | Others |

Coluna de data
Colunas categóricas
Rótulo de saída
Coluna com valores ordinários

Figura 7 - Categorização das Colunas

Aquela Figura também mostra que a equipe entendeu que não há amostragem contendo padrões relevantes nessa divisão, a não ser com relação ao dia da semana. Os outros eram irrelevantes, *Week_Day* foi colocado em destaque, eliminando-se as outras variáveis, conforme a Figura 10.

| | Year | Month | Day | Week_day | Countries | Local | Industry Sector | Potential Accident Level | Genre | Employee ou Terceiro | Risco Critico | Accident Level |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2016 | 1 | 1 | 4 | Country_01 | Local_01 | Mining | IV | Male | Third Party | Pressed | I |
| 1 | 2016 | 1 | 2 | 5 | Country_02 | Local_02 | Mining | IV | Male | Employee | Pressurized Systems | I |
| 2 | 2016 | 1 | 6 | 2 | Country_01 | Local_03 | Mining | III | Male | Third Party (Remote) | Manual Tools | I |
| 3 | 2016 | 1 | 8 | 4 | Country_01 | Local_04 | Mining | I | Male | Third Party | Others | I |
| 4 | 2016 | 1 | 10 | 6 | Country_01 | Local_04 | Mining | IV | Male | Third Party | Others | IV |

Figura 8 -Separação da coluna *Data*

É possível ver mais análises a respeito dos dados na Seção 2.4, onde são mostradas relações de influência entre as variáveis, através de gráficos.

| | Week_day | Countries | Local | Industry Sector | Potential Accident Level | Genre | Employee ou Terceiro | Risco Critico | Accident Level |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 4 | Country_01 | Local_01 | Mining | IV | Male | Third Party | Pressed | I |
| 1 | 5 | Country_02 | Local_02 | Mining | IV | Male | Employee | Pressurized Systems | I |
| 2 | 2 | Country_01 | Local_03 | Mining | III | Male | Third Party (Remote) | Manual Tools | I |
| 3 | 4 | Country_01 | Local_04 | Mining | I | Male | Third Party | Others | I |
| 4 | 6 | Country_01 | Local_04 | Mining | IV | Male | Third Party | Others | IV |

Figura 9 - Apenas *Week_Day*

O *dataset* foi separado em 20% para o conjunto de teste e 80% para o de treinamento. Partiu-se para a ação de parametrizar a estrutura, caso contrário ela não seria utilizável para a aplicação dos algoritmos. As colunas *Accident_Level* e *Potential_AccidentLevel* tiveram seus valores convertidos para números decimais. As demais colunas foram convertidas para o formato *String* para que houvesse a possibilidade de encontrar as correlações entre elas.

É necessário definir valores numéricos e caracterizar as relações de influência entre as variáveis. Caso o modelo fosse categórico, a *Distribuição de Dirichlet* poderia ser aplicada. Essa distribuição pode ser feita para um único nó. Neste trabalho os dados mostram múltiplos nós, e, sendo assim, decidiu-se aplicar o algoritmo Multinomial para a parametrização da Rede Bayesiana.

Então foi implementada em python a iteração sobre todos os nós, de acordo com o *Método Spiegelhalter e Lauritzen*. Sendo assim, o modelo implementado no primeiro classificador foi o *Naive Bayes Multinomial*.

Iniciando a construção dos classificadores, foi implementado o Naive Bayes. É possível perceber a implementação (Anexo 1) usando poucos comandos e, na última linha, a chamada de uma função em python chamada `def resultado_modelo()`, que executa fatores de análise com resultados baseados em *taxa de acerto, matriz de confusão, erro quadrático médio e estatística kappa utilizando a técnica de 10-folds.*

Dessa forma, a função foi executada sobre o modelo *Naive Bayes*, permitindo que fossem gerados dados que foram tabulados para serem comparados com o outro Classificador, tendo esses resultados sido apresentados na Seção 2.2.

Esta Seção foi destinada apenas à *descrição da forma com que os classificadores foram implementados.* A Seção 2.2 mostra várias análises relevantes sobre os dados e os resultados apresentados por cada um dos classificadores, e, além disso, o código-fonte descrito no Anexo 1 - Implementação, mostra com detalhes todos os métodos e bibliotecas usadas.

Após finalizada a construção do primeiro classificador, foi implementado um classificador bayesiano com estrutura determinada a partir dos dados e foram preparados diferentes parâmetros de treinamento com o intuito de encontrar um modelo melhor que o Naive Bayes.

Para esse classificador foram implementadas algumas funções:

- ***def pred_bayes(data, modelo_treinado):*** *retrata a predição dos valores do modelo Bayesiano*, onde *'data'* é um dataset no formato *pandas* no qual o valor de saída, ou, o rótulo, precisa

obrigatoriamente estar na última coluna. '*modelo'* é o modelo da Rede Bayesiana treinada pela biblioteca *bnlearn* e a saída é um objeto *Series* do pandas com os valores da predição.

- ***def kfoldcv(indices, k = 10, seed = randon_state):*** retrata a função k-fold, com dez folds.
- ***def dictSplitTrainTest(dataset_treino, k=10, seed=20):*** retorna um dicionário que contém os índices dos dois sets, de treinamento e de teste em k folds.
- ***def cvBayes(data_treino, modelo_bayes, k=10, seed=20):*** retrata a função de validação cruzada k-fold. Retorna 3 listas com k resultados, sendo a primeira a da acurácia, a segunda a estatística Kappa e a terceira o erro quadrático médio.

## 2.2. DADOS E RESULTADOS DA COMPARAÇÃO

Na construção dos dois classificadores foram implementadas, respectivamente, funções que produziram dados de desempenho dos mesmos, possibilitando realizar uma comparação com base no mesmo conjunto de parâmetros: *taxa de acerto, matriz de confusão, erro quadrático médio e estatística kappa utilizando a técnica de K-folds.*

- O Classificador Naive Bayes

É possível observar as métricas do uso *ten-fold* e aquelas aplicadas aos conjuntos de treinamento e teste. A Figura 10 mostra esses dados em duas tabelas:

| **Métricas com 10 K-fold** | |
|---|---|
| Acurácia Média | 0.7492 |
| Kappa Médio | -0.0068 |
| RMSE Médio | 1.1549 |
| **Métricas no Conjunto de Treinamento e Teste** | |
| Acurácia no Treinamento | 0.7492 |
| Acurácia no Teste | 0.7386 |
| Kappa | 0.0 |
| RMSE | 1.2954 |

| | ***precision*** | ***recall*** | ***f1-score*** | ***support*** |
|---|---|---|---|---|
| **I** | 0.74 | 1.00 | 0.85 | 65 |
| **II** | 0 | 0 | 0 | 9 |
| **III** | 0 | 0 | 0 | 7 |
| **IV** | 0 | 0 | 0 | 5 |
| **V** | 0 | 0 | 0 | 2 |
| **Acurácia** | | | 0.74 | 88 |
| **Média** | 0.15 | 0.20 | 0.17 | 88 |
| **Ponderada** | 0.55 | 0.74 | 0.63 | 88 |

Figura 10 - Resultados Naive Bayes

Esses resultados vieram da execução da função `def resultado_modelo()`. É possível ver a Figura 11 com o resultado na execução da mesma no shell do python.Também é mostrada a Matriz de Confusão do Naive Bayes, chamada na mesma função.

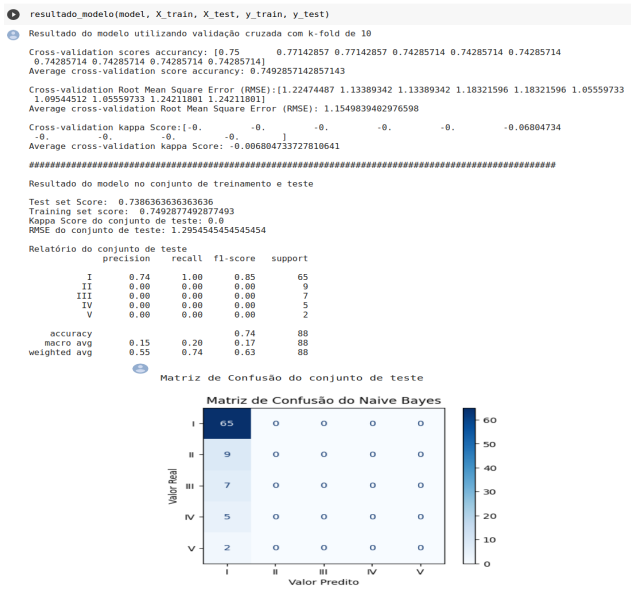


Figura 11 - Função `def resultado_modelo(....)`

Foi feito o score de importância entre as variáveis e o resultado pode ser visto na Figura 12. Mais análises a respeito dos dados são descritas na Seção 2.4, onde foram apresentados alguns padrões importantes em termos de relações de influência entre as variáveis. As chamadas dos métodos python podem ser vistas no Anexo 1.

| | I | II | III | IV | V |
|---|---|---|---|---|---|
| **Risco Critico** | 48.25 | 50.71 | 45.59 | 47.35 | 43.78 |
| **Local** | 16.20 | 12.84 | 13.56 | 12.59 | 12.44 |
| **Week_day** | 11.74 | 9.73 | 12.37 | 13.02 | 12.94 |
| **Potential Accident Level** | 11.49 | 14.53 | 16.10 | 16.02 | 18.41 |
| **Genre** | 3.92 | 3.76 | 4.07 | 3.86 | 3.98 |
| **Industry Sector** | 3.15 | 2.85 | 4.07 | 3.15 | 2.99 |
| **Employee ou Terceiro** | 3.05 | 3.50 | 2.71 | 2.86 | 4.48 |
| **Countries** | 2.20 | 2.08 | 1.53 | 1.14 | 1.00 |

Figura 12 - Ranking de importância das variáveis

- O Classificador Estruturado

É possível observar as métricas do uso *ten-fold* e aquelas aplicadas aos conjuntos de treinamento e teste. A Figura 13, da mesma forma que para o Naive Bayes, mostra esses dados em duas tabelas.

Foi implementada uma função em python que produziu os dados do conjunto de parâmetros necessários à comparação e a equipe tentou encontrar o modelo bayesiano ótimo para o classificador. Foram implementadas com poucos comandos a *Hill Climbing Search* e a métrica *K2*, conforme o código-fonte.

Após isso foi implementada a sequência de Treinamento e Teste do modelo, usando o k-fold com dez. As linhas do python mostram as juntadas do par X e Y (entrada e saída) para os conjuntos de treinamento e de teste, mostram o *bnlearn* possibilitando o treinamento do modelo e a criação de um classificador Bayesiano para cada conjunto.

A Figura 13 mostra a função de geração dos dados de comparação executada e chegou-se no resultado mostrado, onde é possível perceber o shell do python e os dados retirados e tabulados. A imagem mostra o Shell, duas tabelas e a Matriz de confusão.

```
Resultado do modelo utilizando validação cruzada com k-fold de 10

Cross-validation scores accurancy:[0.8571428571428571, 0.7142857142857143, 0.8, 0.7428571428571429, 0.7714285714285715, 0.8, 0.7142857142857143, 0.6, 0.7714285714285715, 0.857
Average cross-validation score accurancy: 0.7628571428571428

Cross-validation kappa Score:[0.5884269662921348, 0.1841491841491841, -0.04255319148936176, 0.19642857142857142, 0.0572390572390572, 0.2620481927710844, 0.1802570694087403,
Average cross-validation kappa Score: 0.20624635845720602

Cross-validation RMSE:[0.2285714285714286, 0.8571428571428571, 1.6, 1.2857142857142858, 1.7428571428571429, 0.2857142857142857, 1.6571428571428573, 2.1142857142857143, 0.8571
Average cross-validation RMSE: 1.1600000000000001

##########################################################################################################

Resultado do modelo no conjunto de treinamento e teste

Test set Score:  0.7613636363636364
Training set score:  0.7663817663817664

Kappa Score do conjunto de teste:0.2817725029148856

RMSE conjunto de teste:0.9772727272727273

Relatório do conjunto de teste
              precision    recall  f1-score   support

           I       0.78      0.95      0.86        65
          II       0.00      0.00      0.00         9
         III       0.00      0.00      0.00         7
          IV       0.50      0.60      0.55         5
           V       0.67      1.00      0.80         2

    accuracy                           0.76        88
   macro avg       0.39      0.51      0.44        88
weighted avg       0.62      0.76      0.69        88


(array([0.5, 1.5, 2.5, 3.5, 4.5]),
 [Text(0, 0.5, 'I'),
  Text(0, 1.5, 'II'),
  Text(0, 2.5, 'III'),
  Text(0, 3.5, 'IV'),
  Text(0, 4.5, 'V')])
```

| Métricas com 10 K-fold | |
|---|---|
| Acurácia Média | 0.7628 |
| Kappa Médio | 0.20624 |
| RMSE Médio | 1.16 |
| **Métricas no Conjunto de Treinamento e Teste** | |
| Acurácia no Treinamento | 0.7663 |
| Acurácia no Teste | 0.7613 |
| Kappa | 0.2817 |
| RMSE | 0.9772 |

| | *precision* | *recall* | *f1-score* | *support* |
|---|---|---|---|---|
| **I** | 0.78 | 0.95 | 0.86 | 65 |
| **II** | 0 | 0 | 0 | 9 |
| **III** | 0 | 0 | 0 | 7 |
| **IV** | 0.50 | 0.60 | 0.55 | 5 |
| **V** | 0.67 | 1.00 | 0.80 | 2 |
| **Acurácia** | | | 0.76 | 88 |
| **Média** | 0.39 | 0.51 | 0.44 | 88 |
| **Ponderada** | 0.62 | 0.76 | 0.69 | 88 |

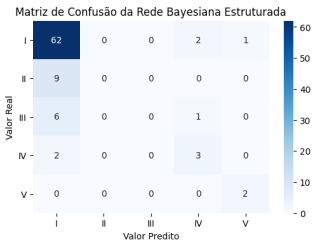


Figura 13 - Resultados do Classificador Estruturado

Neste ponto é possível perceber que o Classificador Estruturado teve um melhor desempenho do que o Classificador Bayesiano. Tanto no treinamento quanto no Teste, os valores de Acurácia foram bem maiores.

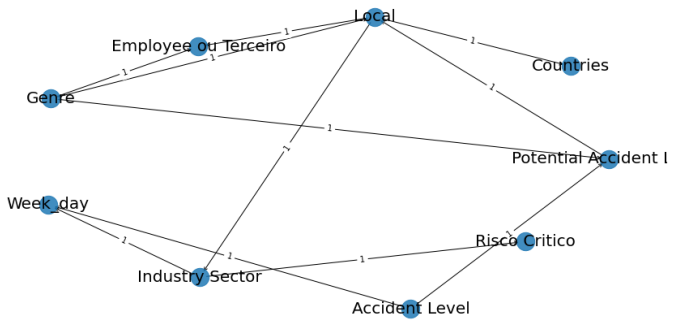


Figura 14 - Modelo gráfico da Rede Bayesiana

Foram encontrados percentuais de *precision, recall* e *f-score* que não estavam presentes no Naive Bayes com relação aos níveis de acidente e houve 65 acertos no primeiro e 67 no segundo. Percebeu-se que o Naive Bayes é adaptável para o problema, mas o desempenho dele faz compensar o trabalho de implementar um classificador estruturado. A Figura 14 mostra o modelo ótimo de forma gráfica

## 2.3. Melhorias possíveis para o classificador

Foram realizadas modificações no dataset no item 2.1, como agrupamento dos *Riscos Críticos* comuns. A equipe acreditou que, no caso de um problema real, deveria haver "acordos" de considerações semânticas a respeito dos Riscos Críticos, com o intuito de montar agrupamentos mais abrangentes de tipos com significados semelhantes.

Seria somente uma questão de interpretação desses tipos e de um consenso no entendimento comum para quem iria utilizar os resultados do experimento para poder realizar mais agrupamentos.

## 2.4. Padrões relevantes existentes nos dados

Foi realizado o teste com a Árvore de Decisão e chegou-se à matriz de confusão da Figura 15. Essa figura mostra uma parte do grafo da árvore e a matriz.

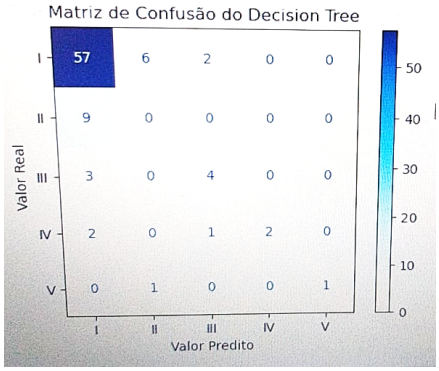


Figura 15 - Resultados do *Decision Tree*

Foram feitas análises de relações entre as variáveis. A Figura 16 mostra uma sequência de gráficos que tornam amigável a visualização das relações entre as variáveis.

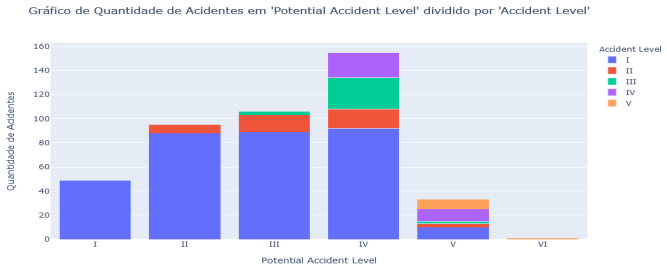


Figura 16 - Gráficos

É possível perceber que o *Accident_Level* com valor "I" é o mais presente em quantidade de acidentes, o que é possível também enxergar no segundo gráfico, em quase todos os tipos de risco crítico ele é o mais presente.

No terceiro gráfico também há essa tendência do nível I de acidente para todos os locais, a quantidade de acidentes por dia da semana mostra a quarta-feira como maior número e nesse gráfico é possível ver que o setor da indústria que mais pesa nesse número é a mineração (*Mining)*. Ainda nos dias da semana, o nível I de acidente também seguiu a tendência de mais presença.

Quando são confrontados a quantidade de acidentes e os gêneros masculino e feminino, é possível perceber, por cada nível de acidente, as quantidades de acidentados de cada tipo, os homens acidentam-se mais que as mulheres.

## 3.0. Conclusão e Considerações

Este estudo permitiu colocar em prática as capacidades ensinadas em aula sobre como analisar conjuntos de dados desconhecidos. A equipe foi capaz de preparar o conjunto fornecido usando bibliotecas simples da linguagem Python e aplicando técnicas que permitiram tornar o dataset um conjunto de dados capaz de ser submetido aos algoritmos aprendidos.

Apesar de ser uma linguagem nova para ⅔ da equipe, a curva de aprendizado foi vencida sem muita dificuldade, acreditamos que devido à aula onde foi apresentado o primeiro notebook jupyter do colab, de autoria do Professor. Foram descobertas outras bibliotecas que seriam capazes de pe

## 3. Implementação

Descrição da implementação, comentários eventualmente necessários para a execução do projeto e Código do Projeto

## 4. Referências

[1] R. G. Wiley, *ELINT : the interception and analysis of radar signals*. Artech House, 2006.