Nanodegree Engenheiro de Machine Learning

Proposta de projeto final: Prevendo salários de vagas de emprego

Leonardo Mizoguti
03 de Fevereiro de 2018

Histórico do assunto

O recrutamento de funcionários é algo tão antigo quanto as organizações em si, e tem evoluído com o passar do tempo. Na era pré-industrial, não se tinha a preocupação em **avaliar candidatos**, pois em geral **não era necessário**: recrutavam-se operários localmente e cargos mais altos eram destinados às pessoas de classes mais altas. Entretanto, com a revolução industrial, e mais atualmente com a revolução da informação, a **estrutura interna** das organizações vieram se tornando cada vez **mais complexas**, exigindo cada vez mais pessoas com **habilidades específicas** para realizar **tarefas específicas** [1]. Dificilmente conseguem-se encontrar candidatos de diversas qualificações em um mesmo local.

Dessa forma, surgiu o **recrutamento moderno**. Antes da internet, a melhor forma de encontrar pessoas era publicar vagas de emprego em jornais. Com a internet surgiram diferentes **serviços de classificados online**, como *Monster* e *Indeed*. Esses serviços certamente revolucionaram o setor ao fornecer o alcance a uma **audiência muito maior** do que os classificados dos jornais jamais puderam oferecer. Entretanto, na essência eles **não funcionam** muito diferentemente dos classificados, pois tanto candidatos como empresas ainda têm o **trabalho de buscar e encontrar** a oportunidade que melhor lhes convém.

Nos últimos anos surgiram novos serviços que se propõem a adicionar inteligência a esse processo. Plataformas como a *Hired* [2] nos Estados Unidos e a *Revelo* [3] no Brasil se dedicam a fazer de forma **automatizada** o *matching* entre oportunidades de emprego e candidatos, tornando o processo muito mais **simplificado** para ambos os lados.

Descrição do problema

Existem muitos fatores a serem considerados para se fazer o *matching* entre candidatos e oportunidades de emprego. É preciso entender quais são as qualificações necessárias, levar em conta a distância física entre o candidato e a empresa, entre outras coisas. Um aspecto também importante é a **conformidade do salário** proposto com as **expectativas salariais dos candidatos**. Muitas vezes empresas preferem **não divulgar o salário** para uma determinada vaga, portanto um sistema de *matching* deveria ser capaz de **estimar** quanto seria o **salário**

proposto para poder atrair e recomendar os candidatos com uma **pretensão salarial correspondente**.

Portanto, o problema a ser resolvido consistem em desenvolver um **modelo de regressão** que recebe como entrada um conjunto de **dados categóricos** sobre determinada vaga (como localização, tipo, setor…) além do **título da vaga e da descrição da vaga** (ambos campos de texto livre, não estruturados), e retorna **um valor numérico** correspondendo a uma estimativa do **salário proposto** para a vaga. O modelo deverá ser treinado utilizando-se dados de vagas com o real salário proposto como variável alvo.

Conjunto de dados e entradas

Para este projeto, gostaria de analisar os dados de vagas de emprego da plataforma *Monster* [4] disponíveis na plataforma *Kaggle* [5].

Este dataset é formado por 22,000 **vagas de emprego** baseadas nos Estados Unidos em diferentes setores da economia e com diferentes níveis de senioridade. O dataset apresenta alguns campos, dentre os quais os mais relevantes para este projeto são:

- *job_description*: descrição da vaga (campo de texto livre, não estruturado);
- *job_title*: título da vaga (campo de texto livre, não estruturado);
- *job_type*: tipo da vaga, i.e. tempo integral ou meio-período (campo de texto livre, padronizado na maioria das vagas);
- *location*: localização da vaga (campo de texto livre, com indicação da cidade e do estado na maioria das vagas);
- *organization*: segmento da empresa (e.g. engenharia, educação, saúde, etc – campo de texto padronizado);
- *sector*: segmento da vaga (e.g. gerência, atendimento ao cliente, logística, etc – campo de texto padronizado);
- *salary*: salário proposto (campo de texto não padronizado).

Descrição da solução

A ideia deste projeto é treinar com parte dos dados um **modelo de regressão** capaz de **prever o salário** (variável alvo) de uma vaga **com base nos demais atributos**, como a descrição, a localização, o segmento da vaga, etc. Uma vez que se tenha o modelo treinado, pode-se verificar sua acurácia com um conjunto de teste.

Alguns dados categóricos deverão ser padronizados (como localização, tipo de vaga), e os dados de texto livre (título e descrição) deverão ser convertidos em vetores numéricos. Neste projeto, serão utilizados dois métodos: *word2vec* [6] e *n-gram + TF-IDF* [7].

Modelo de referência

Há aproximadamente 5 anos, a plataforma *Adzuna* promoveu na plataforma *Kaggle* uma competição [8] cujo objetivo era também **prever salários** de vagas de emprego baseadas no Reino Unido com base em um conjunto de atributos.

Jackman, S. e Reid, G., da University of British Columbia, publicaram no mesmo ano um artigo [9] no qual eles compararam diferentes modelos na resolução do problema proposto pela *Adzuna*, como **regressão LASSO**, **regressão por máxima verossimilhança**, **regressão por rede neural** e **regressão por *random forest***.

A conclusão da dupla foi de que o modelo de **random forest** foi o que melhor performou, com um desvio absoluto médio de **£5,000** no conjunto de teste.

Métrica de avaliação

Para este projeto, a proposta é utilizar como métrica de avaliação a mesma utilizada por Jackman e Reid, i.e. o **desvio absoluto médio (DAM)**. Basicamente essa métrica mede, na média, o quanto o **valor previsto** pelo modelo **diverge** (em termos absolutos) do **valor proveniente dos dados**.

$$DAM = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

Onde n é o número de pontos, y_i corresponde a um valor previsto pelo modelo e x_i ao valor proveniente dos dados correspondente.

Resumindo, o modelo treinado deverá **prever os salários** dos dados de treinamento, e será medido o **desvio médio absoluto** entre os salários previstos e os salários proveniente dos dados de teste.

Design do projeto

1ª etapa: limpeza e normalização dos dados

Antes de tudo, será necessário fazer uma limpeza e normalizar os dados. A grande maioria das vagas presentes não possuem informação de salário, outras não a possuem explicitamente, e as que possuem em geral está em um formato não padronizado. Existem indicações de salário anual, salário por hora, indicações de benefícios não relevantes para a análise, dentre outros problemas.

Será necessário normalizar e limpar outras colunas, como localidade, tipo de vaga… E também remover eventuais entradas duplicadas.

2ª etapa: transformação das descrições em vetores

Um dos problemas com o uso de texto não estruturado para modelos de machine learning é a sua representação. É necessário transformar o texto em um vetor de números, assim como é feito por exemplo com imagens, tratadas como um vetor de pixels.

Será necessário tokenizar essas descrições (remover pontuações e caracteres especiais, transformar letras maiúsculas em minúsculas, extrair as palavras), e aplicar algum modelo que as converta em um formato utilizável.

A abordagem idealizada é utilizar 2 modelos, podendo-se comparar ao final os resultados de cada um: *n-gram* com uma transformação *TF-IDF* e *word2vec* baseado em um modelo pré-treinado pelo Google [10].

As próximas etapas serão realizadas **para ambos os modelos** de tratamento de texto.

3ª etapa: validação dos modelos

Usando um conjunto de teste e um conjunto de validação, a ideia será implementar três modelos de regressão diferentes. A ideia inicial é utilizar os modelos *K-nearest neighbors*, *Máquina de vetores de suporte* e *random forest* para fazer a regressão. Uma vez com os três modelos implementados, será possível verificar qual dos três possui uma performance melhor no conjunto de validação, sendo então escolhido para a próxima etapa.

4ª etapa: ajuste do modelo

Uma vez o modelo final escolhido, pode-se fazer o ajuste dos parâmetros pertinentes. Usando a técnica do *Grid Search*, será possível definir o melhor conjunto de parâmetros para o modelo.

5ª etapa: análise dos resultados

Assim que o modelo final ajustado for obtido, a métrica de avaliação poderá ser medida com um conjunto de dados de teste e comparada com o benchmark indicado. Será feita uma análise das possibilidades que existem para melhorar a acurácia do modelo.

Além disso, serão analisadas as diferenças dos resultados do modelo que usará o *word2vec* para o tratamento dos campos de texto livre e do modelo que usará o *n-gram + TF-IDF*.

Referências

[1] Lamri, J. (2013). A Brief History of Recruitment. The Huffington Post.
http://www.huffingtonpost.co.uk/jeremy-lamri/recruitment_b_4485993.html

[2] Hired – Job Search Marketplace.
https://hired.com

[3] Revelo – Talentos de alto potencial.
https://www.revelo.com.br

[4] Monster Jobs – Job Search, Career Advice & Hiring Resources.
https://www.monster.com

[5] Kaggle. US jobs on Monster.com.
https://www.kaggle.com/PromptCloudHQ/us-jobs-on-monstercom

[6] Word2Vec. Wikimedia Foundation.
https://en.wikipedia.org/wiki/Word2vec

[7] Sci-kit Learn documentation. Feature extraction – Common vectorizer usage
http://scikit-learn.org/stable/modules/feature_extraction.html#common-vectorizer-usage

[8] Kaggle. Job Salary Prediction.
https://www.kaggle.com/c/job-salary-prediction

[9] Jackman, S. & Reid, G. (2013). Predicting Job Salaries from Text Descriptions.
https://open.library.ubc.ca/clRcle/collections/graduateresearch/42591/items/1.0075767

[10] word2Vec. Google Code Archive.
https://code.google.com/archive/p/word2vec/