Universidade Federal de Minas Gerais
Departamento de Ciência da Computação

DCC057 Mineração de Dados

Trabalho Prático 3 — Proposta

Alexander Thomas Mol Holmquist

08 de março de 2021

1 Introdução

A detecção de um câncer de pele antes da ocorrência de metástase é de fundamental importância para a saúde de um paciente [1, 2]. Porém, atualmente a detecção de tais casos só pode ser assegurada por médicos especializados em dermatologia. A possibilidade de fácil detecção de cânceres de pele malignos poderia diminuir significativamente a taxa de mortalidade, mas ainda não é uma realidade.

Uma aplicação de redes neurais convolutivas, desenvolvida em Stanford [3], afirma ser tão eficaz quanto dermatologistas, na classificação de diferentes cânceres de pele. Este projeto propõe uma reprodução de tal experimento em pequena escala, para explorar a dificuldade de classificar figuras de cânceres de pele através de um pequeno banco de dados. Assim, distingue-se os objetivos abaixo:

Objetivos de negócio:

1. (principal) Encontrar modelo que classifica as figuras com alta precisão ($F-SCORE \geq 95\%$). Note que esta precisão não seria suficiente para aplicação na prática, por conta do caráter crítico deste tipo de sistema.

2. Construir uma imagem que melhor represente características de um câncer benigno, e uma imagem que melhor represente cânceres malignos. O critério a ser utilizado para determinar se houve sucesso é a descrição especialista destas duas classes.

2 Fonte de dados

Existe somente uma fonte de dados que está sob consideração inicialmente, e vem da plataforma Kaggle [4]. A origem dos dados é a International Skin Imaging Collaboration (ISIC) [5]. Os dados escolhidos são um conjunto selecionado das imagens providas pela ISIC. No total contém 1600 fotos de regiões de desenvolvimento de câncer de pele, cada uma com resolução 224 x 224 pixels.

Os dados já vieram divididos em conjunto de teste e conjunto de treinamento, em uma razão 1/10. Por hora, a pretenção é deixar como está, mas pode ser que se decida alterar tal razão. Nesse caso, bastar integrar os dois conjuntos de fotos, e fazer uma nova amostra aleatória do novo tamanho. É suposto, no presente projeto, que os dados da plataforma Kaggle foram selecionados da origem de maneira íntegra, e sem viés.

3 Plano de projeto

A seguir é dado o plano inicial do projeto, baseado fortemente na metodologia CRISP-DM, que será utilizada durante todo o projeto. Como ela recomenda, este plano inicial deve ser revisado durante a execução do projeto, pelo menos a cada tarefa iniciada. O plano é dividido por fases, sendo cada fase dada em uma subseção própria. É importante deixar claro, porém, que tal divisão por fase não corresponde exatamente ao processo CRISP-DM, apesar da similaridade.

3.1 Fase 1 — Carregamento dos dados

Passos:

1. Carregar os dados em um "Jupyter notebook" (ferramenta a ser utilizada).

2. Verificar se pressupostos fundamentais sobre os dados são atendidos.

3. Verificar a qualidade dos dados. Se houver valores faltantes, anotar a frequência e propor solução.

4. Verificar se os dados precisam ser tratados de forma especial, por exemplo em uma estrutura de dados adequada, devido ao tamanho ou outra peculiaridade.

5. Produzir relatório de carregamento de dados, atentando especialmente para o registro de conclusões e descobertas. Se for percebido que certos métodos, como o método kernel, é impraticável, deve-se fazer registro.

Saídas:

- Código python encapsulado, para carregamento de dados.
- Relatório de carregamento dos dados.

3.2 Fase 2 — Descrição dos dados

Passos:

1. Sumarizar estatísticas que revelem aspectos importantes dos dados, como: média, variância, assimetria e curtose.

2. Exibir gráficos do banco de dados com dimensionalidade reduzida, para análise visual.

3. Explorar os dados mais a fundo. Este passo é livre em seu curso. O objetivo principal é produzir "insights" dos dados.

4. Produzir relatório de descrição dos dados. Incluir todas as métricas abordadas, gráficos mais importantes, e "insights" obtidos.

Risco: se a qualidade dos dados não for suficiente para continuar, deve-se procurar uma nova fonte de dados, e reiniciar o processo da fase 1.

Saídas:

- Código python para descrição dos dados. Deve ser separado por funções que tenham um objetivo cada, por exemplo: "plot2D", "printBasicMetrics", etc.
- Relatório de descrição dos dados.

3.3 Fase 3 — Preparação dos dados

Passos:

1. Solucionar os problemas de qualidade dos dados, como especificado pelo relatório de descrição dos dados.

2. Se houver necessidade, comprimir os dados, talvez com o mesmo método utilizado na fase dois.

3. Verificar se há ganho na criação de atributos derivados, ou de instâncias artificiais, para o problema de classificação.

4. Verificar se há necessidade de normalizar os dados. Se sim, decidir como normalizar os dados e assim proceder. Se houver diferentes opções para os diferentes modelos propostos, criar uma tabela para cada modelo.

5. Verificar se cada tabela atende plenamente as condições de seu modelo correspondente.

6. Produzir relatório de preparação dos dados.

Saídas:

- Código python para preparação dos dados. Deve conter uma função que normaliza o banco de dados original para cada modelo proposto.
- Relatório de preparação dos dados. Deve conter as principais decisões tomadas, assim como os motivos contra e a favor de cada uma delas. Acidentes de percurso, assim como novas ideias, devem ser incluídas.

3.4 Fase 4 — Modelagem

Modelos a serem considerados inicialmente:

- Rede neural para classificação (talvez não seja uma boa ideia devido à pequena quantidade de dados.)
- Árvores de decisão para classificação. Isso pode ajudar muito a interpretar os resultados.

Passos:

1. Montar esquema de testes. Isto envolve a decisão do tamanho dos conjuntos de teste e validação (se presente).

2. Selecionar uma técnica de modelagem. especificamente, se redes neurais for impraticável, a escolha pode ser árvores de decisão. Se árvores de decisão se mostrarem insuficientes, procurar outras opções, e retornar à fase 3, para preparar os dados para tal novo modelo. Se os dois modelos forem praticáveis, construir tabelas de "lift" e "gain" para auxiliar na decisão.

3. Se existirem metaparâmetros, ajustá-los da melhor maneira possível, e documentar o processo, a conclusão e as razões que embasam a escolha.

4. Descrever o modelo escolhido e os "insights" obtidos durante a utilização.

5. Verificar e argumentar a qualidade do modelo.

6. Analisar a capacidade do modelo de alcançar os objetivos de negócio estabelecidos.

7. Produzir relatório de modelagem.

Saídas:

- Relatório de modelagem. Deve conter todas as decisões tomadas, assim como os argumentos contra e a favor.
- Codigo python para construir o modelo finalmente escolhido. Se os dois modelos foram construidos, o código para cada um deles deverá ser mantido, mas separadamente.

Risco: nenhum dos dois modelos propostos tem resultados satisfatórios. Neste caso, é necessária uma nova iteração, a partir da fase 3, para garantir que os dados estão preparados para um novo modelo.

Observação: pode ser que o código para construção de mais de um modelo seja desenvolvido. Todavia, *somente um modelo* deve ser escolhido para a classificação.

3.5 Fase 5 — Avaliação

Esta fase está essencialmente relacionada com a retrospectiva do processo de mineração de dados, especialmente dos resultados obtidos até o momento.

Passos:

1. Avaliar os resultados do modelo de acordo com os critérios de sucesso para os objetivos de negócio. Analisar com um pouco mais de cuidado se algum "insight" indica que um novo caminho deve ser tomado. Se não, então pode-se prosseguir para o próximo passo, e é assumido que os resultados do modelo são satisfatórios.
2. Revisar todo o processo de mineração de dados. Este passo ajuda a evitar que algum componente essencial da mineração de dados foi desconsiderado. Deve incluir uma retrospectiva sobre falhas, caminhos mal andados, escolhas alternativas que poderiam ter sido tomadas.
3. Refletir sobre cada objetivo de negócio, e como o modelo ajudou a atingi-lo.
4. Produzir relatório de avaliação.

Risco: se houver necessidade de voltar no processo devido a um erro crasso, mas que leve muito tempo, o projeto será continuado apesar de inadequado, e o relatório final deve explicitamente incluir este fracasso.

Saída: relatório de avaliação

3.6 Fase 6 — Produção do relatório final

Aqui, o relatório final deve ser produzido. Já não há mais considerações a serem feitas. Ele deve ser algo parecido com uma integração de todos os pequenos relatórios produzidos durante o curso do projeto. Além dele, um simples "Jupyter notebook" que sumarize os passos do processo de banco de dados deve ser desenvolvido, para apresentação do passo-a-passo do processo de mineração de dados, caso se faça necessário.

Passos:

1. Escrever Jupyter notebook com os passos mais importantes, utilizando o código Python produzido durante o projeto.
2. Escrever relatório final.

Referências

[1] https://www.sbd.org.br/dermatologia/pele/doencas-e-problemas/cancer-da-pele/64/. Acessado em 7 de março de 2021.

[2] https://www.isic-archive.com/!/topWithHeader/wideContentTop/main. Acessado em 7 de março de 2021.

[3] https://www.nature.com/articles/nature21056. Acessado em 7 de março de 2021.

[4] https://www.kaggle.com/fanconic/skin-cancer-malignant-vs-benign. Acessado em 6 de março de 2021.

[5] https://www.isic-archive.com/!/topWithHeader/tightContentTop/about/isicArchive. Acessado em 8 de março de 2021.