Udacity

Nanodegree Engenheiro de Machine Learning

Projeto Final

Predição de Aprovação em Visto Permanente Americano

Autor:
Lucas Baião

9 de agosto de 2018

São Paulo

Sumário

1 Definição **2**
1.1 Visão Geral de Projeto . . . 2
1.1.1 Origem do Problema . . . 2
1.2 Descrição do Problema . . . 3
1.2.1 Solução Atual e Conjuntura . . . 3
1.2.2 Conjunto de Dados . . . 3
1.2.3 Aplicação e Algoritmos . . . 4
1.3 Métricas e Avaliação . . . 4

2 Análise **6**
2.1 Exploração dos dados . . . 6
2.1.1 Limpeza Inicial dos Dados . . . 7
2.2 Visualização dos dados . . . 7
2.3 Algoritmos e Técnicas . . . 9
2.4 Benchmark . . . 10

3 Metodologia **11**
3.1 Pre-processamento dos Dados . . . 11
3.2 Implementação da solução . . . 11
3.2.1 Regressão Logística . . . 12
3.2.2 Random Forest . . . 12
3.2.3 Gradient Boosting . . . 13
3.2.4 Naive Bayes . . . 13
3.2.5 Escolha do Algoritmo Final . . . 14
3.3 Refinamento . . . 14

4 Resultados **15**
4.1 Avaliação do Modelo e Validação . . . 16
4.2 Justificativa . . . 16

5 Conclusão **16**
5.1 Avaliação Geral do Modelo Escolhido . . . 16
5.2 Avaliação da Solução . . . 18
5.3 Melhorias . . . 18

1 Definição

1.1 Visão Geral de Projeto

Desde a metade do século XX a movimentação humana e os processos migratórios têm se intensificado sem precedentes, levando a uma grande quantidade de processos de expatriação e pedidos de vistos permanentes em diversos países. Notadamente, países com maior grau de desenvolvimento humano costumam ter uma alta taxa destas solicitações, e é cada vez mais comum a ocorrência deste tipo de situação a partir de propostas de trabalho em diferentes países. Em países com alta taxa de migração existe a tendência ao controle de entrada de estrangeiros com processos de pedido de visto.

1.1.1 Origem do Problema

Todo tipo de movimentação humana gera grande quantidade de informações, tanto para a população que migra, quanto para as localidades de origem e destino. A partir destes dados estatísticos é possível prever o sucesso, ou não, de uma solicitação de visto permanente. Para esta análise o país alvo será os Estados Unidos da América, devido a sua grande taxa de crescimento populacional desde sua fundação (de aproximadamente 200 milhões para mais de 325 milhões de pessoas apenas no período entre 1975 e 2018, segundo [1]), sua grande taxa de imigração (mais de 96 milhões desde 1610, também segundo[1]) e o controle de fronteiras estabelecido de forma contínua e documentada.

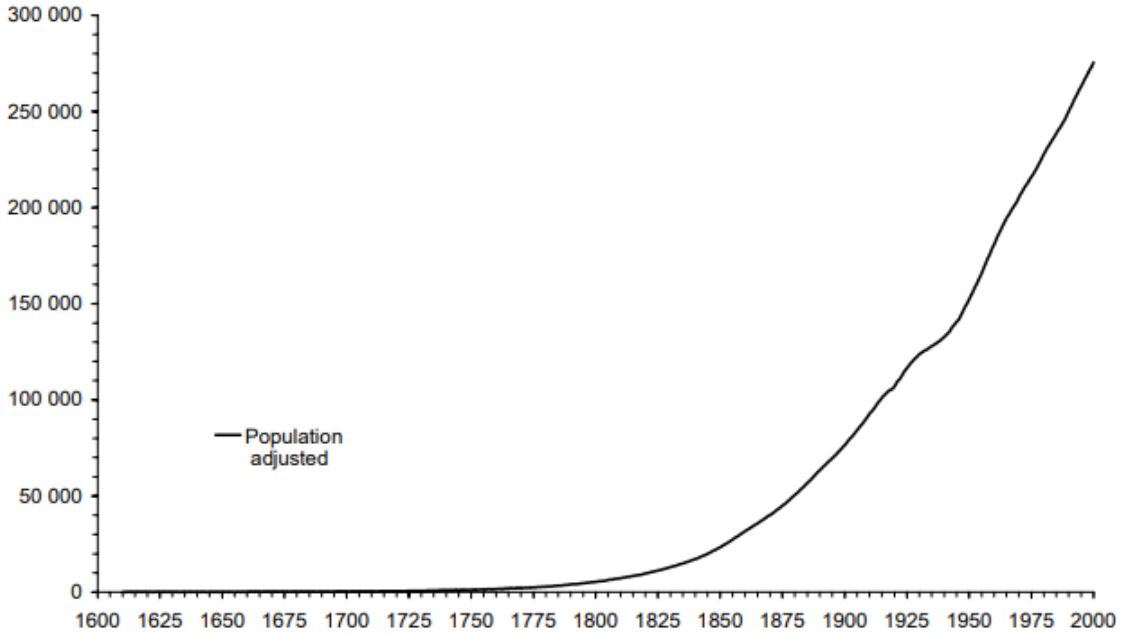

Figura 1: População dos EUA entre 1600 e 2000 [1]

Period	Immigrants settled
1610-1699	67 000
1700-1799	1 145 000
1800-1899	27 093 000
1900-1999	68 491 000
Total	96 795 000

Figura 2: Imigração para os EUA por período[1]

1.2 Descrição do Problema

Pensando em auxiliar o processo de solicitação de vistos permanentes para os EUA e fazendo uso da grande quantidade de dados existentes sobre este tipo de processo, o intuito deste projeto é criar um sistema de classificação prevendo o sucesso ou não de uma solicitação para imigração para os EUA.

1.2.1 Solução Atual e Conjuntura

Hoje a avaliação das solicitações é feita somente pelos órgãos competentes junto ao governo americano, portanto, ao solicitar o visto permanente não é possível ter qualquer certeza quanto a autorização ou não para migração.

Este tipo de situação se torna ainda mais complicada ao se observar que a solicitação do visto vem carregada de outras decisões, expectativas e ações tomadas pelo solicitante, e que em caso de um negativo podem trazer diversos problemas. Com um algoritmo preditivo é possível, portanto, encontrar os pontos com necessidade de algum tipo de intervenção para aumentar a chance de aprovação junto ao governo americano, além de evitar ações antes de uma aprovação.

1.2.2 Conjunto de Dados

O conjunto de dados utilizado foi obtido a partir da plataforma de datasets Kaggle [2]. Os dados estão em formato .csv, contendo 154 colunas (uma para cada parâmetro de solicitação) por 374.362 linhas (uma para cada processo de solicitação de visto). Entre estas colunas temos dados como: País de origem do solicitante, salário, cidade e estado de destino do solicitante, data, tipo de visto, além da informação de aceite ou não da solicitação.

As informações do dataset estão abertas a consulta na internet,tendo sido disponibilizadas pelo ”Departament of Labour”(DOL) a partir de dados coletados pelo ”Department of Homeland Security’s U.S. Citizenship and Immigration Services”(USCIS). Este Dataset conta com todos os registros de solicitação de visto permanente para os EUA no período compreendido entre 2011 e 2016, sendo aproximadamente 5 anos e meio de dados.

A maioria das colunas é de grande importância para a criação de um ”perfil”do solicitante ao visto, sendo este primordial para a decisão das autoridades quanto a autorização ou não de imigração a um país, sendo este perfil uma forma de traçar a ”probabilidade”de autorização ou não de entrada.

1.2.3 Aplicação e Algoritmos

A partir do problema definido, o projeto a seguir compreenderá o desenvolvimento de uma solução baseada em machine learning, capaz de prever a autorização, ou não, de imigração para os EUA. O algoritmo será desenvolvido sob as bases da biblioteca Sci-kit learn, utilizando diversos tipos de algoritmo (regressões, SVM, redes neurais e etc...). A definição e aplicação destas ferramentas sera fundamentada a partir da correlação entre os dados e os resultados de análises estatísticas sobre os mesmos.

A partir dos resultados dessa análise será possível identificar quais algoritmos se adequam melhor a solução esperada, de forma a gerar a classificação (visto concedido ou não) de forma assertiva.

1.3 Métricas e Avaliação

A principal métrica de avaliação utilizada é acurácia do sistema (porcentagem de acertos das predições do algoritmo criado, utilizando como input uma parcela de dados de teste).

$$A_c = \frac{C}{p}$$

C = Acertos nas predições
P = quantidade de predições
Ac = Acurácia

Figura 3: Equação da Acurácia de um modelo classificador

A fim de refinar a comparação e entender melhor o comportamento do algoritmo, foram exibidas as matrizes de confusão para cada algoritmo testado exibindo erros e acertos reais ou falsos. Conforme a matriz exemplo abaixo.

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Figura 4: Exemplo de Matriz de Confusão

A partir da visualização da matriz de confusão, também é possível avaliar outros indicadores como precisão e recall, de modo a compreender como os falsos acertos e falsos erros influenciam no resultado do algoritmo. Sendo assim serão utilizados os três seguintes indicadores.

- Precisão: Índice de assertividade geral do algoritmo, definido pela equação abaixo

$$P = \frac{Tp}{Tp + Fp}$$

Tp = Verdadeiros Positivos
Fp = Falsos Positivos
P = Precisão

Figura 5: Equação de Precisão

- Recall: Índice de assertividade algoritmo em relação aos seus falsos negativos, definido pela equação abaixo

$$R = \frac{Tp}{Tp + Fn}$$

Tp = Verdadeiros Positivos
Fn = Falsos Negativos
R = Recall

Figura 6: Equação de Recall

- F1: Relação entre Precisão e Recall, dado pela média harmônica dos dois valores e definido pela equação abaixo

$$F_1 = 2\frac{P \times R}{P + R}$$

P = Precisão
R = Recall

Figura 7: Equação de F1

Levando em consideração a finalidade da classificação a ser realizada, é bastante importante que tenhamos um recall alto, visto que falsos negativos podem gerar uma certa 'desistência' quanto a continuidade no processo de obtenção de visto. Sendo assim o foco será em um sistema de alto recall e alta acurácia.

2 Análise

2.1 Exploração dos dados

Como consequência direta da extração dos dados de um site como o Kaggle, para este projeto possuímos um dataset bem definido, organizado e em formato de fácil leitura (.csv). Sendo assim é possível importar estes dados para o notebook python diretamente por meio do pandas. É possível então visualizar as primeiras colunas do dataset, conforme imagem á seguir:

schd_a_sheepherder	us_economic_sector	wage_offer_from_9089	wage_offer_to_9089	wage_offer_unit_of_pay_9089	wage_offered_from_9089
NaN	IT	75629.0	NaN	yr	NaN
NaN	Other Economic Sector	37024.0	NaN	yr	NaN
NaN	Aerospace	47923.0	NaN	yr	NaN
NaN	Other Economic Sector	10.97	NaN	hr	NaN
NaN	Advanced Mfg	100000.0	NaN	yr	NaN
NaN	Other Economic Sector	37024.0	NaN	yr	NaN
NaN	Educational Services	47084.0	52000.0	yr	NaN

Figura 8: Trecho do Dataset

As colunas do dataset compreendem diversos tipos de informação sobre o perfil do solicitante, tendo dados principais como:

- País de origem;
- Salário Oferecido;
- Empresa de Destino;
- Cidade de Destino;
- etc;

Dentre as 154 colunas totais, a maioria se encontra com grande quantidade de valores nulos, conforme observável a partir de uma contagem visual simples no python:

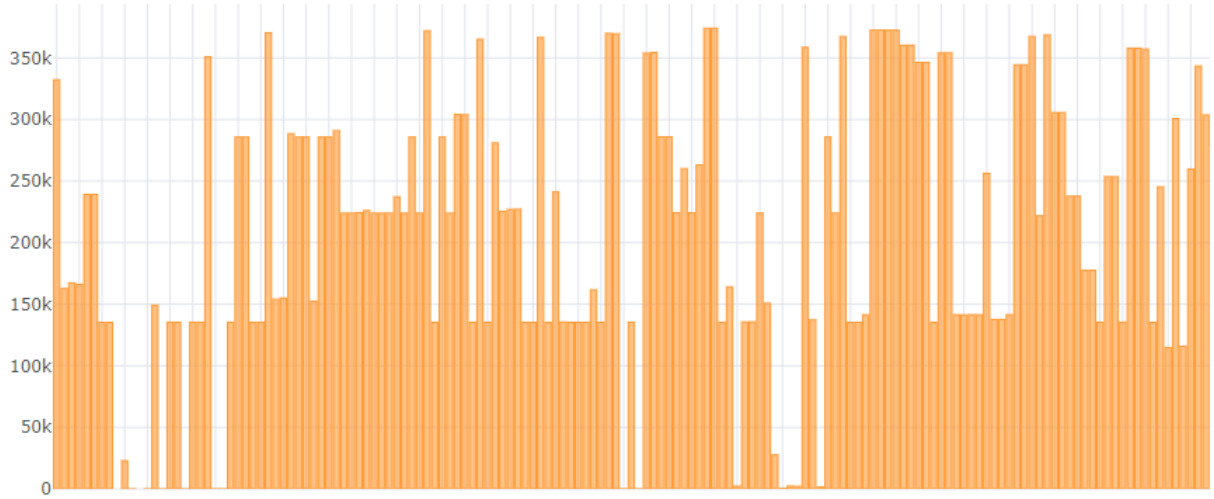

Figura 9: Contagem de Valores Nulos em Cada Coluna

Conforme é possível observar no gráfico, existem diversas colunas com uma grande quantidade de valores nulos, sendo assim estes valores não poderão ser utilizados e serão descartados ou reaproveitados de outra forma.

2.1.1 Limpeza Inicial dos Dados

De forma a minimizar os erros e a necessidade de poder computacional para realizar as análises estatísticas sobre os dados, o primeiro passo escolhido foi eliminar colunas com quantidade muito grande de valores nulos. A limpeza destas colunas se deu a partir de uma escolha estatística de forma a não comprometer os dados ou causar qualquer desvio de tendência dentro do conjunto.

Utilizando a contagem de nulos, foi verificada a quantidade total de dados existentes e considerada a quantidade de dados restantes após uma limpeza definida por um limiar. Em busca de manter o dataset com a maior quantidade de indivíduos possíveis, o valor definido como limiar para eliminação de cada coluna foi de aproximadamente 1% do dataset, portanto o limiar ficou em torno de 3600 valores nulos, qualquer coluna com valor maior que esse foi eliminada do dataframe.

Em busca da uniformização destes dados foram realizados diversos tipos de limpeza e filtragem em busca de eliminar os valores nulos e inconsistentes, entre eles, remoção de linhas com grande quantidade de colunas nulas, aplicação de médias ou módulos em valores inexistentes e elaboração de filtros para remoção de caracteres especiais. Além disso, nos campos de texto foi realizada a uniformização de nomes de forma a deixar todos minúsculos, em siglas foi aplicado filtro para maiúsculas.

2.2 Visualização dos dados

Diversas informações podem ser extraídas dos dados a partir de diferentes visualizações, para aumentar a compreensão quanto aos dados e sua complexidade foram plotados di-

versos gráficos com indicadores chave para compreender como identificar os padrões nos dados e sendo assim, qual algoritmo aplicar para esta solução.

O primeiro atributo escolhido para análise foi a visualização de contagem de pedidos de visto por país. É interessante verificar, pela plotagem, que existem países com grande quantidade de solicitações em comparação a outros, portanto, essa é uma ”feature” bastante importante para a nossa análise.

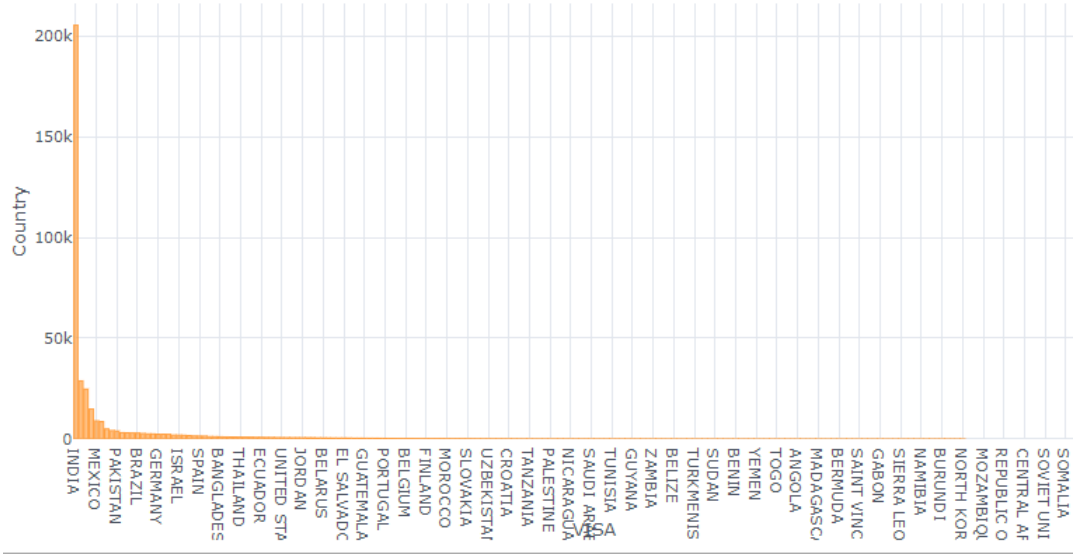

Figura 10: Pedidos de Visto por País

Outro aspecto bastante importante para este tipo de classificação é a sazonalidade. Em busca de manter neutra a sazonalidade dos dados, o campo data foi transformado em uma contagem de dias por ano, portanto a data inicial será considerada como um valor numérico a cada dia do ano, assim foi possível verificar a cada período do ano o numero de requisições e verificar suas frequências.

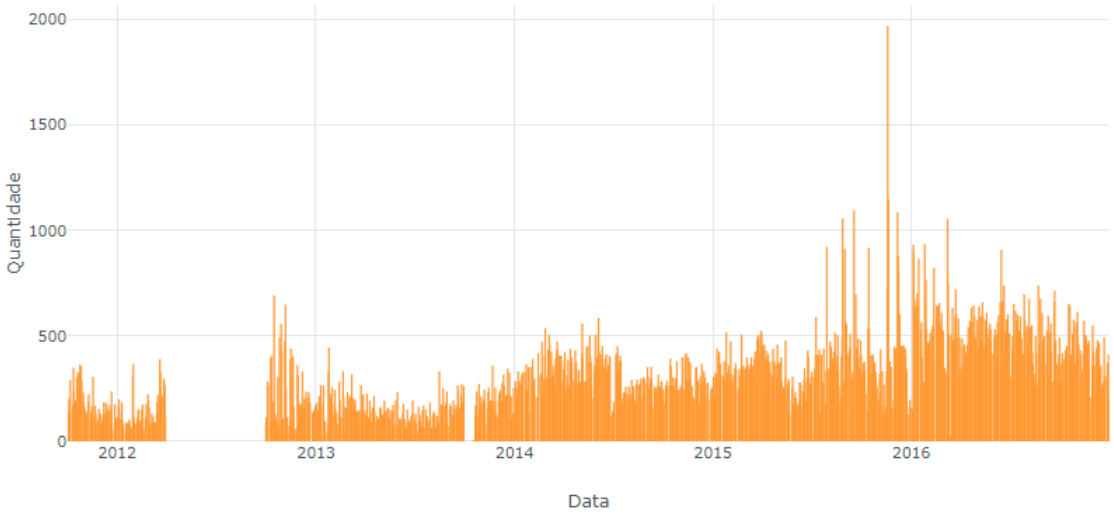

Figura 11: Sazonalidade de Pedidos

Partindo dos dados obtidos, para confirmar as tendências de alta correlação das principais colunas de dados do dataset, foi feita a análise dos salários anuais dos requisitantes. A ideia principal foi verificar alguma tendência de aprovação para alguma fração específica de salário. Para isso foram plotados contagens de aprovação e total por faixas de salário.

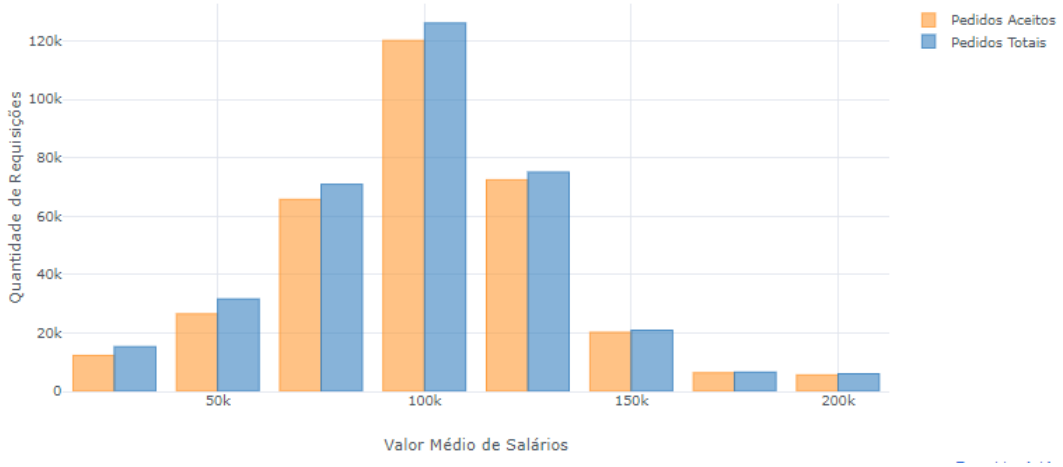

Figura 12: Requisições por Faixa de Salário

Como é possível observar pelo gráfico, a quantidade de requisições varia em uma distribuição próxima a normal, entretanto, existe uma tendencia de maior aprovação para as pessoas com maiores salários, sendo este um fator crucial para a realização da classificação necessária.

2.3 Algoritmos e Técnicas

A solução inicial se utilizou de métodos de Grid Search para identificar o melhor algoritmo destinado a esta aplicação. Conforme Haykin [4] a técnica de Grid Search é extremamente indicada para aplicações baseadas em dados numéricos e datasets para classificação, encontrando o melhor resultado para o tipo de dado escolhido. A grid search é uma tabela de atributos que serão testados a partir de um método de ”força bruta” de forma a identificar os melhores valores de refinamento.

Esta aplicação de Grid Search foi utilizada com os seguintes algoritmos:

- Regressão Logística;
- Random Forest;
- Naive Bayes;
- Gradient Boost;

Foram feitos diversos treinamentos com cada um dos algoritmos aplicados, utilizando diversos valores de hiper parâmetros, os quais o Grid Search percorreu e otimizou. Por

se tratar de um processo iterativo e de grande consumo de poder computacional, todos os classificadores utilizados tiveram seus hiper parâmetros modificados diversas vezes em busca de maximar todos os atributos. Foram testadas diferentes combinações passadas ao Grid Search, de forma a encontrar o ponto médio entre os parâmetros passados, sem a existência de nenhuma tendencia de melhoria para valores maiores ou menores que os encontrados.

Da mesma forma, cada classificador foi treinado com os mesmos dados em diferentes épocas, de forma a maximizar as amostras de dados utilizados, com o dataset dividido em 80% dos registros para treinamento e os 20% restantes para validação.

O resultado final foi comparado por meio das métricas anteriormente apresentadas, tendo obtido resultados diversos, sendo estes apresentados na sessão de implementação da solução.

2.4 Benchmark

O benchmark definido para o projeto será o exemplo postado na plataforma Kaggle por Ambarish Ganguly [5]. A escolha de um modelo deste tipo para benchmark parte do vasto compartilhamento de conhecimentos contidos na plataforma kaggle. A aplicação de Ganguly é altamente divulgada e possui grande numero de avaliações no site.

Os resultados obtidos pelo autor se baseiam inicialmente em diversos métodos de data visualization e posteriormente a aplicação de um algoritmo de boosting (XG Boosting) para realizar a mesma classificação obtida neste trabalho. Entretanto, o autor não utiliza nenhum tipo de otimização para o algoritmo de boosting, não testa qualquer outro algoritmo e utiliza como indicadores de performance do sistema apenas a sua acurácia. Sendo assim, o presente trabalho procurou aplicar ferramentas mais avançadas, de forma a melhorar os resultados obtidos, independente da utilização de um algoritmo semelhante, ou não, ao benchmark apresentado.

As principais melhorias aplicadas foram a utilização de Grid Search, a aplicação de diversos tipos de algoritmo, maior tratamento e limpeza dos dados, além de treinamentos mais complexos com maior quantidade de poder computacional dispensado. A titulo de registro, a imagem abaixo caracteriza o modelo aplicado por Ganguly.

Modelo Aplicado	XGBoost
Number of Rounds	100
Max Depth	3
ETA	0,05
Gamma	0
Colsample by Tree	0,8
Minimum Child Weight	1
Sub Sample	1
Acurácia Final	51,40%

Figura 13: Modelo Aplicado por Ganguly [5]

3 Metodologia

3.1 Pre-processamento dos Dados

Após a limpeza prévia, os dados foram analisados quanto a seu formato, o campo de ”zip code” foi o primeiro alvo de pesquisas mais extensas, conforme a tabela a seguir, o código postal americano é um sistema linear, em que cada área tem seu código determinado e ordenado [3], sendo assim áreas geográficas adjacentes tem códigos sequenciais, sendo possível converter este zip code em um numero inteiro e utilizá-lo como designador de localização para cada indivíduo, tornando-o um valor altamente correlacionado com o sucesso ou não do processo de imigração.

Figura 14: Decodificação do Zip Code Americano

Outro dado de aspecto bastante importante para este tipo de análise é o salário que o imigrante receberá já nos EUA. Este valor foi apresentado originalmente em duas colunas, uma com a frequência de pagamento e outro com o valor para esta frequência, sendo em alguns casos anual, em outros mensal e etc. Antes mesmo da análise estatística dos dados foi feita a conversão para que todos os salários fossem apresentados em caráter anual, de forma uniformizar a classificação do sistema de machine learning.

Todas as colunas do dataset sofreram limpeza de dados, correção dos valores não existentes a partir da média em valores numéricos, ou a moda em não numéricos, por fim, caracteres especiais foram removidos, todos os campos textuais tiveram suas letras definidas para minúsculas se palavras e maiúsculas se siglas, para uniformizar os componentes.

3.2 Implementação da solução

Durante o período de testes com aplicação do Grid Search ocorreram diversos resultados não consistes, com um processo de iteração contínua com os hiper parâmetros. Inicialmente foi utilizada uma grande gama de parâmetros com batches de treinamento curtos, ou seja, em busca de reduzir o tempo inicial de treinamento, os testes foram feitos com pouco tempo de processamento, porém alterando diversas combinações de ”Performance” do algoritmo.

3.2.1 Regressão Logística

O primeiro algoritmo testado foi o de Regressão Logística, um algoritmo básico e bastante generalista, embora o dataset fosse grande suas dimensões ainda poderiam ser utilizadas neste algoritmo. O único hiper parâmetro modificado foi "C" (Inverso da força de regularização), visto que é o único parâmetro puramente numérico que pode alterar os resultados diretamente, a ideia foi modificar somente este inicialmente e os métodos de penalidade serem modificados em um refinamento final. Entretanto, o primeiro resultado foi de não convergência para o algoritmo proposto. O resultado de classificação foi extremamente ruim, com todas as classificações sendo feitas como positivas para o visto, levando ao descarte desta opção.

AC_Treino	0.93
AC_Teste	0.93
Precisão	0.0
Recall	0.0
F1	0.0

Figura 15: Resultado da Regressão Logística

Os resultados obtidos com o Grid Search foram gerados a partir do parâmetro "C" otimizado, sendo este:

- C = 0.01

3.2.2 Random Forest

A seguir o modelo escolhido foi o de random forest, é um algortimo bastante conceituado e que se guia pelos conceitos clássicos de árvores de decisão, sendo facilmente rastreável e bastante polivalente, indicado pela própria biblioteca Scitkit learn para solução de datasets com grandes quantidades de registros. Durante o treinamento foram modicados a profundidade máxima, a quantidade máxima de características e por fim a quantidade de arvores máxima. Os resultados foram bastante interessantes, com a acurácia aumentando consideravelmente.

AC_Treino	0.94
AC_Teste	0.94
Precisão	0.2
Recall	0.73
F1	0.31

Figura 16: Resultado do Random Forest

Os resultados obtidos com o Grid Search foram gerados a partir dos parâmetros otimizados, sendo estes:

- max_depth = 40
- max_features = 3
- n_estimators = 20

3.2.3 Gradient Boosting

O próximo modelo escolhido foi o de gradient boosting, por se utilizar de diversos classificadores ”fracos’,’ o gradient boosting consegue resolver problemas complexos e dados de características diversas, visto que não depende de um único tipo de algoritmo para gerar a classificação. Os hiper parâmetros modificados inicialmente foram: Profundidade máxima e taxa de aprendizado, posteriormente o numero de classificadores. O resultado obtido foi bastante interessante, com os seguintes indicadores:

Os resultados obtidos com o Grid Search foram gerados a partir dos parâmetros otimizados, sendo estes:

- learning_rate = 0,05
- max_depth = 16
- n_estimators = 30

AC_Treino	0.94
AC_Teste	0.94
Precisão	0.2
Recall	0.73
F1	0.31

Figura 17: Resultado do Gradient Boosting

3.2.4 Naive Bayes

Por fim o algoritmo de naive bayes também foi testado, embora o problema abordado deixe claro sua correlação entre os dados e, portanto, a dificuldade de aplicação do algoritmo, foi realizado um teste com este a fim de verificar se de fato as percepções obtidas sobre o dataset eram válidas. O resultado assim como esperado foi bastante fraco, com baixa precisão e recall, sendo este descartado também.

Train_ACC	0.93
Test_ACC	0.93
P	0.0
R	0.28
F1	0.0

Figura 18: Resultado do Naive Bayes

O algoritmo de naive bayes não conta com nenhum hiper parâmetro que possa ser utilizado para este tipo de classificação, a função Grid Search só foi utilizada para aplicação de cross validation.

3.2.5 Escolha do Algoritmo Final

A partir dos resultados finais e considerando a tabela abaixo para comparação do desempenho de cada um destes, foi escolhido o algoritmo de Gradient Boosting e Random Forest, porém após mais algum grau de refinamento o algoritmo de Random Forest não evoluiu em resultado, já o Gradient Boosting ao utilizar maior tempo de processamento, ou seja, mais "estimators' obteve resultado superior'. Sendo selecionado finalmente o algoritmo de Gradient Boosting, deixando claro sua versatilidade, devido a utilização dos classificadores fracos, e sua facilidade de aplicação em datases grandes, o torna o melhor algoritmo dos testados para a solução do problema apresentado neste projeto.

Algoritmo	Regressão Logística	Random Forest	Gradient Boosting	Naive Bayes
Acurácia em Treino	93%	94%	94%	93%
Acurácia em Teste	93%	95%	94%	93%
Precisão	0%	26%	20%	6%
Recall	0%	75%	73%	0%
F1	0%	39%	31%	0%

Figura 19: Comparação Entre os Algoritmos Testados

3.3 Refinamento

O algoritmo foi refinado a partir de todos os hiper-parâmetros disponíveis na biblioteca SciKit Learn, inicialmente por meio do grid search e posteriormente por meio de ajustes manuais. Após verificar os parâmetros ideais com o grid search, o numero de iterações, ou seja, de classificadores fracos utilizados foi aumentado e o gráfico de "loss" do treinamento foi plotado, até atingir valores em que o poder computacional necessário para melhora dos resultados não era mais vantajoso.

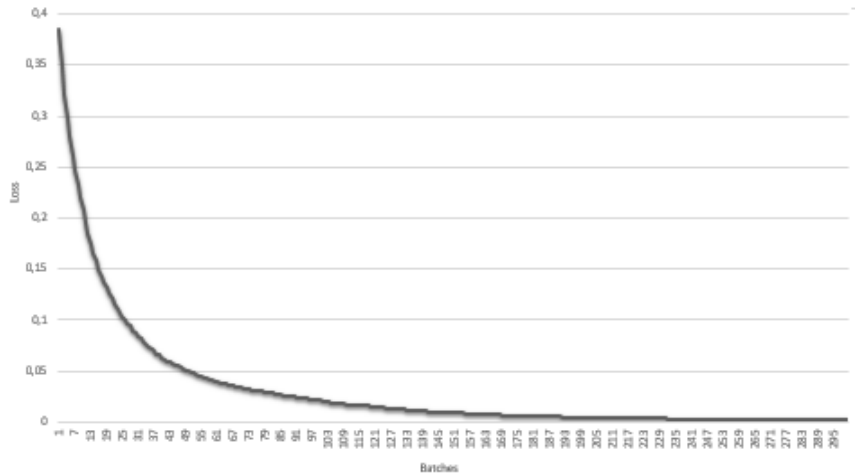

Figura 20: Loss vs Batch

Ocorreram diversas fases de ajustes e ao modificar um valor os outros eram afetados, após grande quantidade de iterações, procurou-se encontrar os ”extremos globais” de treinamento, ou seja, os valores em que o resultado ideal é obtido, sendo que com qualquer alteração para mais ou menos, estes resultados são degradados.

Inicialmente o caminho encontrado foi aumentar somente o valore de ”n_estimators”, porém, com grandes valores destes a profundidade máxima da rede tendia a crescer. Foi percebido que a melhor relação poder computacional vs resultado foi encontrada neste momento com valores altos para ”n_estimators”. Ao aumentar a profundidade máxima, a necessidade de poder computacional para treinamento subiu exponencialmente, porém com muito menos classificadores o resultado já era bem expressivo, sendo assim a estratégia foi modificada e passou-se a modificar a profundidade, com o learning rate ideal já encontrado, e após encontrar a melhor profundidade começar a aumentar os classificadores até encontrar o ponto ideal.

Foi verificado então que os valores de loss inicial não baixavam a partir de uma profundidade máxima com valor 18. Sendo assim, esta foi a profundidade máxima definida e outros valores foram testados para quantidade de classificadores.

4 Resultados

Os resultados obtidos por meio do algoritmo são apresentados na tabela a seguir:

AC_Treino	0.95
AC_Teste	0.95
Precisão	0.28
Recall	0.75
F1	0.41

Figura 21: Resultado do Gradient Boosting

Como pode-se verificar, em relação aos primeiros valores obtidos houve uma considerável melhora, tendo em vista que a quantidade de valores positivos negativos para a negação do visto é muito maior que o de valores positivos, a acurácia dos modelos sempre parace bem alta, entretanto, os valores de recall e precisão tiveram um grande aumento, levando a um valor de F1 bastante razoável para a aplicação.

Os ajustes finais realizados resultaram nos seguintes hiper parâmetros para o algoritmo escolhido:

Modelo Aplicado	Gradient Boosting
Number os Estimators	300
Max Depth	18
Learning Rate	0,1

Figura 22: Hiper Parâmetros do Gradient Boosting

Utilizando o método de cross validation, em que se usa amostras diversas e aleatórias de dados para teste e treinamento, estes resultados são bastante confiáveis e retratam com bastante segurança toda a diversidade do dataset apresentado.

4.1 Avaliação do Modelo e Validação

Conforme é possível observar, o desempenho de acurácia foi bastante expressivo, com grande quantidade de acertos nas predições totais, podemos observar um valor de Recall e Precisão razoáveis e bastante suficientes para a aplicação esperada. É importante ter em mente que a aplicação de predição de vistos permanentes precisa evitar falsos positivos para negação dos vistos, de forma a não desestimular nenhum pedido de visto, a menos que os limites de solicitação estejam muito longe.

4.2 Justificativa

Comparado ao benchmark escolhido, o sistema obteve performance bastante superior, com acurácia por volta de 35% maior, representando uma boa melhora em relação ao modelo empregado.

O dataset foi tratado de forma a utilizar campos gerais, que poderiam ser preenchidos por qualquer pessoa, sendo assim, foi objetivada a eliminação de campos exclusivamente pessoais que facilitariam a predição dentro dos dados de teste (gerando possível overfit), embora tornassem impossível a comparação destes dados em um sistema funcional para prever estes resultados para qualquer pessoa que preenchesse a predição. Dados como ano específico de solicitação, endereço específico de destino nos EUA, entre outros, levariam a grande facilidade na predição, porém eliminariam a possibilidade de utilização do sistema em qualquer esfera prática.

5 Conclusão

5.1 Avaliação Geral do Modelo Escolhido

O modelo escolhido se mostra bastante consistente, embora seja possível melhorar os resultados e considerando a aplicação e seu tipo de utilização, os métodos de ensemble learning quando aplicados para resolução da classificações numéricas em dados tabelados se mantém como uma solução muito boa.

Quanto a utilização da solução em relação a biblioteca SciKit Learn, o algoritmo apresentado tem resultados satisfatórios, embora em outras distribuições possa ser mais refinado. É interessante observar a facilidade na aplicação e principalmente na obtenção de bons resultados por meio do ensemble learning.

É possível comparar visualmente o resultado do algoritmo com os dados de teste definidos no dataset a partir do gráfico a seguir:

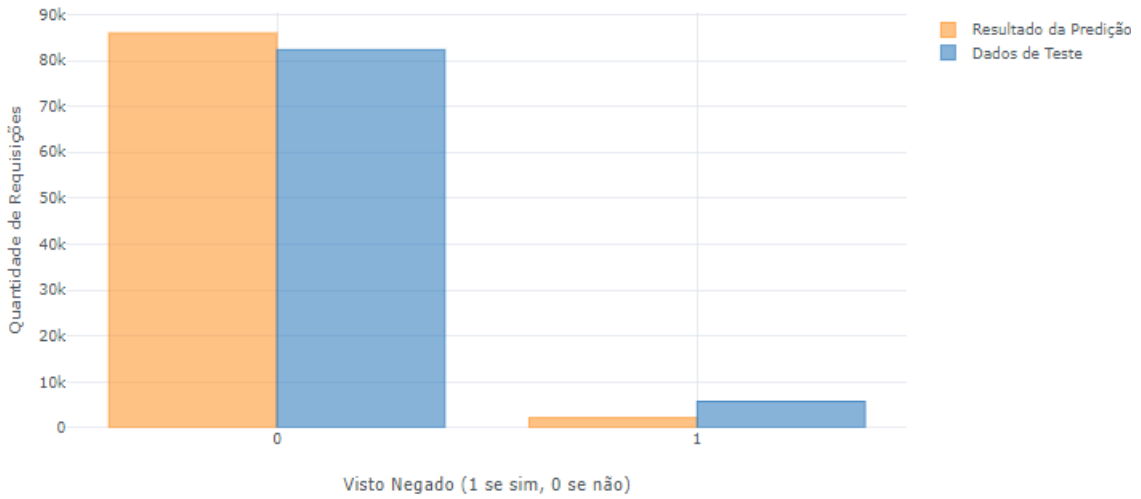

Figura 23: Aprovações em Dados de Teste vs Dados de Predição

Conforme pode-se visualizar pelo gráfico acima, a solução apresenta grande taxa de sucesso na tarefa, visto que os valores se encontram bastante próximos e coerentes para o que se espera de resultado.

De forma a exemplificar esse sucesso foram feitas duas consultas completas com predição, de forma a verificar os dados e o resultado para um certo indivíduo, seguem as tabelas com as duas predições abaixo.

```
Situação Real da Requisição: Denied
Situação preditada pelo algoritmo para a Requisição: Denied
Cidade do Empregador: longview
Nome do Empregador: alpha first dental, p.a.
País de Origem: south korea
Estado do Empregador: TX
Cidade em que irá trabalhar: tyler
Estado em que irá trabalhar: TX
Tipo de Origem do Processo: OES
Salário á receber por ano: 180000.0
```

Figura 24: Exemplo de Predição 1

```
Situação Real da Requisição: Certified
Situação preditada pelo algoritmo para a Requisição: Certified
Cidade do Empregador: edison
Nome do Empregador: larsen & toubro infotech limited
País de Origem: india
Estado do Empregador: NEW JERSEY
Cidade em que irá trabalhar: edison
Estado em que irá trabalhar: NEW JERSEY
Tipo de Origem do Processo: OES
Salário á receber por ano: 93101.0
```

Figura 25: Exemplo de Predição 2

É possível ver a utilização de ferramentas da biblioteca Sci Kit Learn de forma a retornar os dados a sua forma original, sendo possível assim aplicar este algoritmo para qualquer interessado em um visto permanente americano. É possível também revisitar a distribuição de salários vista anteriormente e verificar a assertividade do algoritmo para cada faixa de salário, exibindo então um comparativo de sucesso do algoritmo para cada faixa, conforme o gráfico a seguir:

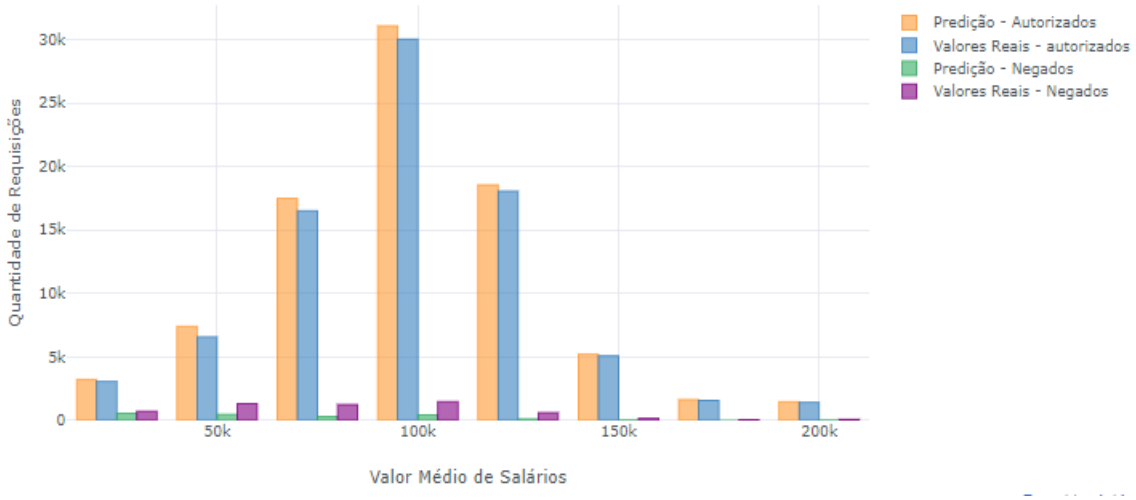

Figura 26: Aprovações Reais e Preditadas por Faixa de Salário

Assim como esperado, fica visualmente claro os 95% de acuracidade do algoritmo, com grandes quantidades de acertos principalmente nas faixas mais baixas e mais altas de salário, a maior parte dos erros de classificação do algoritmo se encontram então nos valores médios de salário, aqueles em que naturalmente se espera uma condição mais difícil de tomada de decisão para o algoritmo.

5.2 Avaliação da Solução

O resultado final da solução tem como principais características a ampla gama de aplicação, visto que os dados foram tratados de forma a serem bastante generalistas, a profissão de cada solicitante foi considerada por meio do sistema de classificação do governo dos EUA, os estados e cidades de destino por meio de sua sigla e zipcode, dados pessoais como salário foram considerados em mesma unidade e facilmente recolhidos de um ”usuário” da solução.

Sendo assim, a solução apresentada buscou, além de obter indicadores de acurácia e precisão altos, também ser de clara aplicação real. Isto foi feito pela forma como os dados foram tratados e também como sua utilização foi delimitada.

A solução então pode ser aplicada em qualquer tipo de produto que envolva a avaliação de permissão de vistos, desde um site de solicitação de vistos, até uma corretora ou empresa que busque facilitar a obtenção deste visto por qualquer solicitante.

5.3 Melhorias

Possíveis melhorias para o problema apresentado podem ser a consideração da utilização de outras bibliotecas de ensemble learning com maior confiabilidade e robustez no mercado como a própria XG Boost, utilização de mais hiper parâmetros complexos para o refinamento, busca por outros métodos de obtenção de dados para o dataset.

É possível também tornar a solução mais robusta por meio da busca na internet por dados incompletos no dataset, ou a utilização de mais camadas de verificação dos dados antes mesmo da predição. Outro elemento passível de melhoria é a utilização de treinamento continuado, com o retreino da solução a partir dos dados inseridos por usuários.

Referências

[1] Dalkhat M. Ediev. (2000). Reconstruction of the US immigration history: demographic potential approach. - *Karachay-Cherkessian State Technological Institute.*

[2] Jacob Boysen. (2017). US Permanent Visa Applications: Detailed information on 374k visa decisions. - *https://www.kaggle.com/jboysen/us-perm-visas*

[3] Wikipedia. (2018). ZIP Code - *https://en.wikipedia.org/wiki/ZIP_Code*

[4] Haykin. (1998). Neural Networks: A Comprehensive Foundation. - *Prentice Hall, New York, NY, USA, 2th edition.*

[5] Ganguly. (2017). Exploratory Data Analysis US Permanent Visa Applications with Feature Analysis - *https://www.kaggle.com/ambarish/eda-us-permanent-visas-with-feature-analysis.*