Análise de Ocorrências da Guarda Municipal utilizando Ciência de Dados

Carlos Humberto Lopes Costa e Láercio Silva de Campos Junior
Universidade Federal do Paraná (UFPR)
Disciplina de Ciência de Dados para Segurança

I. INTRODUÇÃO

Esse trabalho foi elaborado como Projeto Final para a disciplina "*Ciência de Dados para Segurança*", professor André Grégio (Dtinf/UFPR). O objetivo desse trabalho é aplicar as técnicas e ferramentas de Ciência de Dados para explorar um conjunto de dados específico.

Foi escolhido para análise o conjunto de dados de ocorrências atendidas pela Guarda Municipal de Curitiba/PR. Esse trabalho abordará todas as etapas do processo de Ciência de Dados – *Obtenção de Dados, Limpeza, Exploração, Modelagem e Interpretação*. Esse trabalho aplicou ainda os algoritmos de KNN, Random Forest e Support Vector Machine no conjunto de dados.

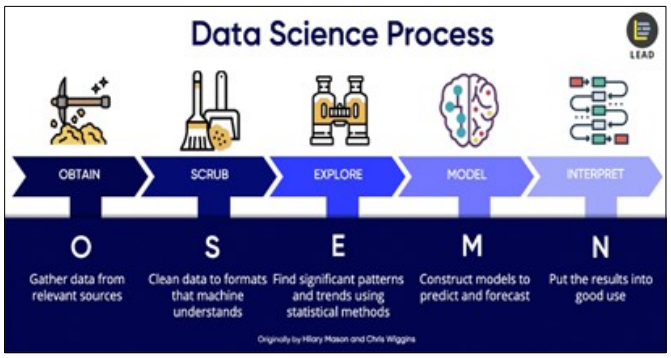


Fig. 1 - https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492

II. ANÁLISE DOS DADOS

a) Fonte de Dados

O conjunto de dados foi obtido do portal de dados abertos da Prefeitura de Curitiba/PR[1]. Esse dataset conta com dados do sistema *"SiGesGuarda"* - sistema contendo os dados das ocorrências atendidas pela Guarda Municipal de Curitiba/PR. O dataset está no formato CSV e o espectro temporal é de 2009 até 01/02/2021. O dataset fornece informações como: *categoria, subcategoria, data, hora, bairro e origem da ocorrência*.

b) Pré-Processamento

O dataset foi pré-processamento para remoção de registros vazios, remoção de atributos desnecessários, correção de inconsistências e criação de novos atributos.

Foi criado o *script* Python "*PreProcessamento.py*" para realização dessa tarefa, sendo o script disponibilizado o GitHub "*https://github.com/chlcosta/CDadosSeg/tree/main/TrabalhoFinal*".

Originalmente o dataset possui 35 colunas/atributos, após o pré-processamento ficou com 11 atributos, apresentados na Tabela 1 a seguir.

Tabela 1 – Colunas/Atributos do Dataset	
1	OC_ANO – *Ano da Ocorrência*
2015, 2016, 2017, 2018, 2019, 2020	
2	OC_MES – *Mês da Ocorrência*
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	
3	OC_DIA_SEMANA – *Dia da Semana (Numérico)*
1, 2, 3, 4, 5, 6, 7	
4	OC_DIA_SEMANA_TXT – *Dia da Semana (Textual)*
1-DOMINGO', '2-SEGUNDA', '3-TERCA', '4-QUARTA', '5-QUINTA', '6-SEXTA', '7-SABADO'	
5	OC_DIA – *Dia do mês*
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31	
6	OC_PERIODO_DIA – *Período do Dia (Numérico)*
1, 2, 3, 4	
7	OC_PERIODO_DIA_TXT – *Período do Dia (Textual)*
'1-MANHA', '2-TARDE', '3-NOITE', '4-MADRUGADA'	
8	OC_BAIRRO – *Bairro da Ocorrência (Numérico)*
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, ..., 71, 72, 73, 74	
9	OC_BAIRRO_TXT – *Bairro da Ocorrência (Textual)*
'ABRANCHES', 'ÁGUA VERDE', 'AHÚ', 'ALTO BOQUEIRÃO', 'ALTO DA GLÓRIA', 'ALTO DA RUA XV', 'ATUBA', 'AUGUSTA', 'BACACHERI', 'BAIRRO ALTO', 'BARREIRINHA', 'BATEL'...	
10	OC_SUBCATEGORIA – *Tipo da Ocorrência (Numérico)*
1, 2, 3, 4, 5, 6, 7, 8, 9, 10	
11	OC_SUBCATEGORIA_TXT – *Tipo da Ocorrência (Textual)*
'Arrombamento', 'Cão solto em via pública', 'Desordem', 'Disparo de Alarme (violação)', 'Estacionamento irregular', 'Invasão de equipamento/patrimônio público', 'Pichação', 'Transporte Coletivo', 'Uso de substância ilícita', 'Vandalismo'	

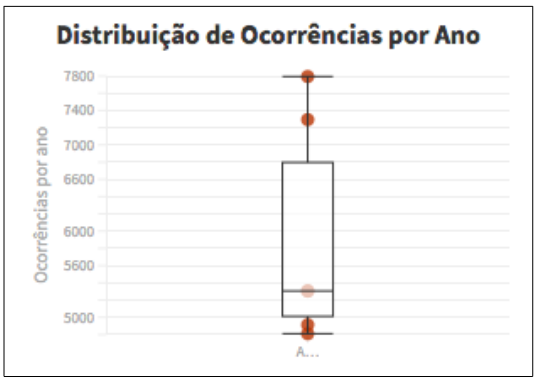
O atributo de período do dia (OC_PERIODO_DIA_TXT) foi criado utilizando as seguintes referências, A madrugada vai da zero hora às 6h. A manhã, das 6h às 12h (ou ao meio-dia). A tarde, das 12h às 18h. A noite, das 18h às 24h (ou meia-noite).

Para o estudo optou-se por definir o intervalo dos últimos 6 anos para análise (2020-2015) e as 10 ocorrências de maior frequência no período, totalizando 35.424 registros.

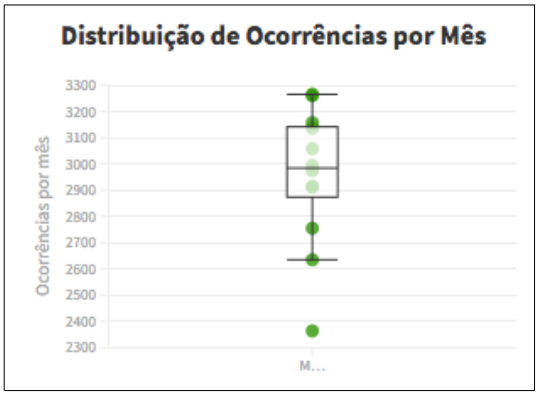
1 https://www.curitiba.pr.gov.br/dadosabertos/

c) Análise dos Dados

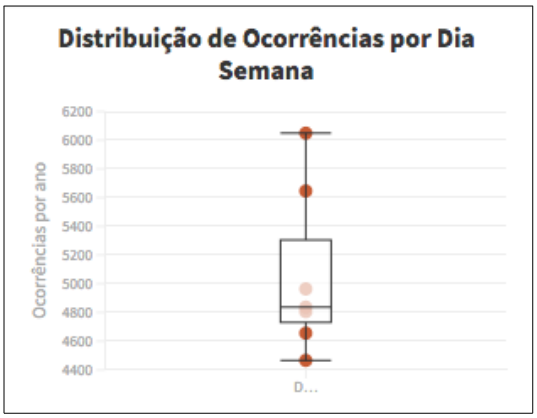
Na sequência são apresentados os gráficos de distribuição do dataset baseado no Ano, Mês e Dia da Ocorrência.



(a)



(b)



(c)

Fig. 2 – Distribuições por (a) Ano, (b) Mês e (c) Dia da Semana

	Ano	Mês	Dia Semana
Terceiro Quartil	6.797	3.141	5.305
Mediana	5.306	2.984	4.837
Primeiro Quartil	5.011	2.873	4.731

Fig. 3 – Número de Ocorrências por Ano

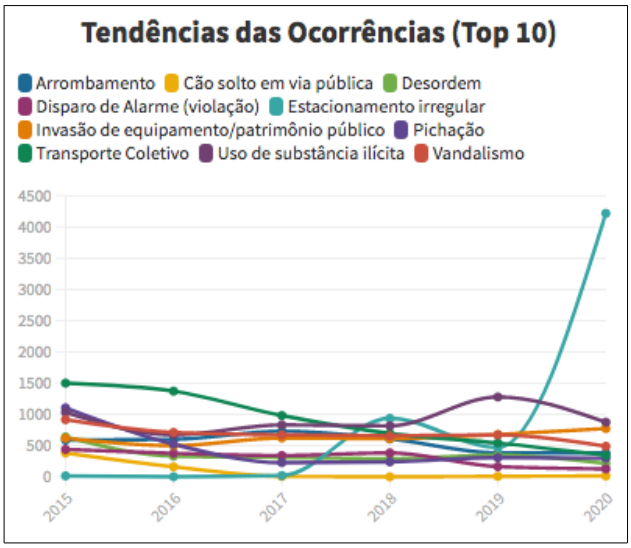


Fig. 4 – Tendências das "Top 10" ocorrências

A partir da análise das principais ocorrências pode-se observar um aumento de quase 1.000% nas ocorrências de "Estacionamento Irregular" entre 2019 e 2020. Com exceção da ocorrência da "Estacionamento Irregular" as demais ocorrências sofreram uma leve queda de 2019 para 2020, possivelmente devido ao efeito da pandemia do COVID-19. No entanto, o "Uso de Substância Ilícita" teve uma queda aproximada de 31% no mesmo período.

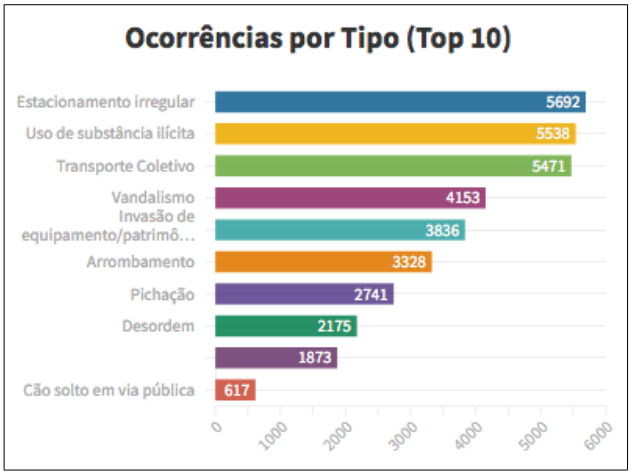


Fig. 5 – Número de ocorrências por Tipo

Observou-se através das Figuras 5 e 6 que a distribuição da quantidade de ocorrências é bastante irregular entre os tipos de ocorrências e entre os bairros.

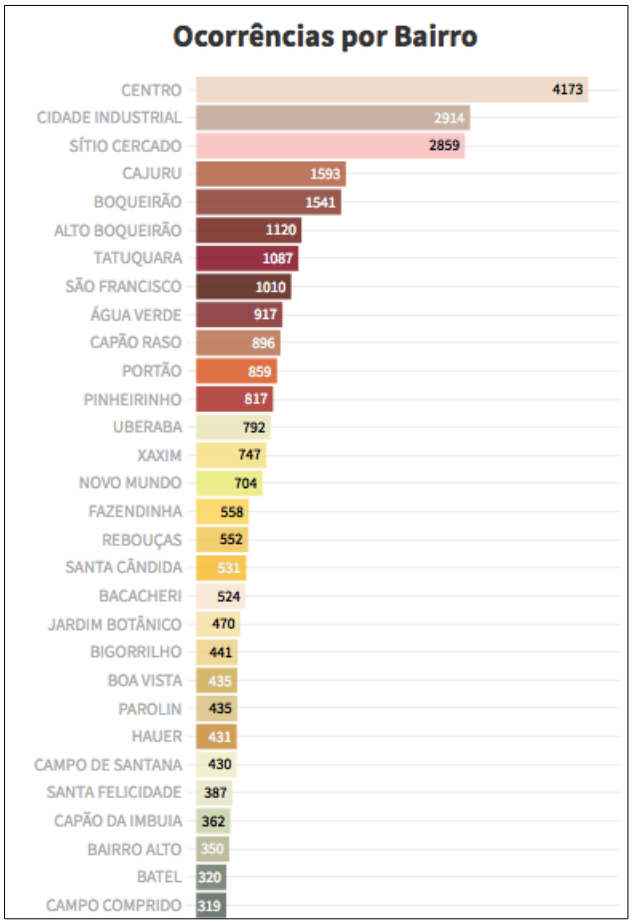


Fig. 6 – Número de ocorrências por Bairro

Fig. 7 – Número de ocorrências por Período do Dia

Conforme a Figura 7, as ocorrências atendidas pela Guarda Municipal de Curitiba, para os "Top 10" tipos de ocorrências, seguem a tendência de aumento ao longo do dia, no entanto, com redução considerável no período da madrugada.

Para essa etapa de exploração dos dados foi criado o *script* Python "*ExploracaoDados.py*", que também foi disponibilizado no endereço do GitHub informado.

III. CONSTRUÇÃO DO MODELO

Considerando que o propósito do presente trabalho é a classificação dos tipos de ocorrências atendidas pela Guarda Municipal de Curitiba/PR, utilizou-se para construção dos modelos os algoritmos de classificação K-Nearest Neighbour (KNN), Random Forests e Support Vector Machine (SVM). Cada algoritmo tem suas próprias vantagens e desvantagens em termos de complexidade, precisão e tempo de treinamento, podendo prover diferentes resultados para um mesmo conjunto de dados de entrada.

Antes da construção dos modelos, os dados precisaram ser alterados para formatos compatíveis com os modelos. Os atributos categóricos foram convertidos em atributos numéricos com um único ID. Todos os tipos de ocorrências e bairros possuem um diferente ID. Os bairros são representados através de 74 IDs e os tipos de ocorrências através de 10 IDs. A relação entre os atributos textuais e numéricos podem ser visualizados na Tabela 1, apresentada anteriormente.

A etapa de pré-processamento realizou a divisão do dataset em conjunto de treinamento e teste na proporção "80/20", respectivamente. Os algoritmos foram aplicados sobre esses mesmos conjuntos de dados. Realizou-se também a validação cruzada (*Cross-Validation*) com 5 pastas para cada algoritmo. A validação cruzada previne o problema de *overfitting* e assegura que a predição do modelo possui performance satisfatória para dados ainda não vistos.

Antes da aplicação de cada modelo, o dataset foi novamente divido na proporção "80/20".

Para aplicação dos modelos no dataset foi criado o *script* Python "*Modelos.py*", que está disponível no endereço do GitHub informado.

a) K-Nearest Neighbour (KNN)

O KNN foi aplicado com os parâmetros 1, 3 e 5, sendo os resultados coletados e apresentados abaixo:

Tabela 2 – Resultados KNN (80%)

	K =1	**K = 3**	**K = 5**
Precisão	0.220	0.220	0.221
Acurácia	0.268	0.290	0.304
Erro	2.616	2.633	2.586

Abaixo é apresentada a Matriz de Confusão aplicando K = 1. E na sequência as Curvas ROC das ocorrências tipo 1 e tipo 10, mostrando a variação na curva.

Matriz de Confusão

```
[[214  19  63  72 102  75  87  45 131  46]
 [ 19  13   6   9  13   7  10   4  14   1]
 [ 46   4  46  31  26  43  36  19  23  15]
 [ 65  13  23  50  57  53  46  37  56  31]
 [ 97   9  39  73 105  81  82  44  99  49]
 [ 72   4  40  41  68 117  59  32  48  39]
 [ 80   5  33  41  73  64  91  42  81  95]
 [ 50   5  24  42  47  31  39  40  53  28]
 [128  23  22  68 101  62  76  37 298  98]
 [ 45   3  24  23  52  21  77  35 100 543]]
```

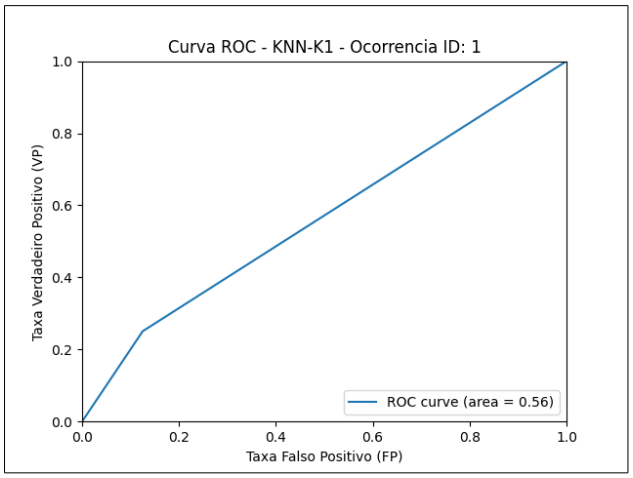


Fig. 8 – Curva ROC – KNN (K=1) – Ocorrência Tipo 1

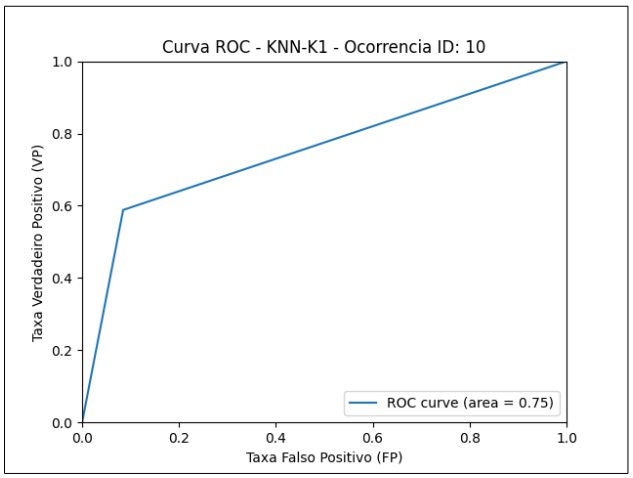


Fig. 9 – Curva ROC – KNN (K=1) – Ocorrência Tipo 10

Observou-se que a precisão geral é bastante baixa para o caso em questão. No entanto, a precisão pode ser melhor para determinado tipo de ocorrência.

Na sequência são apresentadas as Curvas ROC da ocorrência tipo 1 e 10 utilizando KNN (K=1) com Validação Cruzada com 5 pastas (k-fold = 5).

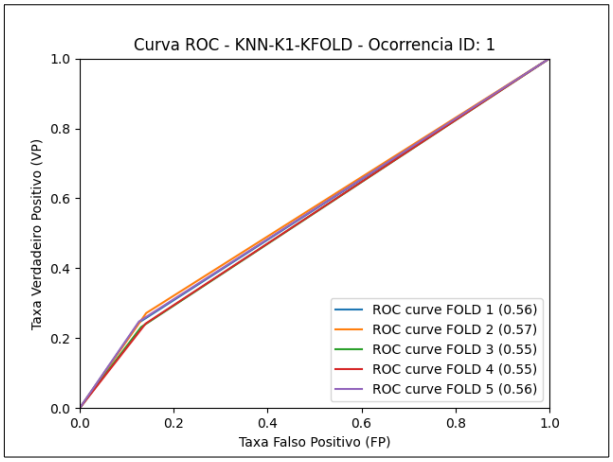


Fig. 10 – Curva ROC – KNN (K=1) com Validação Cruzada (K-fold=5) – Ocorrência Tipo 1

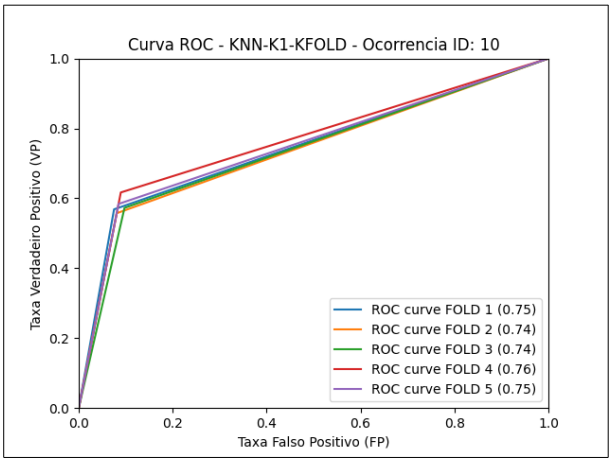


Fig. 11 – Curva ROC – KNN (K=1) com Validação Cruzada (K-fold=5) – Ocorrência Tipo 10

O log completo da execução do modelo, com as precisões gerais, precisões utilizando validação cruzada e matrizes de confusão pode ser visto no arquivo "*log-Modelos-80.txt*" no GitHub do trabalho.

Em seguida, foi realizado o teste com os outros 20% dos dados do dataset e se obteve uma precisão bastante próxima, conforme observado na tabela 3.

Tabela 3 – Resultados KNN (20%)

	K =1	**K = 3**	**K = 5**
Precisão	0.219	0.217	0.193
Acurácia	0.266	0.279	0.289
Erro	2.631	2.666	2.664

b) RandomForests

Aplicou-se também o modelo RandomForest, que utiliza o conceito de árvores de decisão. O algoritmo cria uma estrutura similar a um fluxograma, com "nós" onde uma condição é verificada, e se atendida o fluxo segue por um ramo, caso contrário, por outro, sempre levando ao próximo nó, até a finalização da árvore.

O algoritmo foi parametrizado com 10 árvores, antes de tomar uma votação ou fazer uma média de predições. A precisão, erro e a Matriz de Confusão são apresentados abaixo.

Tabela 4 – Resultados RandomForests (80%)	
	n_estimators = 10
Precisão	0.270
Acurácia	0.333
Erro	2.426

Matriz de Confusão

```
[[351  16  36  51  90  63  59  35 101  52]
 [ 32  12   4   9  10   6   8   2  12   1]
 [ 46   7  71  19  25  45  41   6   9  20]
 [ 89   8  25  82  52  30  37  25  54  29]
 [120  11  32  68 129  83  69  28  96  42]
 [ 91  10  48  30  78 109  61  15  37  41]
 [ 99   6  31  35  83  61 103  26  63  98]
 [ 82   5  12  38  54  25  25  35  52  31]
 [145  24  12  58  95  36  55  33 371  84]
 [ 39   2  21  13  40  17  57  17  94 623]]
```

Apesar a precisão se manter baixa, para o caso em análise, já se mostrou melhor que a classificação com KNN.

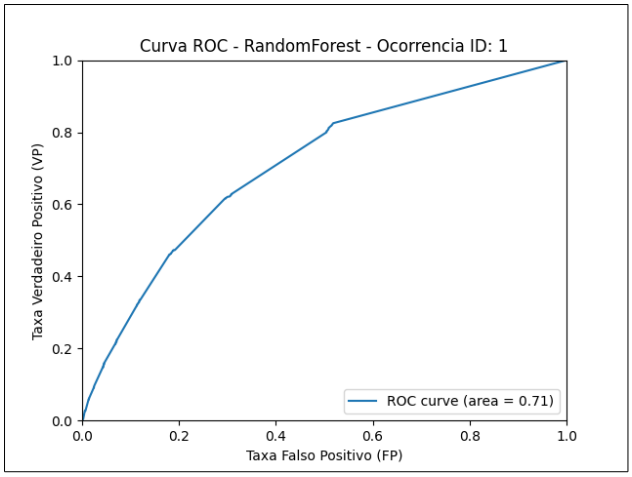


Fig. 12 – Curva ROC – RandomForest – Ocorrência Tipo 1

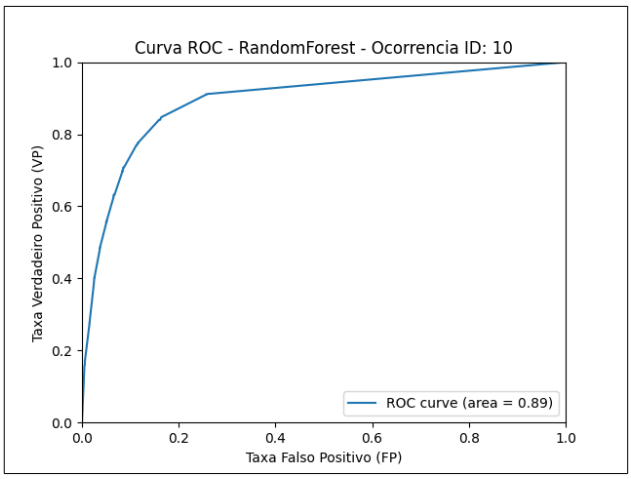


Fig. 13 – Curva ROC – RandomForest – Ocorrência Tipo 10

Assim como no algoritmo KNN, observamos forte discrepâncias nas precisões dos tipos de Ocorrências.

Na sequência são apresentadas as Curvas ROC da ocorrência tipo 1 e 10 utilizando RandomForest com Validação Cruzada com 5 pastas (k-fold = 5).

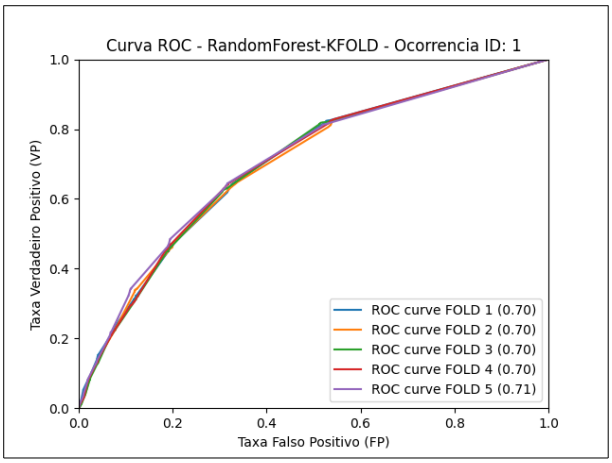


Fig. 14 – Curva ROC – RandomForests com Validação Cruzada (K-fold=5) – Ocorrência Tipo 1

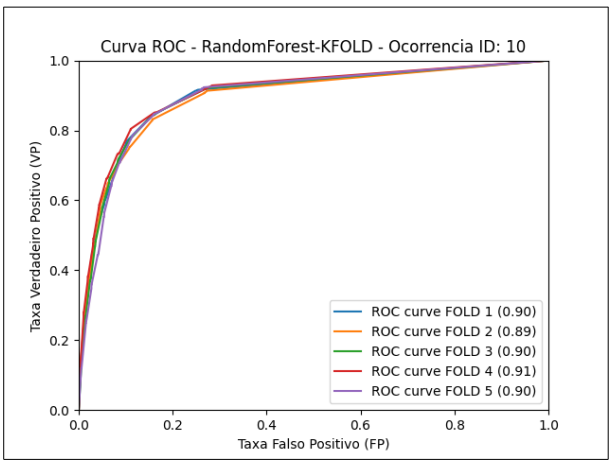


Fig. 15 – Curva ROC – RandomForests com Validação Cruzada (K-fold=5) – Ocorrência Tipo 10

Em seguida, foi realizado o teste com os outros 20% dos dados do dataset e se obteve uma precisão bastante próxima, conforme observado na tabela 5.

Tabela 5 – Resultados RandomForests (20%)	
	n_estimators = 10
Precisão	0.267
Acurácia	0.327
Erro	2.468

c) Support Vector Machine (SVM)

O último algoritmo analisado foi o Support Vector Machines (SVM). O algoritmo foi utilizado com o parâmetro Kernel = "Linear'.

Os resultados são apresentados abaixo:

Tabela 6 – Resultados SVM (80%)	
	Kernel = Linear
Precisão	0.144
Acurácia	0.212
Erro	3.103

Matriz de Confusão

```
[[211  94 118  47  32   2   1 184  77  88]
 [ 29   2  13   0  15   0   2  17  13   5]
 [ 63  64  29  10   7   0   1  74  10  31]
 [106  48  54  17   8   0   0 107  30  61]
 [148  65  84  37  30   4   3 145  55 107]
 [ 96  60  64  42  24   0   5 120  45  64]
 [122  52  67  38  21   1   3 113  35 153]
 [ 87  29  33  22  15   1   0  84  25  63]
 [156  34 153  29  78   5   4 109 131 214]
 [ 44  11  71  24  12   0   2  59   3 697]]
```

Esse modelo apresentou a pior precisão dentre os demais modelos, para todos os tipos de ocorrências.

Abaixo são apresentadas as Curvas ROC para o modelo.

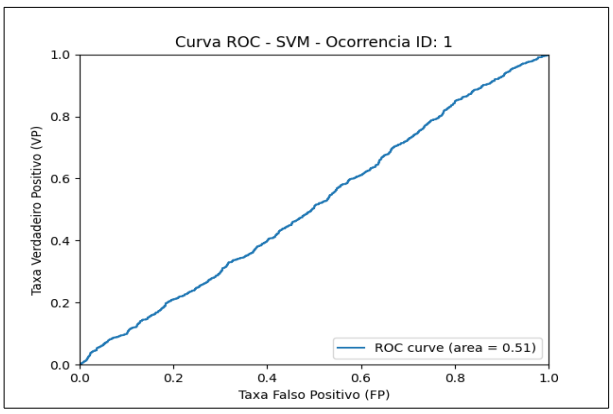


Fig. 16 – Curva ROC – SVM – Ocorrência Tipo 1

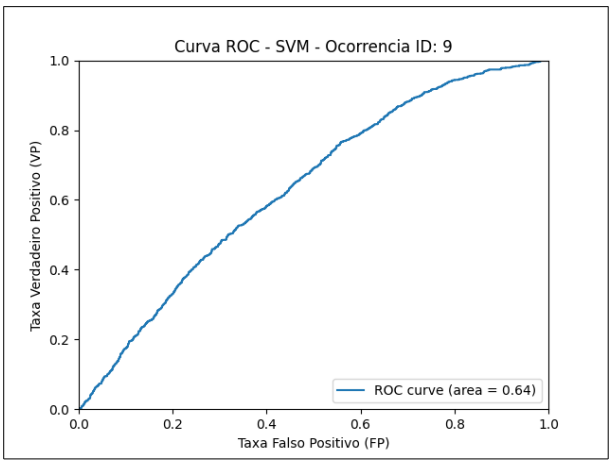


Fig. 17 – Curva ROC – SVM – Ocorrência Tipo 10

Na sequência são apresentadas as Curvas ROC da ocorrência tipo 1 e 10 utilizando SVM com Validação Cruzada com 5 pastas (k-fold = 5).

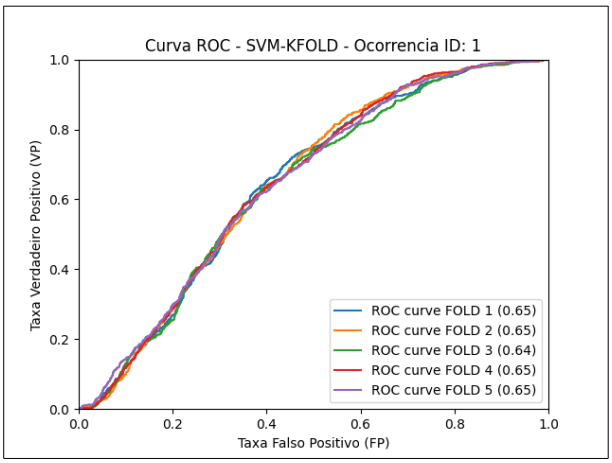


Fig. 18 – Curva ROC – SVM com Validação Cruzada] (K-fold=5) – Ocorrência Tipo 1

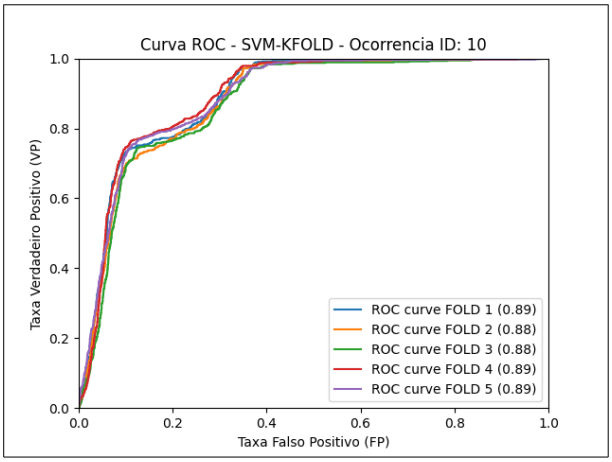


Fig. 19 – Curva ROC – SVM com Validação Cruzada] (K-fold=5) – Ocorrência Tipo 10

Em seguida, foi realizado o teste com os outros 20% dos dados do dataset e se obteve uma precisão bastante próxima, conforme observado na tabela 7.

Tabela 7 – Resultados SVM (20%)	
	Kernel = Linear
Precisão	0.137
Acurácia	0.219
Erro	2.933

IV. CONCLUSÃO

Nesta trabalho foram utilizados dados de atendimento de ocorrências da Guarda Municipal de Curitiba/PR nos últimos 6 anos (2015 a 2020) e com as 10 ocorrências com maior frequência no período. Os dados foram divididos na proporção “80/20”, criando conjunto de treinamento e teste. As precisões dos modelos foram bastante baixas para o conjunto de dados, o melhor resultado obtido foi com RandomForest que obteve precisão de 27% e o pior resultado foi com SVM, que obteve precisão de 14%. O modelo KNN apresentou precisão muito parecidas para K=1, 3 e 5, entre 19% e 21%. Embora este modelo tenha baixa precisão como modelo de previsão, ele fornece uma estrutura preliminar para análises futuras.

V. REFERÊNCIAS

[1] Ceschin, F.; Oliveira, L. E. S.; Grégio, A. R. A. *Aprendizado de Máquina para Segurança: Algoritmos e Aplicações*. Capítulo 2 do Livro de Minicursos do XIX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais, 2019. https://sbseg2019.ime.usp.br/minicursos.pdf