Projeto Final

I.Definição

Visão geral do projeto

Este projeto analisa dados de registros de reclamações feitas na plataforma consumidor.gov.br entre maio e dezembro de 2014. Esta análise é importante porque procura entender quais aspectos são levados em conta na avaliação do consumidor ao atendimento da empresa. Este tema foi escolhido devido a experiências pessoais desagradáveis com relação ao assunto e pela acessibilidade aos dados. O conjunto de dados é fornecido pelo Portal Brasileiro de Dados Abertos (link para acesso: http://dados.gov.br/dataset/reclamacoes-do-consumidor-gov-br1). Foi fornecida uma planilha em formato Excel com as características do serviço de atendimento.

Descrição do projeto

As reclamações com serviços prestados pelas empresas são constantes, com esses dados em domínio público, os consumidores poderão ter uma projeção de como as empresas tratam seus clientes antes de contratarem um serviço. O projeto tem em vista analisar os principais aspectos dos registros de forma que se identifique um padrão de nota dos consumidores de acordo com serviço prestado, tempo de resposta, faixa etária, entre outros.

Entrada: Dados em planilha Excel, onde cada linha representa um registro de reclamação do consumidor sobre um produto ou serviço adquirido. As colunas descrevem as características do registro, como idade, faixa etária, região e etc.

Saída: Nota do Consumidor

O projeto tem como objetivo construir um modelo de classificação de notas do consumidor, que se desenvolverá nas seguintes etapas:

1. Análise exploratória dos dados, a fim de conhecer os detalhes e extrair informações relevantes;
2. Limpeza dos dados, a fim de criar uma estrutura de dados livre de outliers, valores perdidos e dados irrelevantes;
3. Preprocessamento de dados, a fim de organizar e estruturar os dados para a criação de modelos de classificação;
4. Avaliação e validação dos modelos, a fim de selecionar qual modelo teve um desempenho melhor;

5. Conclusão, a fim de julgar se o modelo final teve resultado satisfatório e se pode servir de benchmark para futuros projetos.

Métricas

Para avaliação do modelo será utilizado o f-score. A avaliação se dá pela seguinte equação:

f-score = **2*((precision * recall) / (precision + recall)**

Onde a precisão indica o número de vezes que uma classe foi predita corretamente, dividida pelo número de vezes que foi predita. A precisão se dá pela seguinte equação:

precision = true positive / (true positive + false positive)

E onde o recall indica o número de vezes que uma classe foi predita corretamente, dividido pelo número de vezes que a classe aparece no dado de teste.

recall = true positive / (true positive + false negative)

II. Análise

Exploração de dados

O conjunto de dados tem um volume de 37153 registros com 24 colunas, os quais carregam as seguintes características: Região, UF, Cidade, Sexo, Faixa Etária, Ano Abertura, Mês Abertura, Data Abertura, Data Resposta, Data Finalização, Tempo Resposta, Nome Fantasia, Segmento de Mercado, Área, Assunto, Grupo Problema, Problema, Como Comprou Contratou, Procurou Empresa, Respondida, Situação, Avaliação Reclamação, Nota do Consumidor, Total.

Das 24 características, 4 são numéricas e 20 são categóricas/object

Informações do conjunto de dados:

Região	37153 non-null object
UF	37153 non-null object
Cidade	37153 non-null object
Sexo	37153 non-null object
Faixa Etária	37153 non-null object
Ano Abertura	37153 non-null int64
Mês Abertura	37153 non-null int64

Data Abertura	37153 non-null object
Data Resposta	36972 non-null object
Data Finalização	37153 non-null object
Tempo Resposta	36972 non-null float64
Nome Fantasia	37153 non-null object
Segmento de Mercado	37153 non-null object
Área	37153 non-null object
Assunto	37153 non-null object
Grupo Problema	37153 non-null object
Problema	37153 non-null object
Como Comprou Contratou	37153 non-null object
Procurou Empresa	37153 non-null object
Respondida	37153 non-null object
Situação	37153 non-null object
Avaliação Reclamação	37153 non-null object
Nota do Consumidor	22407 non-null float64
Total,,,,,,,,,	37153 non-null object

dtypes: float64(2), int64(2), object(20)

Amostra do conjunto:

Região	NE
UF	MA
Cidade	São Luís
Sexo	M
Faixa Etária	entre 21 a 30 anos
Ano Abertura	2014
Mês Abertura	5
Data Abertura	16/05/2014
Data Resposta	16/05/2014
Data Finalização	19/05/2014
Tempo Resposta	0
Nome Fantasia	Tim

Segmento de Mercado	Operadoras de Telecomunicações (Telefonia,Inte...
Área	Telecomunicações
Assunto	Telefonia Móvel Pré-paga
Grupo Problema	Vício de Qualidade
Problema	Funcionamento inadequado do serviço (má qualid...
Como Comprou Contratou	Internet
Procurou Empresa	S
Respondida	S
Situação	Finalizada avaliada
Avaliação Reclamação	Resolvida
Nota do Consumidor	1
Total,,,,,,,,,,	1,,,,,,

Name: 0, dtype: object

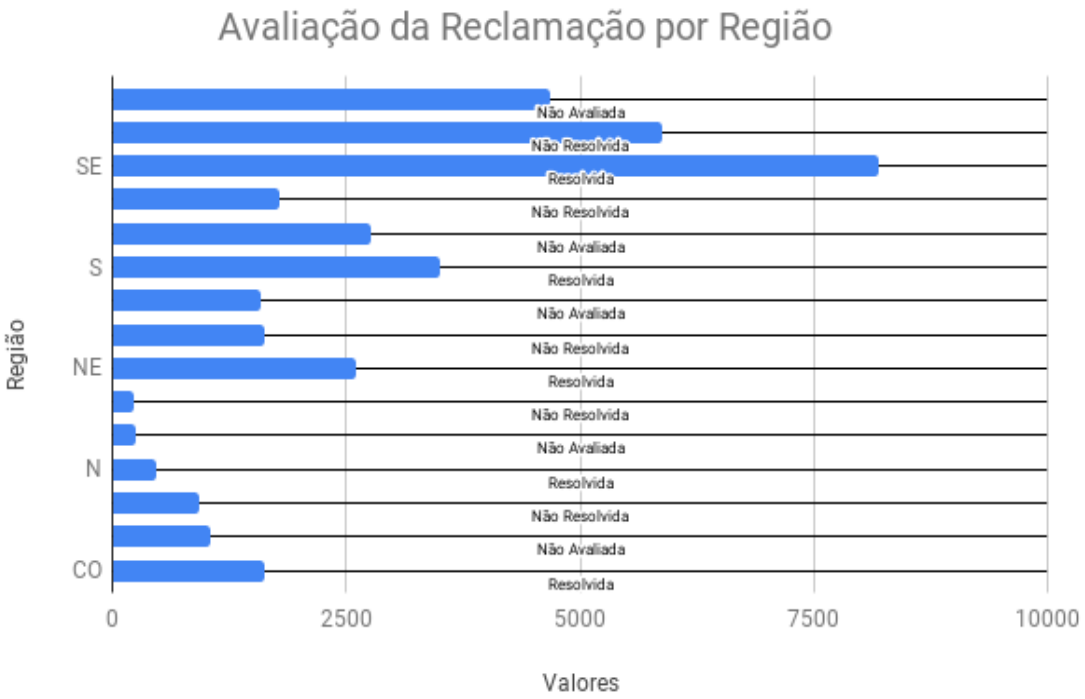
Através da análise exploratória pude tirar algumas conclusões:

1. 1.7% dos dados estavam faltando. Apenas três colunas apresentam valores perdidos: Data resposta, com 181; Tempo resposta, com 181; Nota do Consumidor, com 14746, o que representa 39.7% dos seus dados como ‘valores perdidos’.
2. O tempo de resposta das empresas deveria ser de no máximo 10 dias, 100% dos registros ficaram dentro do prazo. A média de tempo de resposta foi de 6.5897, houve casos em que a resposta aconteceu no mesmo dia. Talvez essa característica possa ter uma correlação forte com a nota do consumidor.
3. 27.8% dos registros não foram avaliados, esses registros certamente não serão considerados no modelo.
4. 63.9% dos registros foram feitos pelo sexo masculino, que tiveram média de 2.93 na nota do consumidor e 36.1% pelo sexo feminino, que tiveram 3.09 de média. Mostrando que as mulheres tendem a dar uma nota maior nas avaliações.
5. Com relação a faixa etária, a quantidade de homens foi superior em comparação as mulheres, exceto na faixa dos acima de 70 anos.

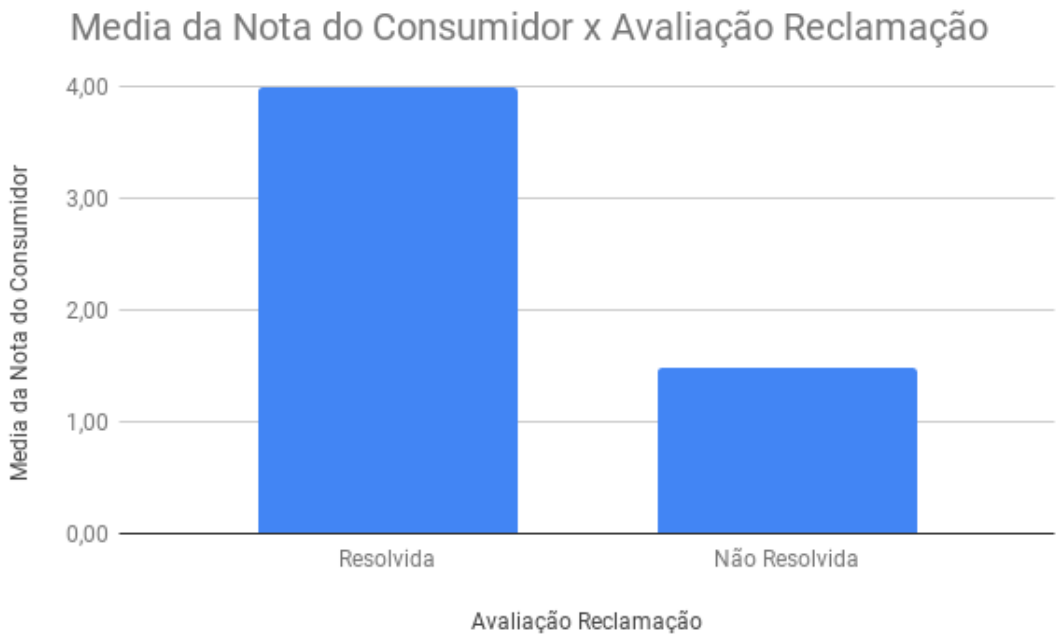
6. Dos 129 valores de problemas, apenas 2 tiveram média de 5.0, curiosamente, esses 2 valores tiveram 100% do registro feitos pelo gênero masculino. Por outro lado, 14 valores de problemas tiveram média 1.0.
7. Dados indicando datas devem estar como datatime.

Visualização Exploratória

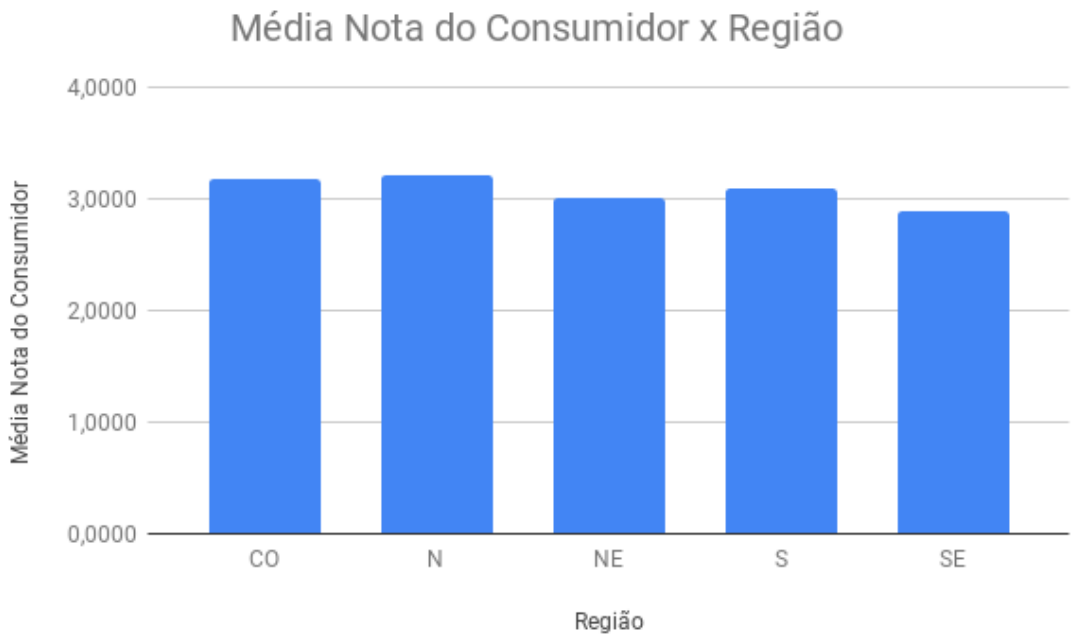
O primeiro gráfico mostra a distribuição da avaliação da reclamação por região, ele indica que a região SE (Sudeste) apresenta o maior volume de registros. Também mostra um certo padrão na distribuição das reclamações, pode-se observar que as regiões se comportam similar. Considero essa informação importante, pois se visualiza um padrão nos dados da região.



O segundo gráfico mostra a média da nota do consumidor em relação ao valor da reclamação (se foi resolvida ou não resolvida). Esse gráfico indica uma tendência, se o problema foi resolvido, a nota do consumidor tende a ser alta. Se o problema não foi resolvido, a nota do consumidor tende a ser baixa.



O terceiro gráfico mostra as médias das notas do consumidor por região. Considerei a informação importante para observar se havia diferença entre as regiões, mas o gráfico informa certa média comum. A Região Norte apresenta a maior média e a região Sudeste apresenta a menor média.



A tabela a seguir mostra a correlação entre as colunas numéricas. E, entre elas, a relação entre o tempo de resposta e a nota do consumidor. Essa informação é

importante, pois derruba uma forte hipótese de que, o tempo de resposta era um fator decisivo na nota do consumidor.

	Ano Abertura	Mês Abertura	Tempo Resposta	Nota do Consumidor
Ano Abertura	NaN	NaN	NaN	NaN
Mês Abertura	NaN	1.000000	-0.082528	0.084351
Tempo Resposta	NaN	-0.082528	1.000000	-0.189658
Nota do Consumidor	NaN	0.084351	-0.189658	1.000000

Algoritmos e técnicas

Para o problema de classificação e para fins didáticos, pretendo utilizar mais de 1 (um) algoritmo de classificação, podendo assim comparar o desempenho entre eles. Utilizarei 2 (dois) algoritmos:

1. **Gradient Tree Boosting:** Um modelo de aprendizagem supervisionada, que usa Árvore de decisão como base. O modelo faz uso de weak learners para prever/classificar os dados. No primeiro passo do modelo, o weak learner tenta prever/classificar os dados, a predição/classificação é comparada com o y (variável objetivo) e o erro residual é computado. No próximo passo, o weak learner tenta prever/classificar os resíduos, a fim de criar um modelo que generalize e diminua o erro residual. O modelo tende a repetir esses passos, até que o erro seja o mínimo possível. O modelo possui como características a potência preditiva e a não tendência à overfitting. Ele pode ser considerado uma boa escolha, pois métodos ensemble são classificadores de alta qualidade e dão a possibilidade de trabalhar com modelos multiclasses.
2. **Linear Support Vector Classification (LinearSVC):** Um modelo de aprendizagem supervisionada, que usa support vector machine como base. O modelo tem como objetivo criar um hiperplano (linha) que separam os dados de acordo com as suas características. Uma característica desse modelo são as linhas de suporte, que marcam a distância entre o hiperplano e os dados, quanto

maior é a margem (distância entre o hiperplano e os dados), melhor é o modelo. O LinearSVC se sai muito bem quando temos uma grande quantidade de características. O arquivo escolhido não tem grande quantidade de características, porém precisarei transformar características não numéricas em numéricas, o que pode aumentar bastante o número de colunas. O algoritmo em referência pode ser considerado uma boa escolha, pois possibilita trabalhar com modelos multiclasses e com grande quantidade de características na tabela.

Benchmark

O benchmark de referência para classificação do modelo será um classificador naive. Utilizarei o classificador MultinomialNB, por ser um modelo multiclasse. Esse recurso foi necessário, pois não encontrei trabalhos anteriores para fazer a comparação com o modelo que irei desenvolver.

III. Metodologia

Pré-processamento de dados

1. Fiz 1 (uma) cópia dos dados;
2. Removi os registros com dados que estavam faltando;
3. Removi as colunas Ano Abertura e Total, que se mostraram irrelevantes para o desenvolvimento do projeto;
4. Objetivando criar um modelo que generalize para os segmentos, removi a coluna Nome Fantasia do dataframe;
5. Para otimizar os dados das colunas Data Abertura, Data Resposta e Data Finalização, transformei as informações de formato dd/mm/YYYY em dia da semana;
6. Para otimizar os dados das colunas Data Abertura, Data Resposta e Data Finalização, transformei as informações de formato dd/mm/YYYY em mês do ano;
7. Após executar a conclusão ‘6’, as colunas Respondida e Situação não apresentaram variação nos dados, removi essas colunas pois se mostraram irrelevantes para o desenvolvimento do projeto;
8. Removi as colunas Data Abertura, Data Finalização, Data Resposta. Após executar a conclusão ‘5’ e ‘6’, essas informações foram diluídas em outras colunas;
9. Separei a coluna que servirá como label, das colunas que são features;

10. Escalei os dados das colunas numéricas utilizando a função MinMaxScaler();
11. Transformei as colunas com valores categóricos em valores numéricos, utilizando a função get_dummies();
12. Transformei a coluna label em int;
13. Separei os dados em teste e treino, utilizando o método cross validation. Os dados de treino somam 80% e os de teste, 20% dos registros;
14. Fiz o processo de seleção de 50% das features, utilizando a função SelectPercentile();

Implementação

Na primeira parte da implementação, criei um classificador naive utilizando MultinomialNB(), a fim de servir como uma base de comparação para os modelos que serão criados. Utilizei esse modelo, pois atende a necessidade de classificar multiclasses.

A implementação do modelo se deu na seguinte forma:

1. Importei o classificador MultinomialNB() da classe naive_bayes;
2. Importei a métrica de avaliação f1_score da classe metrics;
3. Criei o modelo MultinomialNB() e treinei com as variáveis X_train_selected e y_train;
4. Utilizei o modelo criado para prever o y_test através da variável X_test_selected;
5. Utilizei o método de avalição f1-score, com os parâmetros y_test, predicted e average="micro", que retornou o valor de 0.555331.

Para a criação dos modelos Gradient Tree Boosting e LinearSVC, foi necessária a utilização do algoritmo OneVsRest para multiclassificação. O referido algoritmo consiste em ajustar um classificador por classe, ou seja, para cada classificador, será ajustada uma classe contra as demais, possibilitando assim a classificação de multiclasses.

Após a implementação do benchmark, criei um classificador Gradient Boosting, da seguinte forma:

1. Importei o classificador GradientBoostingClassifier() da classe ensemble;
2. Importei o classificador OneVsRestClassifier() da classe multiclass;
3. Criei o modelo OneVsRestClassifier tendo como estimator um modelo GradientBoostingClassifier() com o parâmetro random_state=101;
4. Treinei o modelo com as variáveis X_train_selected e y_train;
5. Utilizei o modelo criado para prever o y_test, através da variável X_test_selected;

6. Utilizei o método de avalição f1-score, com os parâmetros y_test, predicted e average="micro", que retornou o valor de 0.584453.

E então, criei um classificador LinearSVC, a fim de comparar a performance com o classificador Gradient Boosting. A implementação se deu da seguinte forma:

1. Importei o classificador LinearSVC() da classe svm;
2. Criei o modelo OneVsRestClassifier tendo como estimator um modelo LinearSVC() com o parâmetro random_state=101;
3. Treinei o modelo com as variáveis X_train_selected e y_train;
4. Utilizei o modelo criado para prever o y_test, através da variável X_test_selected;
5. Utilizei o método de avalição f1-score, com os parâmetros y_test, predicted e average="micro", que retornou o valor de 0.571237.

Houveram várias dificuldades na implementação dos modelos, a maior parte delas foi pela falta de conhecimento em criar modelos multiclassse. Durante o desenvolvimento da proposta de projeto, indiquei que iria utilizar os modelos Adaboost e SVM, porém durante a implementação, tomei conhecimento que esses modelos não atendem problemas de multiclasse. Após consultar o sklearn através do link https://scikit-learn.org/stable/modules/multiclass.html, escolhi os modelos Gradient Boosting e LinearSVC para continuar com o padrão de 1(um) modelo da classe ensemble e 1(um) modelo da classse SVM.

A implementação do OneVsRestClassifier teve um grau de dificuldade baixo, a única dificuldade foi entender o processo utilizado pelo algoritmo para fazer o processo de classificação multiclasses e entender a diferença em relação algoritmo OneVsOne.

Com relação as métricas de avaliação, a dificuldade em utilizar o f1-score para multiclasses foi em entender como se comporta o parâmetro average, necessário para multiclasse.

Refinamento

O refinamento do modelo GradientBoosting foi feito através do algoritmo de GridSearchCV(). Utilizei o algoritmo de aprimoramento junto com o modelo OneVsRestClassifier, tendo o GradientBoosting como estimator. Para o aprimoramento, utilizei os seguintes parâmetros 'estimator__learning_rate': [0.1, 0.5, 1], 'estimator__n_estimators': [50,60,70], 'estimator__min_samples_split': [2,5,8], 'estimator__min_samples_leaf': [30,40,50], 'estimator__max_depth': [3,5,8], 'estimator__subsample': [0.5,0.8]. Os melhores parâmetros gerados pelo GridSearchCV foram learning_rate = 0.1, n_estimators = 60, min_samples_split = 2, min_samples_leaf

= 40, max_depth = 5, subsample = 0.8. Utilizei os melhores parâmetros para criar um modelo GradientBoosting, treinei o modelo e utilizei a métrica de avaliação f1-score, que retornou o valor de 0.585125.

O refinamento do modelo LinearSVC foi feito através do algoritmo de GridSearchCV(). Utilizei o algoritmo de aprimoramento junto com o modelo OneVsRestClassifier, tendo o LinearSVC como estimator. Para o aprimoramento, utilizei o seguinte parâmetro 'estimator__C': [0.1,0.5,1]. O melhor parâmetro gerado pelo GridSearchCV foi C = 0.1. Treinei o modelo criado e utilizei a métrica de avaliação f1-score, que retornou o valor de 0.579077.

IV. Resultados

Modelo de avaliação e validação

Para avaliação, utilizei a métrica f1-score. A tabela abaixo mostra a comparação entre os classificadores e os f1-scores resultantes. Pode-se perceber uma leve vantagem do classificador GradientBoosing em relação aos demais.

Classificador	**F1-Score**
GradientBoosting	0.584453
LinearSVC	0.571237

Após o refinamento de parâmetros, utilizei a mesma métrica para avaliar os modelos melhorados. A tabela abaixo mostra a comparação entre os classificadores e os f1-scores resultantes. Pode-se observar que o GradientBoosting continua com o f1-score maior que o LinearSVC, mas pouco melhorou com relação ao classificador sem o refinamento.

Classificador	**F1-Score**
LinearSVC	0.579077
GradientBoosting	0.585125

Pelo fato do maior valor do F1-Score, o modelo escolhido foi o GradientBoosting.

Justificativa

Ao comparar o modelo selecionado com o modelo de benchmark, pode-se observar a melhora de 0,029794 no F1-Score. Isso representa uma melhora de 5.37%.

Apesar da melhora do modelo, não considerei o resultado muito expressivo. O f1-score não está satisfatório o suficiente para resolver o problema proposto no projeto.

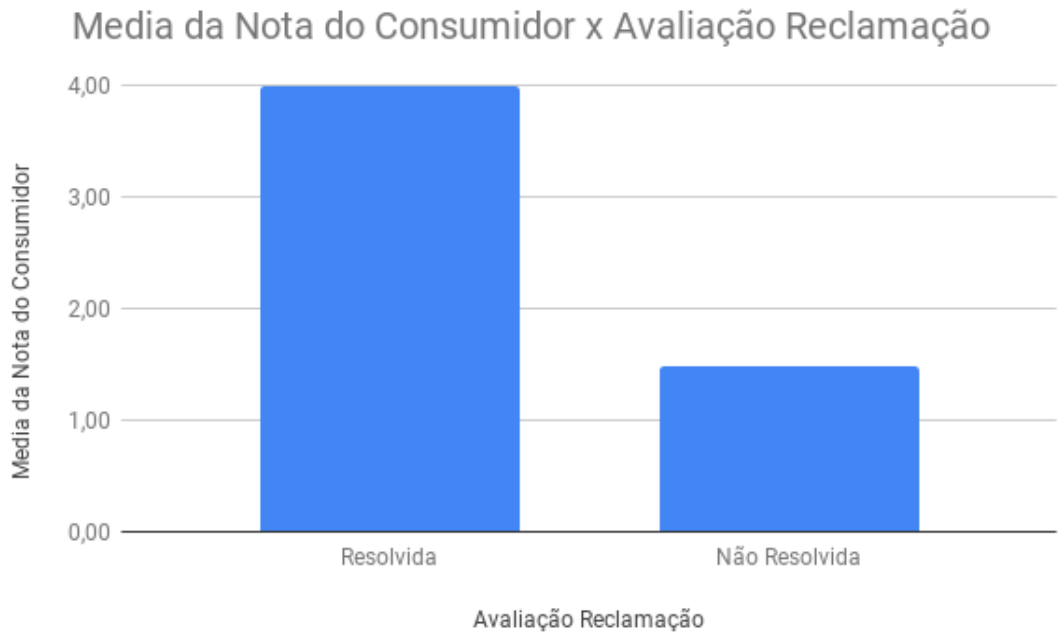
V. Conclusão

Forma livre de visualização

A tabela a seguir mostra as 3 features mais importantes gerados pelo algoritmo de seleção de features. O que se pode observar é que 2 features tem um valor totalmente discrepante comparado com o resto.

Features	**Importância**
Avaliação Reclamação_Resolvida	5.789,411814
Avaliação Reclamação_Não Resolvida	5.789,411814
Tempo Resposta	172,192742

Em complemento, o gráfico Média da Nota do Consumidor x Avaliação Reclamação, gerado durante a análise exploratória, mostra que a média de nota dos consumidores cujo problema foi resolvido é bem superior aos que não tiverem o problema resolvido.



Essas informações reforçam a hipótese de que, os consumidores só levaram em consideração o resultado do atendimento, e não a forma do atendimento, como avaliação.

Reflexão

A concepção do projeto foi baseada em experiências negativas com prestações de serviços de empresas do ramo tecnológico e de telecomunicação. O acesso aos dados foi através do site do governo, onde utilizei os dados referentes ao ano de 2014.

O projeto teve como objetivo criar um modelo de classificação, levando em consideração as características dos registros. A primeira etapa do desenvolvimento foi uma análise exploratória a fim de entender melhor os dados e levantar algumas hipóteses. A segunda etapa foi limpar os dados e organiza-los, criando assim um dataframe com dados mais consistentes e relevantes. A terceira etapa foi o preprocessamento dos dados, as principais ações nessa etapa foram a separação dos dados de label das features, o redimensionamento das features numéricas, a transformação das features não numéricas em numéricas, a separação dos dados de treino e de test, e a seleção das features.

A quarta etapa foi a criação do modelo benchmark e dos modelos de classificação GradientBoosting e LinearSVC, nessa etapa ocorreu a utilização do OneVsRestClassifier, treinamento dos modelos, seus testes e avaliação com a métrica f1-score. A ultima etapa foi o refinamento dos parâmetros dos modelos, as principais ações nessa etapa foram a utilização OneVsRestClassifier, utilização do algoritmo GridSearchCV, treinamento dos modelos, teste dos modelos utilizando os parâmetros encontrados pelo GridSearchCV, seus testes e avaliação com a métrica f1-score.

O modelo final ficou abaixo das minhas expectativas, gostaria de ter gerado um modelo mais preciso. Talvez com dados mais detalhados do atendimento, como clareza nas informações, tratamento ao cliente e solicitude do atendente, possam enriquecer os registros para gerar melhores modelos.

Ao meu ver, esse foi o projeto mais estimulante que eu fiz no curso e o mais desafiador. Ter a experiência de desenvolver o projeto inteiro, desde de a concepção e coleta de dados até criação e avaliação dos modelos foi um processo interessante e motivador.

Tive muita dificuldade em trabalhar com multiclasses. Nunca tinha desenvolvido um modelo para esse tipo de problema. Encontrar conteúdo de como utilizar o OneVsRestClassifier não foi uma tarefa difícil, o maior problema foi no processo de refino dos parâmetros, pois o modelo não estava compreendendo quais parâmetros eram do classificador e quais eram do GridSearchCV. Após muita leitura, consegui referenciar de forma correta e dar continuidade ao projeto.

Melhorias

A fim de tornar o modelo final mais eficiente, considero aumentar o número de dados de entrada. Para o projeto, fiz o uso apenas dos dados de 2014. Outra melhoria consiste em adicionar novas features ao modelo, geradas por mim ou provenientes dos registros de diferentes anos.

Penso também em utilizar outros algoritmos, como o automl. Considerei utiliza-lo no projeto, porém não soube implementa-lo. Também quero utilizar o outro algoritmo multiclasse OneVsOneClassifier e ver como ele se comportaria.