

Proposta para Trabalho de Conclusão do Nanodegree Engenheiro de Machine Learning

Marcelo Hanones ago/2018

Histórico do assunto

A minha motivação pessoal em machine learning é a investigação do comportamento humano em grandes grupos. Eu vejo o machine Learning como um novo “microscópio” que nos permite investigar eventos enquanto grupo interconectado. Essa capacidade de enxergar o coletivo traz uma luz científica de forma a podermos investigar o “efeito borboleta” das coisas. O próprio machine learning em si é um agrupamento, disciplinas diferentes como matemática e estatística que se complementam e ganham poder através da computação.

Acredito que a linguagem escrita seja o meio principal para interação no meio digital e por isso existe hoje em um volume tão grande e crescente. O estudo desse volume pode trazer novos conhecimentos que agreguem na capacidade que uma maquina tem de entender significados e contextos mais profundos. Meu objeto de estudo é o processamento de linguagem natural, ou NLP Natural Language Processing.

Descrição do problema

O problema consiste em fazer o computador captar o sentido de uma string para que esta seja classificada como positiva ou negativa.

A string em questão são os reviews do site [imdb](#). Um review consiste em um texto livre onde um usuário emite a sua opinião sobre determinado filme. Além do texto, o review também acompanha uma nota de 1 a 10. Um filme é classificado como positivo se tiver mais de 6 estrelas, e como negativo se tiver menos de 5 estrelas.

O desafio é fazer o computador classificar um review sem o rating, usando somente o texto. Esse problema é popularmente conhecido como análise de sentimento.

Conjuntos de dados e entradas

A base de dados usada é a da competição do Kaggle [Bag of Words Meets Bags of Popcorn](#). Este dataset contem três arquivos:

- labeledTrainData.tsv com 25.000 entradas classificadas como positiva ou negativa. Estes dados serão divididos em três seções: treino, validação e teste.
- testData.tsv com 25.000 entradas sem classificação.
- unlabeledTrainData.tsv com 50.000 entradas sem nenhuma classificação para aprendizado não supervisionado.

O dataset é perfeitamente balanceado em 50/50 entre as duas classes pos/neg.

Descrição da solução

Trata-se de um problema de aprendizado supervisionado onde o dataset de treinamento contém os reviews já classificados. O trabalho é fazer uso da capacidade de aprendizado dos algoritmos para captar a relação que existe nesse data Set de treinamento para posteriormente aplicar esse modelo aprendido aos dados nunca vistos.

O processo que transforma o texto em números é chamado de vetorização. Esse processo funciona como uma ponte, transformando um texto cru em algo que possa ser processado por um algoritmo. Esse processo é de suma importância pois diversas nuances do texto precisam ser mantidas ao mesmo tempo que traduzidas em relações numéricas que possam ser capturadas por um algoritmo.

O modelo usado é o Bag of Words (BOW). Este modelo leva em consideração a frequência das palavras, resultando numa matriz extensa com todas os termos únicos usados em todo o data Set. Além de computacionalmente custoso, este modelo não leva em consideração a ordem das palavras ou qualquer outro contexto. Para meu próximo passo no aprendizado de NLP pretendo estudar modelos de entendimento de texto mais complexos que levem em consideração elementos mais sutis presentes num texto.



Técnicas de redução da dimensionalidade serão usadas afim de tornar o processo mais factível para tempo e capacidade aceitáveis. Essas técnicas conseguem reduzir o numero de dimensões com perda mínima ou aceitável de informação.

Técnicas de Ensemble também serão usadas. Essa técnica consiste em aplicar um algoritmo para fazer escolhas inteligentes em cima de um grupo de outros algoritmos. O Ensemble opera num nível de abstração maior como se fosse o algoritmo do algoritmo. Da mesma forma que uma string é uma reunião de caracteres numa ordem especifica, um Ensemble é uma reunião de algoritmos numa ordem especifica orientados a atacar determinado problema como overfitting ou underfitting por exemplo.

Os algoritmos escolhidos para a tarefa possuem diferentes características, lineares, não-lineares, paramétricos, não-paramétricos.

Modelo de referência

O modelo de referência para este projeto é o patamar de 95% de auc_score usando *Logistic Regression* como classificador. Este score representa aproximadamente os 170 melhores resultados da competição.

174	↗ 34	Caesar11		0.95049	2	3y
175	new	newprolab.com.vladimir.litvin...		0.94949	11	3y

Este score deve vir acompanhado de uma implementação com tempo de processamento razoável e uma learning_curve sem overfitting.

Métricas de avaliação

A métrica usada é a `roc_auc`. A métrica `accuracy` poderia ser usada uma vez que o data é balanceado. Optei por usar `roc_auc` já simulando os casos mais comuns onde o datasets são desbalanceados. O uso do `roc_auc` traz também requisitos de arquitetura uma vez que o método `predict_proba` é necessário e nem todos algoritmos são aderentes.

Design do projeto

Um dos grandes desafios que encontrei em machine learning foi a aplicação dos conceitos em conjunto de forma a se criar uma linha de raciocínio. A partir do momento que a intuição básica dos conceitos começou a fazer sentido teórico, foi necessário criar uma ferramenta que me permitisse a organização desse fluxo de ideias em código. Uma vez que a quantidade de ferramentas e técnicas são amplas, a experimentação se tornou essencial para descobrir o que pode ou não funcionar para um dataset. Este processo de tentativa e erro demanda documentação clara para que por exemplo uma informação possa ser resgatada num momento posterior onde seja necessário fazer comparações com passos anteriores. A tarefa de machine é bastante “artesanal” e exige doses altas de organização dos passos dados. Portanto esse projeto tem bastante foco na engenharia da ferramenta.

Foi criada uma classe para organizar funções para reduzir a re-digitação de código, reduzindo tempo debugando erros e garantindo assim consistência de que todos resultados vieram de um mesmo código. A medida que as análises foram se estendendo por vários Jupyter notebooks, a unicidade do código passou a ser primordial.

De forma bem geral, essa classe opera em dois níveis distintos. Uma processa elementos individuais e a outra processa grupos de elementos. Por exemplo a função que faz parameter-tuning dos algoritmos opera em âmbito individual, já a função que plota o ranking dos scores opera em âmbito coletivo.

Algumas decisões importantes de arquitetura devem ser consideradas:

- O dataset foi mantido no formato `DataFrame` e não em matriz. Essa decisão foi tomada para facilitar a limpeza dos dados na parte de pre-processamento. Embora este projeto não tenha tido grandes demandas na limpeza do texto, optei por uma arquitetura que simulasse essa necessidade. Essa decisão teve alguns efeitos colaterais no Pipeline, forçando o uso de `CustomTransformers` tanto para selecionar a(s) coluna(s) correta(s) e converter em matriz de acordo com a necessidade.
- Houve a intenção de manter maior parte do pre-processamento sendo realizado “on-the-fly” via Pipeline ao invés de transformar todo o dataset de antemão. Dessa forma evitaria data-leakage uma vez que as transformações seriam aplicadas fold a fold. Essa decisão aumentou a carga computacional tornando o processo mais lento.
- A biblioteca usada para parameter tuning foi o `HyperOpt`. Essa biblioteca implementa métodos Bayesianos que usam escolhas passadas para escolher os próximos parâmetros a serem avaliados. Um segundo algoritmo realiza essas escolhas tomando como base uma distribuição probabilísticas dos possíveis eventos. Ao invés de processar todos os dados iteração a iteração, essa biblioteca faz as escolhas baseadas nessa distribuição otimizada para minimizar a perda.
- Parte do meu objetivo nesse projeto inclui a experimentação de novas bibliotecas como `MLXtend` e `YellowBrick`.

Referências:

<https://www.kaggle.com/c/word2vec-nlp-tutorial#part-3-more-fun-with-word-vectors>

<http://steventhornton.ca/hyperparameter-tuning-with-hyperopt-in-python/>

<https://towardsdatascience.com/automated-machine-learning-hyperparameter-tuning-in-python-dfda59b72f8a>

<https://districtdatalabs.silvrback.com/parameter-tuning-with-hyperopt>

<https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc>

<https://towardsdatascience.com/automated-machine-learning-hyperparameter-tuning-in-python-dfda59b72f8a>