

ITALLO MEDEIROS DIAS

PORTIFÓLIO DE CASES DE MACHINE LEARNING

Case: Score Preditivo de Mudanças Ofensoras

Tipo de modelo: Classificação

Motivador: Um indicador da Diretoria de Operações de TI mostra que 5% de todas as mudanças executadas mensalmente (cerca de 10 mil em ago/2022) são finalizadas com algum tipo de problema. Este problema pode ser um simples desvio com relação ao planejado ou pode ser um incidente que impacta os serviços e produtos do cliente.

Proposta: Criar um modelo de classificação que seja capaz de identificar se uma mudança tem risco de ser ofensora de um incidente. Como resultado, o modelo contribui para um processo que ajuda a mitigar a execução de mudanças arriscadas.

Saída do modelo: Um valor de probabilidade entre 0 e 1, sendo 1 a classe de interesse (mudanças com desvio ou incidente). Os valores de saída são distribuídos em grupos de criticidade entre baixo e crítico.

Variáveis do modelo: Variáveis de cadastro da mudança e variáveis relacionados ao time que executará a mudança:

Exemplo:

- Categoria
- Subcategoria
- Plataforma
- Horário de Início
- Reputação da Squad
- Histórico do IC afetado
- Tipo da pipeline
- Tipo do produto (batch, online)
- Status de IC crítico ou Backlevel
- Status de execução em Freezing

Técnicas utilizadas:

Balanceamento: Oversampling (ImbLearn)

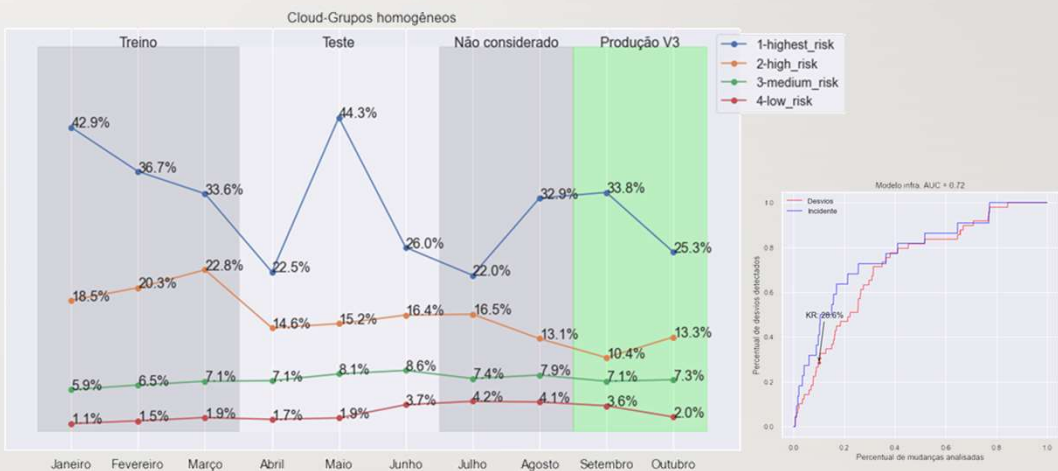
Modelo: Random Forest

Tunning: Boruta

Métricas de acompanhamento:

- Grupos homogêneos por criticidade
- Precision x Recall
- AUC

Resultados:



Outras informações: Modelo deployado em ambiente produtivo. Acompanhamento semanal dos resultados e desenvolvimento constante de novas versões. Dentre as melhorias buscadas, pode-se citar a busca de novas variáveis explicativas e treino de novos modelos.

Case: Detecção de Pessoas em câmera de CFTV

Tipo de modelo: Classificação de imagens

Motivador: Um sistema tradicional de CFTV possui mecanismos para detectar movimento, porém não difere tipos de movimentos.

Proposta: Criar um modelo de classificação de imagens que seja capaz de identificar a presença de uma pessoa na imagem. As imagens foram obtidas por uma integração com servidor FTP

Saída do modelo: Um valor de probabilidade entre 0 e 1, sendo 1 a classe de interesse (pessoa detectada na imagem). Quando for identificada uma pessoa, uma notificação será enviada para o e-mail do responsável.

Variáveis do modelo: São coletadas imagens de uma determinada câmera a cada 1 segundo. A imagem só é submetida para ao modelo quando um movimento é detectado, portanto o modelo deve identificar se trata-se de uma pessoa ou não.



Técnicas utilizadas:

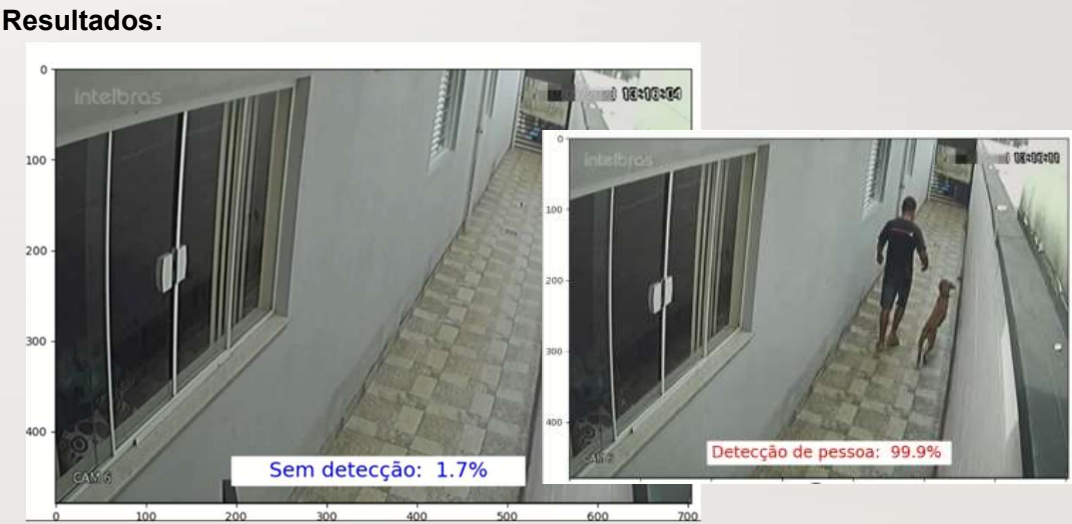
Modelo: Tensorflow – Rede Neural Convolucional (CNN)

Utilizadas 233 imagens, sendo 87 da classe de interesse.

Métricas de acompanhamento:

- Recall
- Acurácia: 88%
- Precisão: 100%

Após submeter para produção, foi identificado horários do dia no qual Falsos Positivos eram identificados.



Outras informações: Modelo deployado em EC2 (AWS). Um servidor FTP coletava as imagens. A aplicação Python com o modelo analisava as imagens e, em caso de detecção de pessoas, a imagem é salva em bucket S3 e enviada por e-mail. Nesta etapa, identifiquei a necessidade de treinar o modelo com mais fotos, pois em certos horários do dia, a incidência do sol provocava falsos positivismo. Github: https://github.com/itallond/CNN_detect_people_cams

Case: Aplicação de agrupamento em pixels de imagens

Tipo de modelo: Agrupamento

Motivador: Motivação meramente experimental e de aprendizado.

Proposta: Avaliar como imagens coloridas em RGB produzem clusters quando agrupamos sua escala de cores em k clusters.

Saída do modelo: Ao agrupar as todas as cores da imagens em K clusters, podemos ver o efeito produzido em, por exemplo 2 a 9 clusters. Foi utilizado o centroide (ou valor médio do cluster) como referência para definir a cor que representa cada cluster. Com isso, colorimos novamente a imagem com 2 a 9 cores, baseada no agrupamento. Técnicas diferentes (k-means, hierárquico, GMM) produzem resultados diferentes.

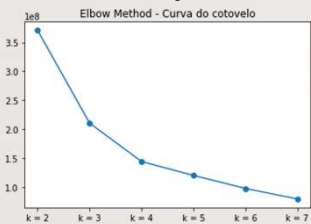
Variáveis do modelo: Qualquer imagem RGB, formando uma matriz composta pelas dimensões da imagem mais uma terceira dimensão da escala RGB. Foto exemplo:



Técnicas utilizadas:

Modelo: K-Means

Métricas de acompanhamento:



Resultados:



Outras informações: Github: https://github.com/itallond/image_clustering