UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS

Gabriel Monteiro de Souza

Aprendizado de máquina aplicado para a predição dos *prospects* mais propensos à compra

Florianópolis
2018

Gabriel Monteiro de Souza

Aprendizado de máquina aplicado para a predição dos *prospects* mais propensos à compra

Relatório submetido à Universidade Federal de Santa Catarina como requisito para a aprovação na disciplina **DAS 5511: Projeto de Fim de Curso** do curso de Graduação em Engenharia de Controle e Automação.

Orientador: Prof. Jomi Fred Hubner

Florianópolis
2018

Gabriel Monteiro de Souza

Aprendizado de máquina aplicado para a predição dos *prospects* mais propensos à compra

Esta monografia foi julgada no contexto da disciplina DAS5511: Projeto de Fim de Curso e aprovada na sua forma final pelo Curso de Engenharia de Controle e

Automação.

Florianópolis, 30 de julho de 2018

Banca Examinadora:

Leandro Costa Schmitz
Orientador na Empresa
Flex - Relacionamento Inteligentes

Prof. Jomi Fred Hubner
Orientador no Curso
Universidade Federal de Santa Catarina

Thiago Lima Silva
Avaliador
Universidade Federal de Santa Catarina

João Henrique Renon Heinzen
Debatedor
Universidade Federal de Santa Catarina

Jean Caetano Perufo Damke
Debatedor
Universidade Federal de Santa Catarina

A todos os colegas que buscam aprender constantemente e não se limitam a sua área de atuação.

AGRADECIMENTOS

Gostaria de agradecer, primeiramente, à Deus, que me possibilita estar aqui hoje, e à minha família, que esteve sempre presente, dando todo o suporte às minhas decisões, bem como me incentivando a buscar e alcançar sempre mais.

À toda a equipe da Flex pela oportunidade, ensinamentos, desafios e feedbacks durante a jornada, em especial a Leandro Schmitz, que me apoiou na realização do projeto.

Ao Professor Jomi Fred Hübner por não só concordar em ser meu orientador, mas também estar sempre disponível.

Ao Departamento de Automação e Sistemas e à Universidade Federal de Santa Catarina por proporcionarem uma educação superior de alta qualidade e por me auxiliarem no meu desenvolvimento profissional e pessoal.

À empresa júnior do curso, Autojun, que me proporcionou o início na carreira de gestão e trouxe ensinamentos essenciais para minha vida profissional.

Aos meus colegas de estudos, Bruno Carvalho, Bruno Marcon, Otávio Trentin, Paulo Pizzolatti e, em especial, Guilherme Hammes, que sempre me ajudaram e me estimularam não só nos estudos, mas também na busca por novas oportunidades e aprendizados.

“let us never regard a question as exhausted, and when we have used our last argument, let us begin again, if need be, with eloquence and irony”

(Pierre-Joseph Proudhon)

RESUMO

A Flex trabalha, principalmente, com vendas através do telefone. A empresa recebe de um cliente uma lista que contém dados de potenciais compradores, chamados de *prospects* ou *leads*. Há uma área chamada *Command Post* na qual há um analista que, de acordo com os dados históricos, classifica e cria regras para definir os perfis mais propensos a comprar. Assim, é priorizado o contato com essas pessoas, as quais podem ser encaminhadas para os melhores vendedores, aumentando a chance de venda. Uma dificuldade é que para um ser humano é complexo associar diversas variáveis, fazendo com que haja uma generalização exagerada e sejam realizadas algumas classificações enviesadas. A abordagem do projeto consistiu em utilizar aprendizado de máquina para analisar os dados históricos e classificar os *prospects* com a sua propensão à compra, pois como o algoritmo associa diversas variáveis a classificação seria melhor e mais eficiente. Foram realizados testes com diversos algoritmos, porém, apesar de possuírem uma performance melhor que a de uma pessoa, ainda há diversos casos que não são identificados corretamente pelo algoritmo.

Palavras-chave: Predição de compradores. Aprendizado de Máquina. Vendas.

ABSTRACT

Flex is mostly a phone-sales company. The company's clients sends over a list with potential buyers (also called prospects or leads) and some information about them. There's a department in the company called Command Post, in which an analyst classifies and creates rules defining the profiles that are more likely to buy according to their buying records. Therefore, the contact with these leads is prioritized. Also, these people are assigned to the best salesmen, increasing the sale chance. However, it is difficult for a human being to associate many variables, so there can be an exaggerated generalization and some of the clients' classifications can be misplaced. The project's approach consisted in using a machine learning technique to analyze historical data and more efficiently classificate prospects according to their buying probability, since the algorithm is capable of associating numerous variables. Many algorithms were tested and their performances were better than a person's, but there are still many cases that are not correctly identified by the algorithm.

Key-words: Predictive Lead Scoring. Machine Learning. Sales.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

RNA – Rede Neural Artificial

ML – Machine Learning (Aprendizado de Máquina)

DL – Deep Learning (Aprendizado Profundo)

CRM – Customer Relationship Management (Gerenciamento de relacionamento com o cliente)

SUMÁRIO

CAPÍTULO 1: INTRODUÇÃO

Esta monografia trata de um projeto realizado pelo acadêmico Gabriel Monteiro de Souza da Universidade Federal de Santa Catarina na Flex - Relacionamentos Inteligentes, empresa que fornece soluções para o relacionamento de empresas com os seus clientes, podendo ser intermediadora dos contatos ou fornecer a tecnologia e plataforma para que a própria empresa possa contatar seus clientes.

Um dos principais produtos da empresa é a intermediação de vendas, onde a Flex recebe uma lista de potenciais compradores e entra em contato com esses para vender algum produto.

Com o aumento das listas de potenciais clientes, devido ao maior acesso a informações por parte das empresas e o maior acesso da população à rede de telefonia, buscaram-se soluções para melhorar os resultados. Uma forma de buscar uma maior efetividade nos contatos encontrada pela empresa foi criar uma área, chamada de *Command Post*, que é responsável por analisar a lista de potenciais compradores e identificar os que são mais propensos à compra, assim é priorizado o contato com estes para se obter um melhor resultado.

A motivação do projeto vem de os analistas da área analisarem os dados históricos e criarem regras para definir o perfil mais propenso a comprar um produto. Uma dificuldade é que para um humano é complexo associar diversas variáveis, fazendo com que haja uma generalização exagerada dos perfis e sejam realizadas algumas classificações enviesadas. Por exemplo, para uma venda de seguros, uma mulher com mais de 40 anos é classificada como de alta propensão, porém, é uma classificação muito genérica e engloba diversos casos que não são propensos à compra.

Foi identificado como problema que há diversos casos identificados como de alta propensão à compra pelos analistas que não efetuavam a compra, fazendo com que fossem investidos recursos, de pessoas e tecnologia para ser feito o contato e não houvesse um retorno adequado.

O objetivo do trabalho é conseguir segmentar melhor a lista de potenciais compradores para que os esforços possam ser mais focados nos que possuem um maior potencial e tornando as abordagens sejam mais efetivas.

A abordagem adotada no projeto foi utilizar técnicas de aprendizado de máquina para conseguir prever o potencial comprador. No decorrer do projeto foi observado que havia uma quantidade muito maior de dados de potenciais clientes que não compravam o produto do que os que compravam, fazendo com que houvesse um desbalanceamento entre as classes e necessitasse de um tratamento diferenciado dos dados para levar essa característica em consideração.

Todas as etapas do projeto foram realizadas pelo acadêmico, recebendo auxílio da área de report center, que é responsável por relatórios, para obtenção de relatórios com os dados dos abordados e compras efetuadas.

No próximo capítulo é apresentada a empresa onde o estudante realizou o projeto, sua história e mercado que está inserida. No terceiro capítulo serão explicados os conceitos teóricos e as ferramentas utilizadas no projeto. Ao quarto capítulo é aprofundado sobre o problema abordado e o projeto proposto. No capítulo 5 é explicitada a execução do projeto, demonstrando as abordagens utilizadas. No sexto capítulo é comentado sobre os resultados obtidos com o projeto e no sétimo capítulo é apresentada a conclusão sobre a implementação do projeto.

CAPÍTULO 2: FLEX - RELACIONAMENTOS INTELIGENTES

Neste capítulo será apresentada a empresa Flex - Relacionamentos Inteligentes, local onde o acadêmico desenvolveu o projeto. O logo da empresa pode ser observado na Figura 1.

Figura 1 - Logo da Flex Relacionamentos Inteligentes

A Flex foi fundada em julho de 2009 em Florianópolis, crescendo rapidamente a cada ano, e já no primeiro ano de empresa atingiu um EBITDA positivo. Atualmente possui 15 unidades, sendo 7 em São Paulo, 3 em Florianópolis, 2 em Lages, 1 em Palhoça, 1 em Xanxerê, e 1 em Engenheiro Coelho. Desde o início o principal produto da empresa é a terceirização do relacionamento de empresas com os seus clientes, realizando vendas, análise e concessão de crédito, cobrança, entre outros serviços, observando o futuro do setor, investe em novas ferramentas e tecnologias para fornecer novas soluções, adequadas a cada perfil de cliente.

Buscando sempre inovar, a Flex criou em 2016 o XLab, o laboratório de inovação da Flex, o logo do laboratório de inovação é apresentado na Figura 2. O projeto explorado nesta monografia foi idealizado durante a vivência do acadêmico na área de inovação da empresa.

Figura 2 – Logo do Laboratório de Inovação XLab

Foi no laboratório de inovação que foi criada a solução Zaas, que é uma solução que fornece toda a tecnologia para que uma empresa que não deseja terceirizar os contatos com seus clientes tenha diversidade de formas de contato com seus clientes e com qualidade.

A empresa possui também alguns chat-bots que realizam todo o processo de negociação de uma dívida sem que haja a necessidade de um humano intermediando.

A empresa valoriza muito os profissionais, sempre buscando a aperfeiçoamento dos mesmos, em 2014 inaugurou o primeiro polo universitário, para oportunizar o ensino superior e pós-graduação aos colaboradores a um preço acessível, a partir daí foram inaugurados polos em todas as cidades que a Flex está presente. Há também diversos cursos de capacitação que são realizados para todos os níveis da empresa, sempre buscando o desenvolvimento dos profissionais.

Em 2015 a Flex adquiriu o grupo RR, o segundo maior grupo de cobrança do Brasil e adicionou a cobrança aos seus segmentos atendidos, concluindo todo o ciclo de relacionamento, podendo oferecer uma solução full service aos seus clientes.

Em 2018 foi concluído o processo para listar a empresa na bolsa de valores de São Paulo, a B3, e agora se prepara para em breve abrir capital e poder angariar fundos para crescer cada vez mais.

CAPÍTULO 3: FERRAMENTAS E TÉCNICAS UTILIZADAS

Neste capítulo as principais ferramentas utilizadas para a resolução dos desafios do projeto serão brevemente descritas e comentadas, a fim de auxiliar o entendimento do trabalho realizado.

3.1 Excel

O Excel é um editor de planilhas desenvolvido pela Microsoft comumente utilizado para manipular e apresentar dados relevantes para o andamento de negócios em empresas. Por ser um software voltado para a análise de dados numéricos, é muito simples e rápido gerar gráficos e implementar funções.

Devido ao conhecimento prévio da ferramenta e por a empresa possuir acesso ao sistema, optou-se por essa ferramenta para compilar e realizar o tratamento inicial dos dados.

3.2 Aprendizado de Máquina

Há dois métodos de aprendizado de máquina: o supervisionado e o não supervisionado. O método supervisionado, escolhido para o presente trabalho, é comumente usado em aplicações nas quais os dados históricos preveem prováveis acontecimentos futuros. Nele, o modelo é treinado usando exemplos rotulados como entrada e a saída desejada já conhecida. Dentro do método supervisionado, há vários tipos específicos de aprendizados de máquina como sendo os principais: regressão e classificação [1].

No presente trabalho, o tipo escolhido foi o supervisionado por classificação sendo que as entradas foram as variáveis de uso da plataforma pelos usuários, e a saída é a classificação de se o cliente é um cliente que tem grande chance de cancelar o serviço.

Em algoritmos de aprendizado de máquina do tipo classificação, pode ocorrer problemas quando existem muito menos casos de algumas classes do que de outras, quando há um desbalanceamento [2].

Os algoritmos de classificação desenvolvidos com métodos que não consideram o desbalanceamento das classes, tendem a valorizar classes predominantes e a ignorar classes de menor representação – conhecidas também como classes raras [2].

Os classificadores gerados a partir de bases de treinamento desbalanceadas apresentam altas taxas de falsos negativos para as classes raras, o que é problemático quando a classe de interesse é classe rara. A fim de evitar esse viés, são utilizadas técnicas de amostragem para um rebalanceamento das classes [3].

Há três principais métodos de amostragem para a resolução dessa questão:

- Subamostragem - eliminação de casos da classe majoritária;
- Superamostragem - replicação de casos da classe minoritária dos dados de treinamento visando obter classificadores melhores do que os obtidos a partir da distribuição original;
- Superamostragem por geração sintética – replicação da classe minoritária sendo que as novas amostras são baseadas na interpolação de instâncias das classes minoritárias.

A remoção de observações com a subamostragem pode fazer com que os dados de treinamento percam informações importantes pertencentes à classe majoritária [3].

Já a simples replicação de casos positivos (instâncias de dados associados à classe de interesse), pode produzir classificadores muito específicos para os casos replicados e com baixo poder de generalização para outros casos positivos [3].

Por fim, a superamostragem por geração sintética evita o baixo poder de generalização. Contudo pode apresentar o efeito indesejável de criação de casos positivos que invadem o espaço de decisão da classe negativa. Essa característica, denominada sobreposição de classes, tende a degradar o desempenho de classificadores obtidos a partir de tais dados [3].

Uma das técnicas para amenizar o problema do desbalanceamento é a SMOTE, que é uma técnica de superamostragem por geração sintética, que cria

exemplos sintéticos através da interpolação dos exemplos próximos, criando um novo exemplo que contêm as características dos exemplos que o geraram [4].

Entre as técnicas de aprendizado de máquina há a rede neural artificial (RNA), que é composta por neurônios artificiais, que realizam operações com as entradas que recebe, considerando um peso para cada entrada e produz uma saída. Um conjunto de neurônios forma uma camada e essas camadas são conectadas, sendo a saída de uma a entrada da outra [5].

Dentre os tipos de RNA há a rede de multicamadas, onde há uma camada de entrada, uma ou mais intermediárias e uma camada de saída.

Cada neurônio da RNA possui uma função de ativação, que é a relação entre a entrada e a saída dele, ela é utilizada com dois propósitos, limitar a saída do neurônio e introduzir não-linearidade no modelo [5].

Uma das funções de ativação é a função ReLu, uma função simples, que é zero no caso de entrada negativa e o próprio valor caso positiva. Essa função é uma das mais utilizadas em RNAs, essa simplicidade que impulsionou o grande crescimento do uso de RNAs [1].

3.3 Python

Python é uma linguagem de programação open source, sem custos para utilização, ela é considerada uma linguagem de alto nível, por ser mais simples e intuitiva de programar através dela [6].

O Python foi escolhido para esse projeto por possuir diversas bibliotecas já prontas e testadas que facilitam o processamento e análise de dados. Além disso o estudante já possuía prática com a linguagem, facilitando o desenvolvimento.

3.3.1 Scikit-Learn

O Scikit-Learn é uma biblioteca open source para Python muito conhecida, ela é colaborativa e foi criada para facilitar o uso de técnicas de aprendizado de máquina por pessoas que não são especialistas, transformando o uso das técnicas em uma linguagem mais simples, de alto nível. Ela apresenta ferramentas para mineração, análise e validação de dados, cobrindo uma vasta gama de métodos

estatísticos e de aprendizado de máquina. Possui a grande maioria dos algoritmos para classificação, regressão e clustering, como random forests, gradient boosting e K Nearest Neighbors, entre outros [7].

3.3.2 NumPy

O NumPy é uma biblioteca para a linguagem Python, que permite trabalhar com arranjos, vetores e matrizes de N dimensões. Ele provê diversas funções e operações sofisticadas, principalmente para manipulação de matrizes. Ele é utilizado para realizar as operações que processam os dados [8].

Alguns de seus principais recursos são:

- Objetos e métodos para vetores N-dimensionais;
- Ferramentas úteis para álgebra linear, transformações de Fourier e manipulação de números randômicos.

3.3.3 Pandas

O Pandas é uma biblioteca open source para Python, ela possui diversas funções para a manipulação e pré-processamento de dados, ela é otimizada e fácil de ser utilizada. Ela é um projeto colaborativo de diversas pessoas no mundo inteiro que buscam aperfeiçoar a biblioteca [9].

Alguns de seus principais recursos são:

- Tratamento de dados nulos;
- Redimensionamento de DataFrames e objetos de maior dimensão;
- Alinhamento automático entre os dados e os cabeçalhos;
- Agrupamento, subsetting, reshaping e pivoting de dados;
- Diversos formatos de entrada, como Excel, CSV, bancos de dados, entre outros.

CAPÍTULO 4: O PROBLEMA E PROJETO PROPOSTO

Durante o primeiro semestre de 2018, o acadêmico desenvolveu um projeto na Flex - Relacionamentos Inteligentes. Neste capítulo, o projeto será contextualizado e apresentado.

A Flex é uma empresa que fornece soluções para o relacionamento de empresas com os seus clientes, podendo ser intermediadora dos contatos ou fornecer a tecnologia e plataforma para que a própria empresa possa contatar seus clientes.

Quando a empresa intermedia vendas, a Flex recebe uma lista de potenciais compradores, chamados de *leads* ou *prospects*, e entra em contato com esses para vender o produto determinado.

Em um contexto e cenário extremamente competitivo no qual as empresas estão inseridas, é fundamental ter dados de qualidade com o propósito de apoiar as decisões tomadas que buscam novas oportunidades ou formas de resolução dos seus problemas.

Na busca pela melhoria do resultado, observando esse cenário competitivo, principalmente buscando ser mais assertiva nas tentativas de venda, e observando que há um maior acesso a informações por parte das empresas, a empresa criou uma área, chamada *Command Post*, que objetiva aumentar a efetividade dos contatos, buscando encontrar padrões entre os *prospects* abordados e definir os perfis de *prospects* que são mais propensos à compra, assim pode ser priorizado o contato com estes e é obtido um melhor resultado.

Nesta área há analistas que analisam os dados históricos de *leads* que efetuaram a compra e *leads* que não efetuaram, com as suas variáveis que o qualificam, identificam semelhança entre eles para poder agrupá-los e segmentá-los em listas com *leads* mais propensos a comprar um produto. Uma dificuldade é que para um humano é complexo associar diversas variáveis, fazendo com que haja uma generalização exagerada dos perfis, utilizando apenas uma ou duas variáveis em alguns casos, e sejam realizadas algumas classificações enviesadas.

Analisando os dados dos contatos com os *prospects* e também o resultado obtido com as abordagens, foi possível encontrar diversas abordagens que foram feitas com o cliente que havia sido classificado como propenso a comprar o produto,

mas que, após a negociação, não foi efetivada a compra. Isso faz com que a empresa invista recursos, de pessoas e tecnologia, para ser feito o contato com o *prospect*, que poderia ser utilizado com outro *lead* realmente propenso, não havendo um retorno condizente com as expectativas.

Buscando aumentar a efetividade dos contatos foi identificada a oportunidade de utilizar uma técnica de aprendizado de máquina para analisar os dados de cada *lead* e se foi obtido sucesso na abordagem com ele, e criar um modelo e utilizar desse modelo para prever se um *lead* é um potencial comprador ou não.

O projeto aplicado utiliza uma técnica de aprendizado de máquina que possibilita serem analisadas as relações entre todas as variáveis e é evitada a generalização que ocorre quando um humano tenta criar um perfil que define o mais propenso à compra, sendo possível uma melhor segmentação da lista de potenciais compradores e consequentemente aumentar a assertividade dos contatos e o resultado.

Devido a existirem diversas operações de venda, foi optado por realizar um piloto com uma operação para se analisar os resultados e, caso satisfatório, replicar o projeto para outras operações. Foi escolhida uma operação de venda de cartões de crédito.

O projeto iniciou com a etapa de levantamento das variáveis de interesse. Nesta etapa, o objetivo era buscar as variáveis disponíveis para se realizar o projeto e quais seriam as mais relevantes. Ao final, foram obtidas sete variáveis, todas elas qualitativas, que, utilizadas em conjunto, poderiam vir a indicar um potencial comprador.

Em seguida foram compilados os dados dos *leads* que compraram e dos que não efetivaram a compra, obtendo os dados das variáveis de interesse para cada um deles. O objetivo foi obter os insumos para serem usados na etapa de implementação do aprendizado de máquina.

Para ser obtido um melhor resultado com os algoritmos foram realizados ajustes nas variáveis, criando faixas de valores para a idade e segmentando as localidades por região do Brasil, por exemplo.

Com a obtenção dos dados foi dado início a elaboração do script de aprendizado de máquina, o objetivo era criar um script que proporcionasse o aprendizado e pudesse ter uma saída relevante para a análise, no caso, se o *lead* possui uma propensão a compra ou não.

CAPÍTULO 5: MODELAGEM DO PROBLEMA E IMPLEMENTAÇÃO DA SOLUÇÃO

Neste capítulo será apresentado como ocorreu o desenvolvimento do projeto, as dificuldades encontradas e solução final implementada. Ele está dividido em tópicos, com cada parte do projeto implementado para obter a solução final.

O primeiro tópico aborda as variáveis que foram obtidas, o que elas trazem de informações e como foram utilizadas. Em seguida é apresentado como foram obtidas as variáveis. No terceiro tópico são detalhados os procedimentos realizados para preparar os dados para o algoritmo de aprendizado de máquina. Por último é apresentado o script utilizado para a realização do projeto e como ele foi construído no decorrer do desenvolvimento.

5.1 Variáveis de interesse

Inicialmente foi analisada a lista dos potenciais clientes enviada pelo banco para entender melhor quais seriam as variáveis que estariam disponíveis para a análise que seria realizada.

Havia disponível como variáveis o código identificador do *lead*, data que foi recebido o potencial cliente, a data que foi entrado em contato, hora do contato, idade, ano de nascimento, mês de nascimento, gênero e estado de residência, e a variável status, que apresenta o resultado da abordagem, podendo ser o status com o motivo da recusa do cartão de crédito ou de aceite da oferta e compra do produto oferecido.

Nas Figuras 3 e 4 é possível observar uma parte dos dados recebidos.

Não foram selecionadas para a análise as variáveis de identificador do *lead*, data do recebimento e data do contato, por serem variáveis que não agregam insumos sobre o potencial cliente, sendo apenas variáveis para o controle interno da empresa.

	A	B	C	D	E	F	G
1	ID_MAILING	DT IMPORTAÇÃO	DT TABULAÇÃO	HR TABULAÇÃO	IDADE	ANO NASC	MÊS NASC
2	1676992	01/02/2018	28/02/2018	15:34:01	45	1973	6
3	1677001	01/02/2018	28/02/2018	10:25:14	62	1956	4
4	1677004	01/02/2018	09/02/2018	18:21:04	38	1980	3
5	1677009	01/02/2018	26/02/2018	16:51:49	57	1961	8
6	1677010	01/02/2018	21/02/2018	13:43:43	43	1975	7
7	1677016	01/02/2018	28/02/2018	18:13:25	53	1965	5
8	1677017	01/02/2018	26/02/2018	13:37:37	36	1982	9
9	1677021	01/02/2018	27/02/2018	10:37:04	56	1962	8
10	1677025	01/02/2018	26/02/2018	18:18:12	31	1987	10
11	1677027	01/02/2018	28/02/2018	12:23:23	29	1989	9
12	1677061	01/02/2018	28/02/2018	20:19:46	28	1990	9
13	1677068	01/02/2018	26/02/2018	09:48:26	42	1976	4
14	1677069	01/02/2018	28/02/2018	14:41:00	71	1947	10
15	1677077	01/02/2018	27/02/2018	17:40:13	32	1986	6
16	1677095	01/02/2018	28/02/2018	17:07:27	33	1985	5
17	1677097	01/02/2018	07/02/2018	09:22:16	34	1984	7
18	1677099	01/02/2018	01/02/2018	19:33:49	39	1979	5
19	1677114	01/02/2018	02/02/2018	12:18:29	56	1962	12
20	1677121	01/02/2018	27/02/2018	09:50:11	40	1978	12
21	1677127	01/02/2018	28/02/2018	10:48:15	67	1951	3
22	1677129	01/02/2018	28/02/2018	17:32:56	43	1975	8
23	1677130	01/02/2018	02/02/2018	20:55:39	30	1988	10
24	1677173	01/02/2018	28/02/2018	16:48:09	31	1987	4
25	1677180	01/02/2018	07/02/2018	17:46:18	37	1981	7
26	1677187	01/02/2018	07/02/2018	11:04:01	41	1977	6
27	1677210	01/02/2018	27/02/2018	19:41:23	25	1993	8

Figura 3 - Dados recebidos, parte 1

	H	I	J	K
1	GENERO	UF	PERFIL	STATUS
2	M	DF	0	CLIENTE RECUSA-SE A CONFIRMAR DADOS
3	M	DF	0	CLIENTE POSSUI CARTÃO CRÉDITO DE OUTRA INSTITUIÇÃO
4	M	SP	0	VENDA
5	F	SP	0	CLIENTE POSSUI CARTÃO CRÉDITO DE OUTRA INSTITUIÇÃO
6	M	SP	0	VENDA
7	M	SP	0	CLIENTE RECUSA-SE A OUVIR PROPOSTA
8	M	SP	0	CLIENTE RECUSA-SE A OUVIR PROPOSTA
9	M	SP	0	CLIENTE POSSUI CARTÃO CRÉDITO DE OUTRA INSTITUIÇÃO
10	M	SP	0	CLIENTE POSSUI CARTÃO CRÉDITO DE OUTRA INSTITUIÇÃO
11	M	SP	0	CLIENTE NÃO DESEJA MAIS CONTATO DO PAN
12	M	AL	0	CLIENTE POSSUI CARTÃO CRÉDITO DE OUTRA INSTITUIÇÃO
13	F	RJ	0	CLIENTE RECUSA-SE A OUVIR PROPOSTA
14	F	CE	0	CLIENTE POSSUI CARTÃO CRÉDITO DE OUTRA INSTITUIÇÃO
15	M	SP	0	CLIENTE POSSUI CARTÃO CRÉDITO DE OUTRA INSTITUIÇÃO
16	M	SP	0	NÃO POSSUI RENDA/DESEMPREGADO
17	F	DF	0	VENDA
18	M	CE	0	VENDA
19	M	PR	0	VENDA
20	M	PR	0	CLIENTE POSSUI CARTÃO CRÉDITO DE OUTRA INSTITUIÇÃO
21	M	SC	0	CLIENTE NÃO TRABALHA COM CARTÃO
22	M	PE	0	CLIENTE POSSUI CARTÃO CRÉDITO DE OUTRA INSTITUIÇÃO
23	M	SP	0	VENDA
24	M	SP	0	CLIENTE NÃO TRABALHA COM CARTÃO
25	F	SP	0	VENDA
26	M	ES	0	VENDA
27	M	SP	0	CLIENTE RECUSA-SE A OUVIR PROPOSTA

Figura 4 - Dados recebidos, parte 2

Como idade e ano de nascimento representam a mesma característica, para evitar a redundância, foi desconsiderado a ano de nascimento, e foi optado por utilizar a idade. Para uma melhor segmentação dos clientes foi optado por utilizar faixas de idade, variando de 10 em 10 anos, iniciando em 19 anos até 59 anos e acima de 60 anos.

O mês de nascimento foi escolhida como uma das variáveis qualitativas para a análise e essa informação foi utilizada para criar a variável de se o mês do contato é o mês de aniversário do *lead*, como uma possibilidade de variável que possa trazer mais insumos para a predição.

Foi utilizado o estado de residência entre as variáveis e também foi gerada a variável de região do Brasil, buscando entender se há uma relação entre as regiões e a propensão à compra.

As variáveis de gênero fornecida pelo banco foi adicionada nas variáveis de interesse, não sendo modificada de nenhuma forma.

A variável de saída do projeto é se o *lead* comprou ou não o cartão de crédito, essa informação está na variável status, como os motivos de recusa não importam para esse projeto, foi optado por criar a variável “comprou” que indica se no final da abordagem houve uma compra ou não, e essa variável foi adicionada para ser relacionada com as outras e ser possível predizer a propensão à compra dos novos *leads* que são encaminhados pelo banco.

Inicialmente foram realizados alguns testes com essas variáveis comentadas acima e o resultado não foi satisfatório, nos primeiros testes o algoritmo acertava cerca de 50% dos casos, o que era próximo ao resultado de um algoritmo que classificava os exemplos aleatoriamente.

Buscando entender mais sobre os dados foi realizada uma conversa com o analista responsável pela estratégia desta operação. Durante a conversa foi revelado que existia uma variável de perfil que era encaminhada pelo banco, classificando o *prospect* em três tipos de perfil, e que esta variável trazia diversos insumos para as decisões estratégicas. Desta forma, adicionou-se essa variável entre as que seriam analisadas no decorrer do projeto.

Havia também a variável de se o *lead* é cliente do banco, foi elencada como uma variável de interesse inicialmente, porém, posteriormente, ao ser obtido os

dados, percebeu-se que essa variável estava classificando todos os *leads* como não sendo clientes do banco, o que, devido a polaridade, não traria nenhum insumo para análise e optou-se por removê-la da análise.

As variáveis utilizadas podem ser observadas na Tabela 1.

Tabela 1 - Variáveis de interesse

Variável	Descrição
Faixa Etária	Range de idade, variando de 10 em 10 anos, iniciando em 19 anos até 59 anos e acima de 60 anos
Mês de Nascimento	Mês do nascimento do *lead*
É Mês do Aniversário	Se o mês do contato é o mês do aniversário do *lead*, variável binária
Gênero	Qual o gênero do *lead*
Estado	Estado que o *lead* reside
Região	Região do Brasil que o *lead* reside
Perfil Banco	Perfil encaminhado pelo banco, podendo ser 0, 2 ou 3
Comprou	Se o *lead* comprou ou não o produto após a abordagem, variável binária

5.2 Compilação dos dados

Após a seleção das variáveis de interesse, foi dado início à coleta e compilação dos dados dos potenciais clientes. A compilação foi feita através de planilhas de excel, utilizando fórmulas para realizar relações entre os dados e criar as variáveis citadas na seção anterior.

Uma dificuldade foi não possuir acesso direto ao banco de dados da empresa, somente a relatórios, fornecidos pela área de report center, que é a área responsável por relatórios, embora isso facilite o acesso aos dados, por não ser necessário acessar o banco de dados diretamente, não sendo necessário realizar os scripts SQL. Porém, não era possível realizar uma consulta para obter outros dados e entender melhor a dinâmica dos *leads*.

Ao final obteve-se cerca de 100 mil *leads* para realizar o projeto, já com o resultado das abordagens nos últimos 3 meses. Os dados compilados, com as variáveis finais de alguns *leads* a serem utilizados de exemplo, podem ser observados nas Figuras 5 e 6.

	A	B	C	D	E
1	ID_MAILING	DT IMPORTACAO	DT TABULACAO	HR TABULACAO	STATUS
2	1676992	01/02/2018	28/02/2018	15:34:01	CLIENTE RECUSA-SE A CONFIRMAR DADOS
3	1677001	01/02/2018	28/02/2018	10:25:14	CLIENTE POSSUI CARTÃO CRÉDITO DE OUTRA INSTITUIÇÃO
4	1677004	01/02/2018	09/02/2018	18:21:04	VENDA
5	1677009	01/02/2018	26/02/2018	16:51:49	CLIENTE POSSUI CARTÃO CRÉDITO DE OUTRA INSTITUIÇÃO
6	1677010	01/02/2018	21/02/2018	13:43:43	VENDA
7	1677016	01/02/2018	28/02/2018	18:13:25	CLIENTE RECUSA-SE A OUVIR PROPOSTA
8	1677017	01/02/2018	26/02/2018	13:37:37	CLIENTE RECUSA-SE A OUVIR PROPOSTA
9	1677021	01/02/2018	27/02/2018	10:37:04	CLIENTE POSSUI CARTÃO CRÉDITO DE OUTRA INSTITUIÇÃO
10	1677025	01/02/2018	26/02/2018	18:18:12	CLIENTE POSSUI CARTÃO CRÉDITO DE OUTRA INSTITUIÇÃO
11	1677027	01/02/2018	28/02/2018	12:23:23	CLIENTE NÃO DESEJA MAIS CONTATO DO PAN
12	1677061	01/02/2018	28/02/2018	20:19:46	CLIENTE POSSUI CARTÃO CRÉDITO DE OUTRA INSTITUIÇÃO
13	1677068	01/02/2018	26/02/2018	09:48:26	CLIENTE RECUSA-SE A OUVIR PROPOSTA
14	1677069	01/02/2018	28/02/2018	14:41:00	CLIENTE POSSUI CARTÃO CRÉDITO DE OUTRA INSTITUIÇÃO
15	1677077	01/02/2018	27/02/2018	17:40:13	CLIENTE POSSUI CARTÃO CRÉDITO DE OUTRA INSTITUIÇÃO
16	1677095	01/02/2018	28/02/2018	17:07:27	NÃO POSSUI RENDA/DESEMPREGADO
17	1677097	01/02/2018	07/02/2018	09:22:16	VENDA
18	1677099	01/02/2018	01/02/2018	19:33:49	VENDA
19	1677114	01/02/2018	02/02/2018	12:18:29	VENDA
20	1677121	01/02/2018	27/02/2018	09:50:11	CLIENTE POSSUI CARTÃO CRÉDITO DE OUTRA INSTITUIÇÃO
21	1677127	01/02/2018	28/02/2018	10:48:15	CLIENTE NÃO TRABALHA COM CARTÃO
22	1677129	01/02/2018	28/02/2018	17:32:56	CLIENTE POSSUI CARTÃO CRÉDITO DE OUTRA INSTITUIÇÃO
23	1677130	01/02/2018	02/02/2018	20:55:39	VENDA
24	1677173	01/02/2018	28/02/2018	16:48:09	CLIENTE NÃO TRABALHA COM CARTÃO
25	1677180	01/02/2018	07/02/2018	17:46:18	VENDA
26	1677187	01/02/2018	07/02/2018	11:04:01	VENDA
27	1677210	01/02/2018	27/02/2018	19:41:23	CLIENTE RECUSA-SE A OUVIR PROPOSTA

Figura 5 - Dados compilados, parte 1

	F	G	H	I	J	K	L	M
1	É ANIVERSARIO	FAIXA ETÁRIA	MÊS NASC	GÊNERO	UF	REGIÃO	PERFIL	COMPROU
2	0	39 a 49	6	M	DF	CENTRO OESTE	0	0
3	0	59 a 99	4	M	DF	CENTRO OESTE	0	0
4	0	29 a 39	3	M	SP	SUDESTE	0	1
5	0	49 a 59	8	F	SP	SUDESTE	0	0
6	0	39 a 49	7	M	SP	SUDESTE	0	1
7	0	49 a 59	5	M	SP	SUDESTE	0	0
8	0	29 a 39	9	M	SP	SUDESTE	0	0
9	0	49 a 59	8	M	SP	SUDESTE	0	0
10	0	29 a 39	10	M	SP	SUDESTE	0	0
11	0	29 a 39	9	M	SP	SUDESTE	0	0
12	0	19 a 29	9	M	AL	NORDESTE	0	0
13	0	39 a 49	4	F	RJ	SUDESTE	0	0
14	0	59 a 99	10	F	CE	NORDESTE	0	0
15	0	29 a 39	6	M	SP	SUDESTE	0	0
16	0	29 a 39	5	M	SP	SUDESTE	0	0
17	0	29 a 39	7	F	DF	CENTRO OESTE	0	1
18	0	39 a 49	5	M	CE	NORDESTE	0	1
19	0	49 a 59	12	M	PR	SUL	0	1
20	0	39 a 49	12	M	PR	SUL	0	0
21	0	59 a 99	3	M	SC	SUL	0	0
22	0	39 a 49	8	M	PE	NORDESTE	0	0
23	0	29 a 39	10	M	SP	SUDESTE	0	1
24	0	29 a 39	4	M	SP	SUDESTE	0	0
25	0	29 a 39	7	F	SP	SUDESTE	0	1
26	0	39 a 49	6	M	ES	SUDESTE	0	1
27	0	19 a 29	8	M	SP	SUDESTE	0	0

Figura 6 - Dados compilados, parte 2

A última coluna indica se o potencial cliente comprou ou não o cartão de crédito, sendo 1 para uma compra efetivada e 0 para uma recusa. Através dos dados que foram compilados foi possível entender mais sobre o comportamento dos *leads*.

5.3 Pré-processamento dos dados

Dados provenientes de fontes reais geralmente necessitam de um pré-processamento a fim de serem utilizados pelas técnicas de aprendizado de máquina previamente apresentadas. As etapas mais importantes nessa etapa são a limpeza e seleção de variáveis e a amostragem [10].

Valores incorretos podem causar distorções nos modelos obtidos, pois, em geral, os algoritmos de aprendizado de máquina assumem que todos os valores utilizados são completamente corretos. A verificação buscando esse tipo de erro deve ser realizada caso os resultados estejam muito diferentes do esperado, ou quando se supõe a priori que os dados possam ser incorretos.

Para encontrar valores potencialmente incorretos em variáveis categóricas, primeiramente se deve examinar as frequências. Utilizar um histograma, por exemplo, permite buscar valores não usuais. No caso de serem poucos os exemplos contendo essas discrepâncias, pode-se examinar um por vez, para verificar se os valores são factíveis. Para casos em que a quantidade de dados analisada é muito grande, deve-se buscar uma ferramenta que realize essa verificação e correção (ou simplesmente uma sinalização para o desenvolvedor) de forma automática.

Essa análise pode requerer familiaridade e experiência com o tipo de dados estudado, e ainda assim, para alguns valores é difícil determinar qual o valor correto, ou esperado, para aquele campo. Quando dados supostamente incorretos ocorrem em pouca frequência e não podem ser interpretados ou substituídos, pode-se considerá-los como dados faltantes.

Já para variáveis contínuas, valores incorretos podem ser detectados como picos em distribuições, representando assim uma discrepância. Caso esses dados estejam prejudicando os resultados, podem ser tratados com a aplicação de filtros. Em casos onde os dados contendo valores incorretos são tão raros que não podem

ser detectados facilmente, pode-se mantê-los, pois provavelmente eles não apresentarão relevância no processo de modelagem.

A falta de dados é o maior dos problemas relacionados ao condicionamento dos dados. Geralmente, dados faltantes são representados como valores nulos ou como células vazias. Uma análise, porém, sempre deve ser realizada nos dados a serem utilizados pois, em alguns casos, os dados faltantes são representados com valores não nulos, podendo levar o sistema a interpretá-los erroneamente como registros válidos [11].

As causas para a falta de dados são variadas. Em alguns casos pode ser um problema na etapa de aquisição dos dados. Outras vezes podem ser simplesmente dados cujos valores são desconhecidos. Em certas ocasiões também pode ocorrer uma perda de dados armazenados em uma base. Em outros casos, porém, os dados faltantes são omitidos propositalmente pelas fontes durante o processo de coleta. Em uma pesquisa, por exemplo, alguns dos entrevistados podem se recusar a responder algumas das questões propostas.

Cabe um estudo nos dados utilizados em cada aplicação para obter a codificação correta dos valores nulos para cada um dos campos que compõe a base de dados, e entender como esses valores podem impactar os modelos gerados pelas técnicas de aprendizado de máquina.

No caso dos dados disponíveis para esta análise observou-se que havia alguns casos com a variável idade menor de 18 anos, o que mostra um erro no lançamento da idade, uma violação da regra de negócio, pois não é permitido vender para uma pessoa que não seja maior de idade, foi optado por considerar esses dados como dados faltantes, para não ser prejudicada a análise. Além dessa variável também havia a variável gênero com algumas linhas faltantes dessa informação.

Como a representatividade das amostras com dados faltantes era muito baixa, cerca de 2% do total de amostras, e havia poucas variáveis disponíveis para ser utilizada uma técnica de preenchimento artificial, que é o caso de utilizar as variáveis já preenchidas, compará-las com as outras amostras para poder preencher o valor nulo. Foi optado por remover as amostras com dados faltantes para que a análise não fosse prejudicada, assim aumentou-se a confiabilidade dos dados.

É utilizando esses dados já pré-processados e efetuado o processo de limpeza que será rodado o script de aprendizado de máquina e predição, apresentado no próximo tópico.

5.4 Script de aprendizado de máquina

Com os dados compilados e pré-processados, foi elaborado um script em Python para realizar o processamento dos mesmos e fornecer se um *lead* é propenso ou não a comprar o produto.

O script foi baseado em um script utilizado em um curso de machine learning da plataforma Udemy. O mesmo utilizou algumas bibliotecas voltadas para o processamento dos dados e aplicação de técnicas de aprendizado de máquina, como numpy e Pandas. Tais bibliotecas possuem funções que facilitam a aplicação das técnicas utilizadas, uma vez que possuem os algoritmos já programados e com as funções específicas para o uso.

Devido ao desbalanceamento das classes presentes nos dados, com a grande maioria das amostras não tendo realizado a compra foi observado que seria necessário utilizar uma técnica de amostragem. Foi optado por se testar duas técnicas, uma de subamostragem, o *Random Under Sampler*, que remove aleatoriamente alguns casos da classe majoritária, para reduzir o desbalanceamento [12]. E uma técnica de superamostragem por geração sintética. A técnica escolhida para esse tipo de amostragem foi a SMOTE, que foi configurada para replicar as minorias, gerando novos exemplos sintéticos dos casos menos presentes [4].

É importante notar que as técnicas de amostragem devem ser realizadas após a separação da base de aprendizado e de teste, para garantir que não haja elementos da base de teste na base de aprendizado.

A última etapa do desenvolvimento do modelo é a escolha do algoritmo de predição. Com a definição das características a serem analisadas e o tipo de amostragem, foram testados 9 modelos com algoritmos diferentes sendo eles avaliados pela matriz de confusão e seus indicadores.

Foram escolhidos os principais modelos de aprendizado de máquina de classificação, sendo eles, a regressão logística, que calcula a probabilidade de uma amostra pertencer a cada uma das classes. Árvore de decisão, que separa em

nodos de decisão, cada um com uma variável e uma condição para seguir para outro nodo ou decisão final. *Random Forest*, que cria diversas árvores de decisão e utiliza todas as decisões das árvores para encontrar a decisão final. *Extra Trees*, é uma variação do *random forest*, que a cada nodo das árvores escolhe uma variável aleatória para ser utilizada, ao invés de utilizar alguns critérios, trazendo mais aleatoriedade para o modelo. *K Nearest Neighbors*, que determina a classe de classificação de uma amostra baseada nas amostras vizinhas analisadas no treinamento. *AdaBoost*, que combina diversos classificadores fracos e a cada iteração ajusta o peso dado a cada um deles na decisão final. *Bagging*, que combina diversos classificadores e utiliza a decisão mais frequente entre eles como a decisão final. *Gradient Boosting*, que combina diversos classificadores fracos e a cada iteração busca por padrões nos resíduos, até que eles sejam considerados aleatórios. Rede Neural que possui neurônios, que consideram pesos para cada variável e produzem uma saída, e combinados esses neurônios formam a rede neural.

Cada modelo é criado e é feito a predição em cima de uma base de dados histórica na qual não fora usada para o treinamento do modelo.

O script final utilizado com todos os algoritmos está no Apêndice A desta monografia.

CAPÍTULO 6: RESULTADOS OBTIDOS

Para a escolha do algoritmo de aprendizado de máquina foram testados os 9 algoritmos de predição escolhidos com os dois algoritmos de amostragem, os resultados dos testes são apresentados, com suas respectivas matrizes de confusão, nas figuras 7 à 24, que é em um gráfico dos acertos e erros do algoritmo, comparando o valor real das amostras com o predito pelo algoritmo.

Para a avaliação dos resultados dos algoritmos serão utilizados os indicadores de taxa de acerto, taxa de acerto na segmentação classificada como positiva e a quantidade de acertos na segmentação positiva. A escolha se dá pois a motivação do projeto é uma melhor segmentação dos perfis propensos à compra, e através desses indicadores é possível identificar o algoritmo que melhor alcança esse objetivo.

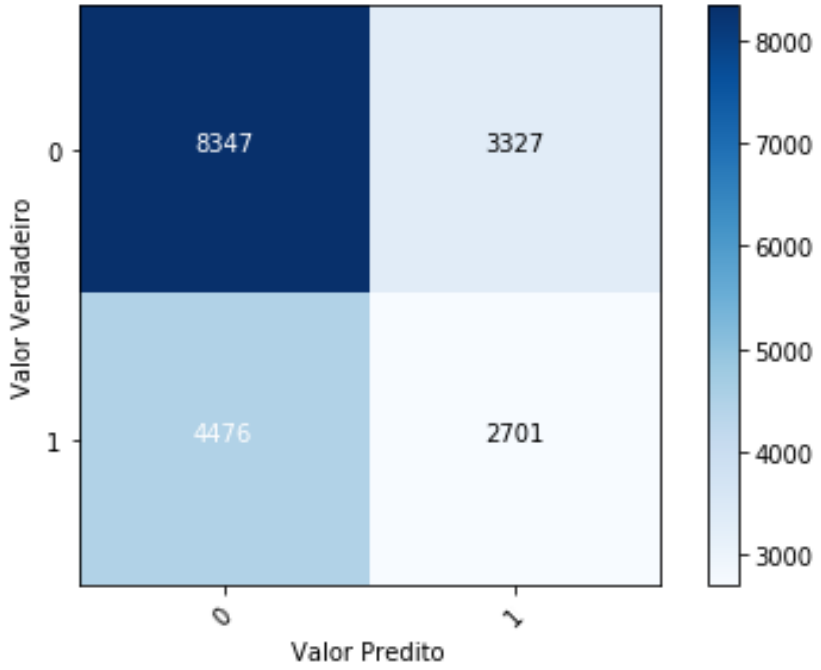

Figura 7 - Árvore de decisão com SMOTE

- Taxa de acerto: 58,6%
- Taxa de acerto na segmentação positiva: 44,8%
- Quantidade de acertos na segmentação positiva: 2701

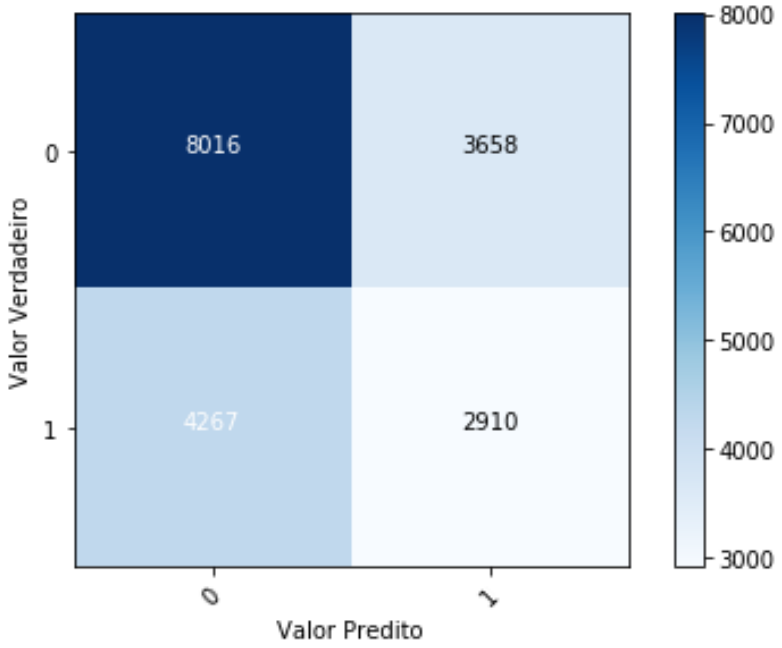

Figura 8 - Árvore de decisão com Random Under Sampler

- Taxa de acerto: 58,0%
- Taxa de acerto na segmentação positiva: 44,3%
- Quantidade de acertos na segmentação positiva: 2910

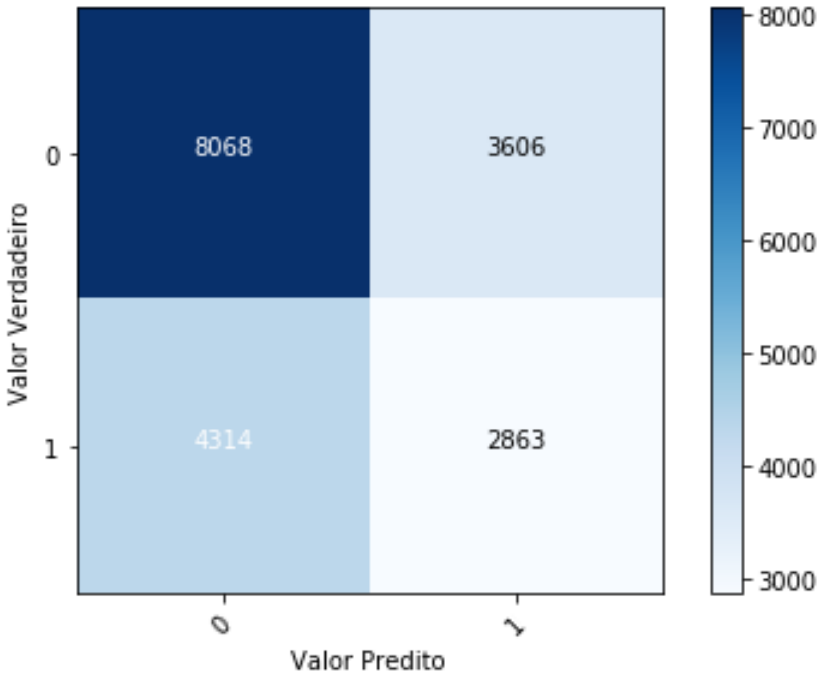

Figura 9 - Random Forest com SMOTE

- Taxa de acerto: 58,0%
- Taxa de acerto na segmentação positiva: 44,3%
- Quantidade de acertos na segmentação positiva: 2863

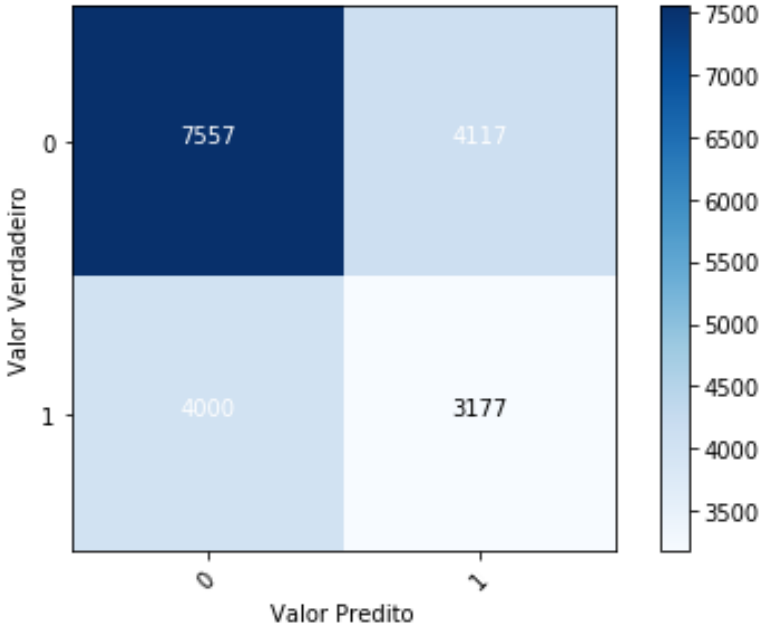

Figura 10 - Random Forest com Random Under Sampler

- Taxa de acerto: 56,9%
- Taxa de acerto na segmentação positiva: 43,6%
- Quantidade de acertos na segmentação positiva: 3177

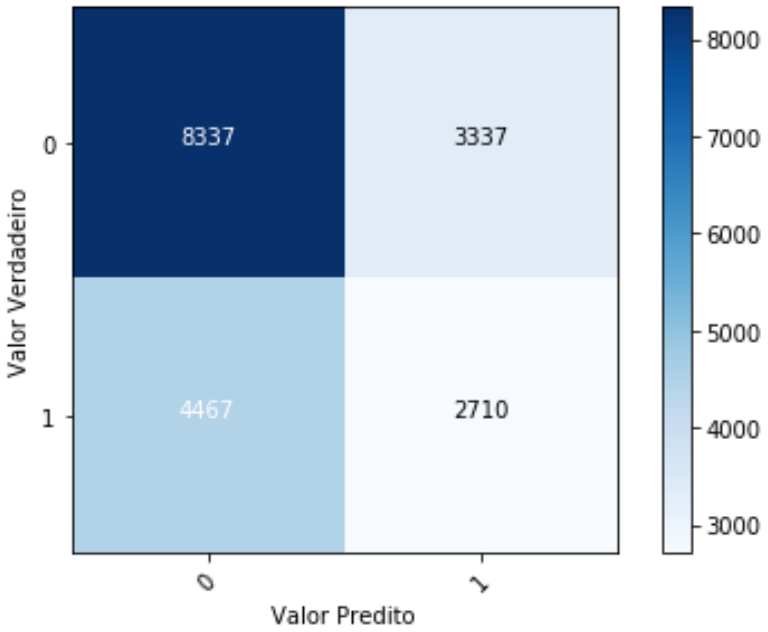

Figura 11 - Extra Trees com SMOTE

- Taxa de acerto: 58,6%
- Taxa de acerto na segmentação positiva: 44,8%
- Quantidade de acertos na segmentação positiva: 2710

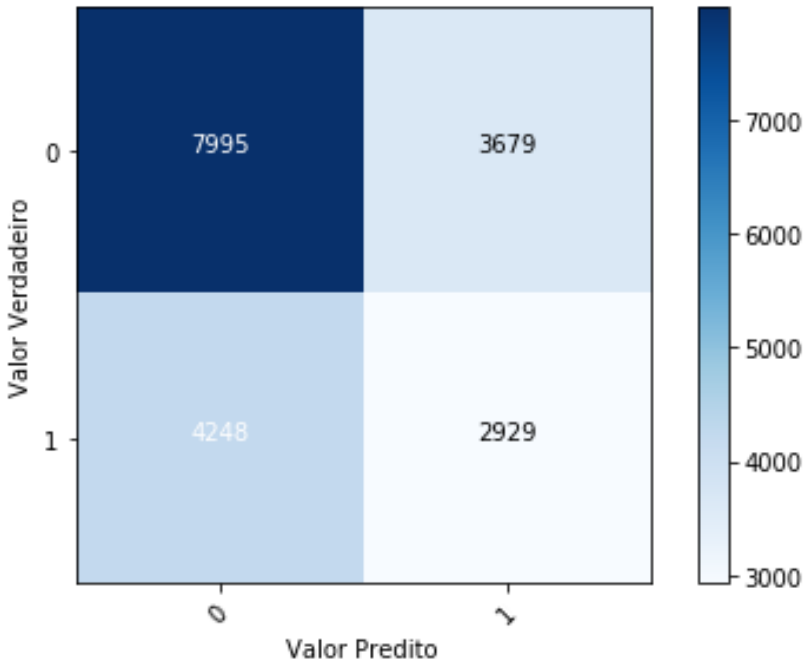

Figura 12 - Extra Trees com Random Under Sampler

- Taxa de acerto: 57,9%
- Taxa de acerto na segmentação positiva: 44,3%
- Quantidade de acertos na segmentação positiva: 2929

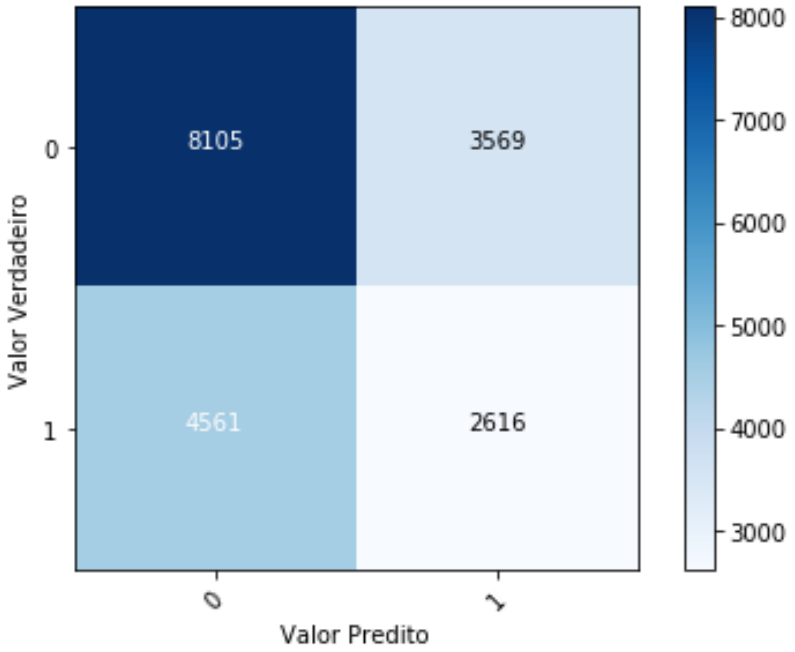

Figura 13 - K-Nearest Neighbors com SMOTE

- Taxa de acerto: 56,9%
- Taxa de acerto na segmentação positiva: 42,3%
- Quantidade de acertos na segmentação positiva: 2616

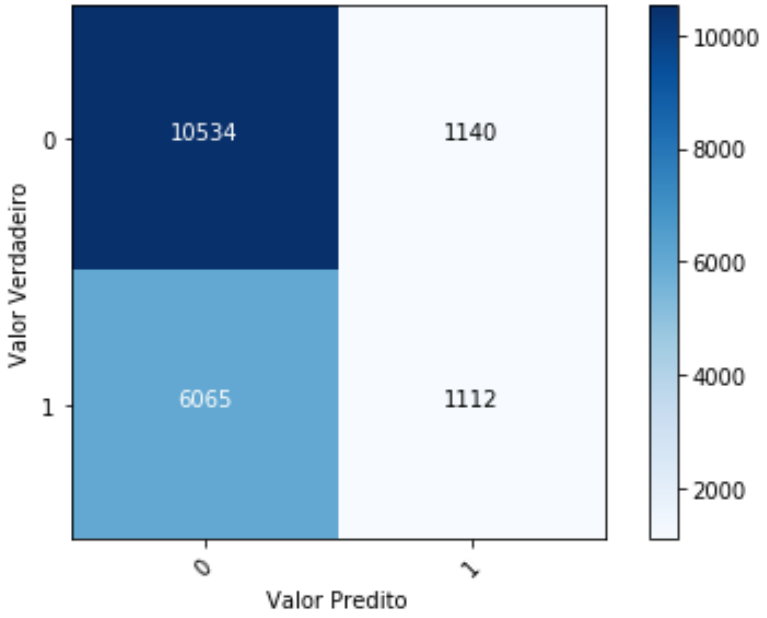

Figura 14 - K-Nearest Neighbors com Random Under Sampler

- Taxa de acerto: 61,8%
- Taxa de acerto na segmentação positiva: 49,4%
- Quantidade de acertos na segmentação positiva: 1112

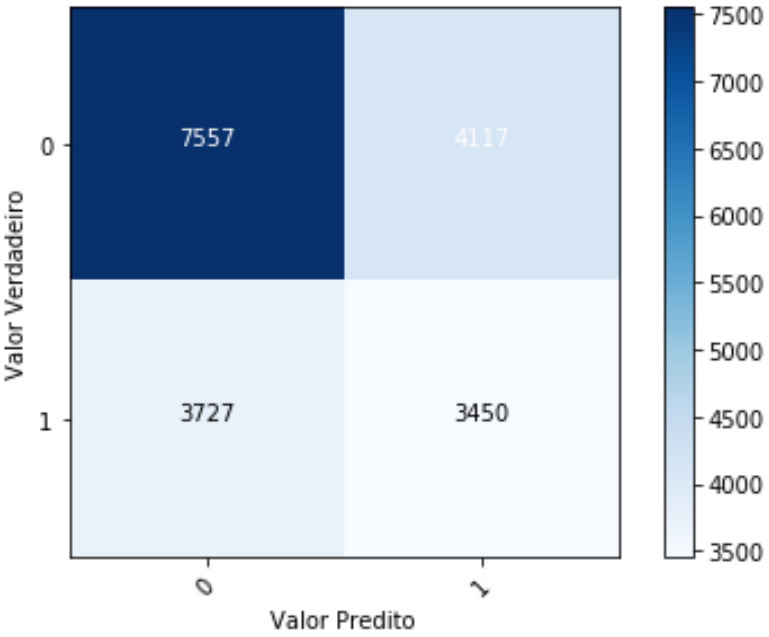

Figura 15 - AdaBoost com SMOTE

- Taxa de acerto: 58,4%
- Taxa de acerto na segmentação positiva: 45,6%
- Quantidade de acertos na segmentação positiva: 3450

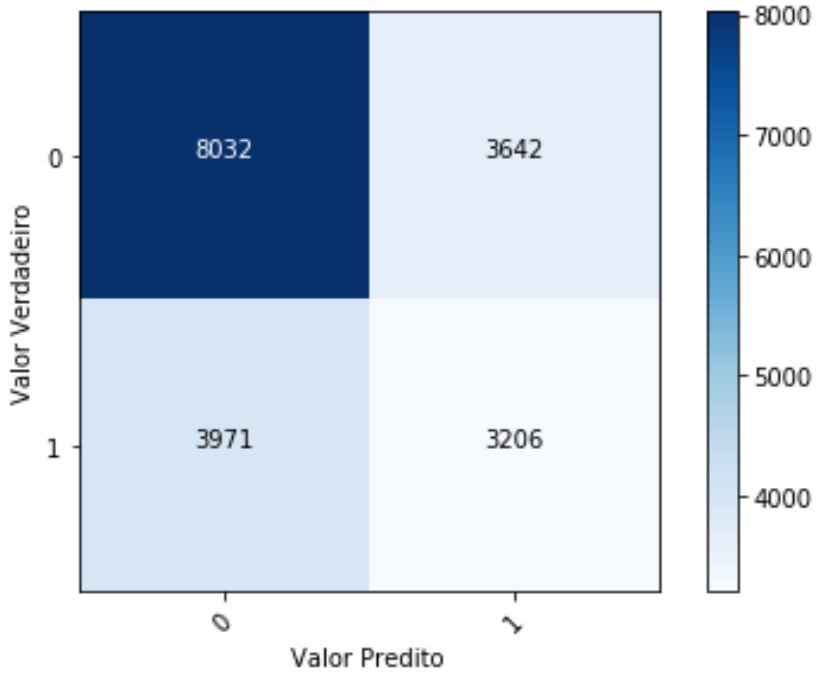

Figura 16 - AdaBoost com Random Under Sampler

- Taxa de acerto: 59,6%
- Taxa de acerto na segmentação positiva: 46,8%
- Quantidade de acertos na segmentação positiva: 3206

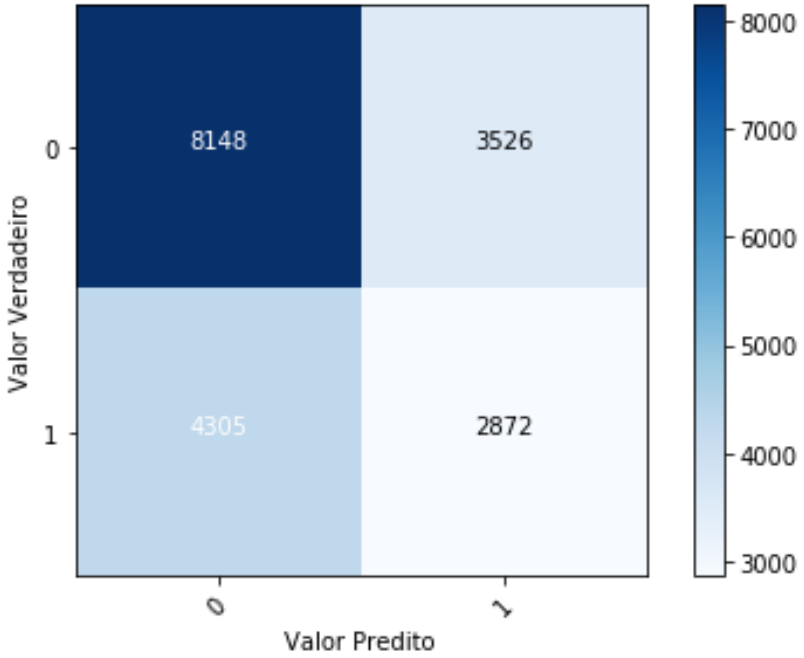

Figura 17 - Bagging com SMOTE

- Taxa de acerto: 58,5%
- Taxa de acerto na segmentação positiva: 44,9%
- Quantidade de acertos na segmentação positiva: 2872

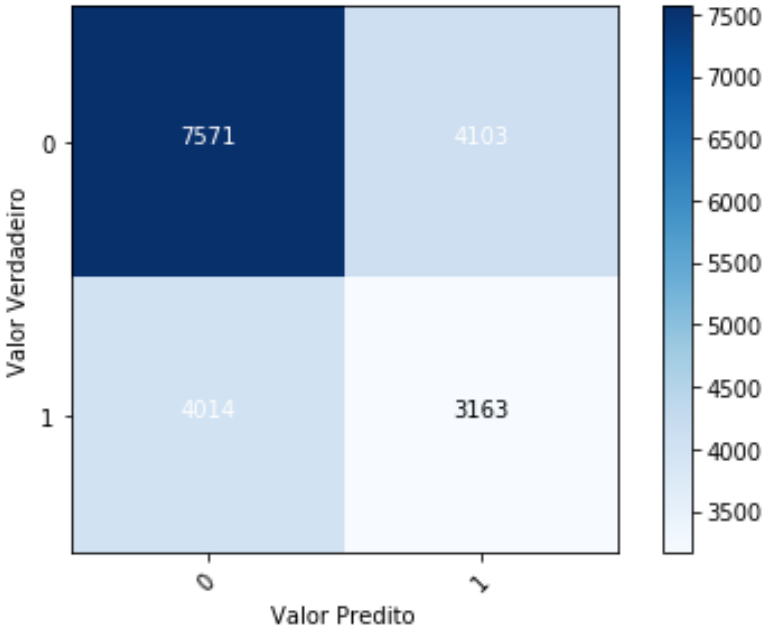

Figura 18 - Bagging com Random Under Sampler

- Taxa de acerto: 56,9%
- Taxa de acerto na segmentação positiva: 43,5%
- Quantidade de acertos na segmentação positiva: 3163

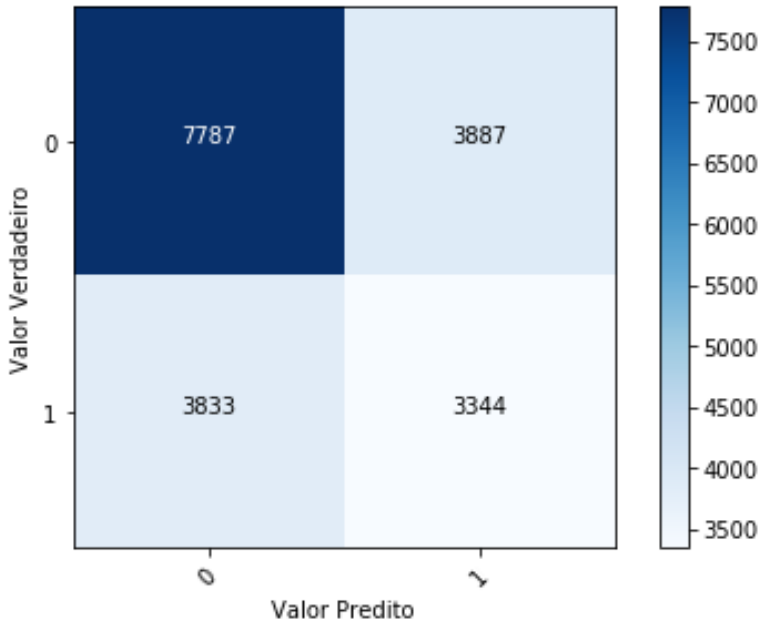

Figura 19 - Regressão Logística com SMOTE

- Taxa de acerto: 59,0%
- Taxa de acerto na segmentação positiva: 46,2%
- Quantidade de acertos na segmentação positiva: 3344

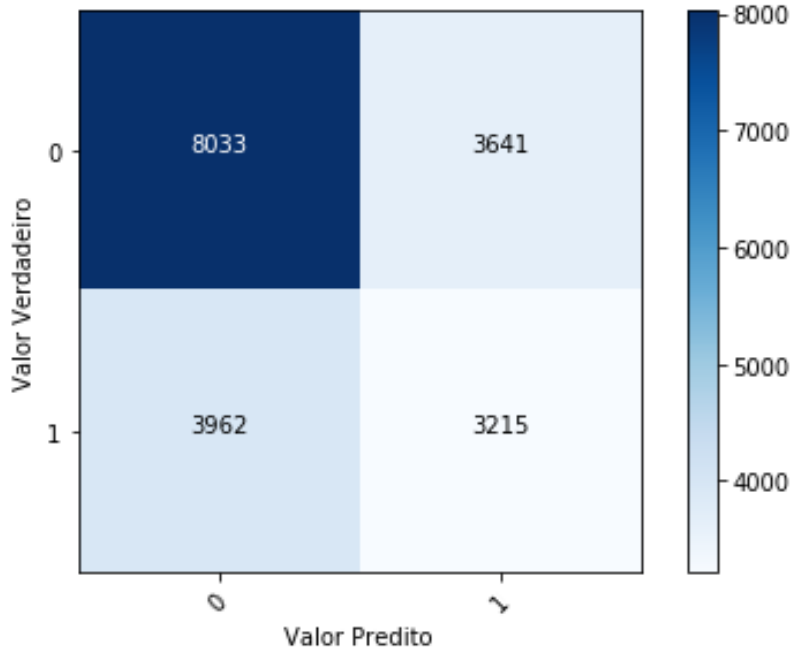

Figura 20 - Regressão Logística com Random Under Sampler

- Taxa de acerto: 59,7%
- Taxa de acerto na segmentação positiva: 46,9%
- Quantidade de acertos na segmentação positiva: 3215

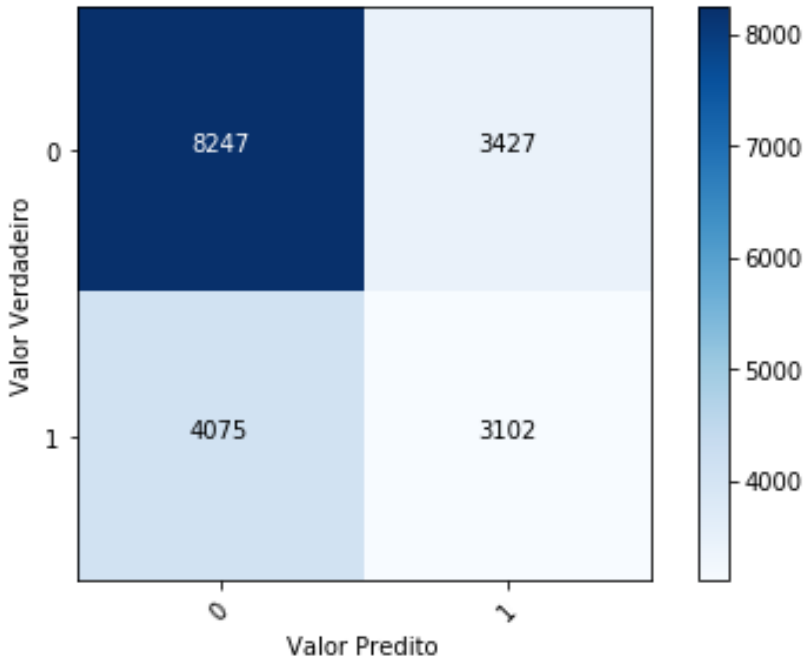

Figura 21 - Gradient Boosting com SMOTE

- Taxa de acerto: 60,2%
- Taxa de acerto na segmentação positiva: 47,5%
- Quantidade de acertos na segmentação positiva: 3102

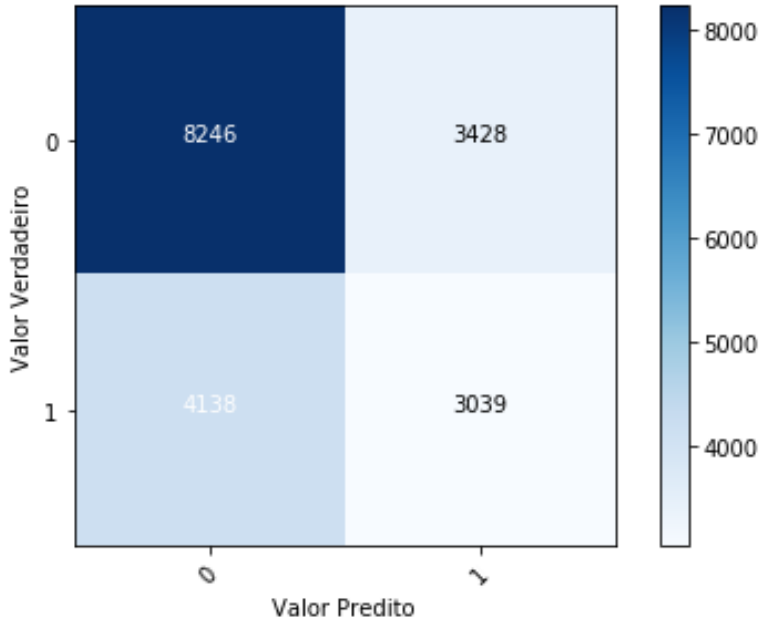

Figura 22 - Gradient Boosting com Random Under Sampler

- Taxa de acerto: 59,6%
- Taxa de acerto na segmentação positiva: 46,9%
- Quantidade de acertos na segmentação positiva: 3029

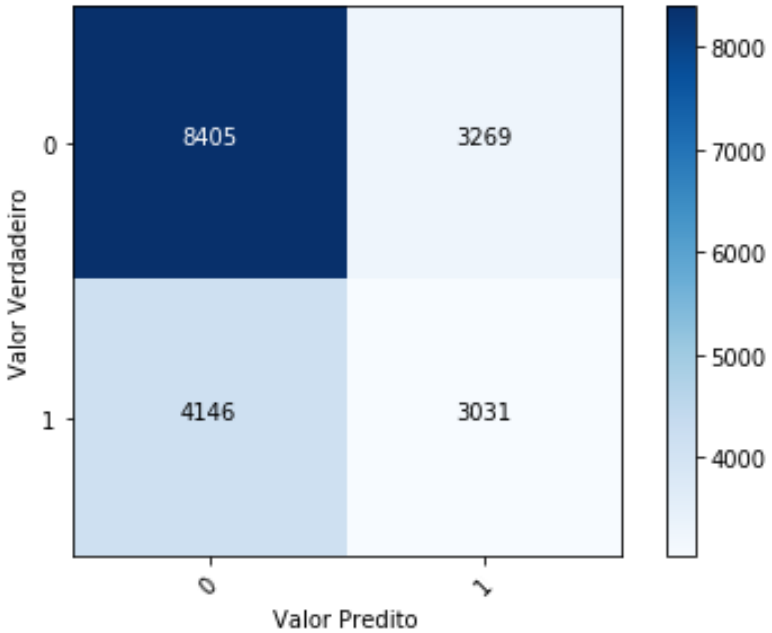

Figura 23 - Rede Neural com SMOTE

- Taxa de acerto: 60,7%
- Taxa de acerto na segmentação positiva: 48,1%
- Quantidade de acertos na segmentação positiva: 3031

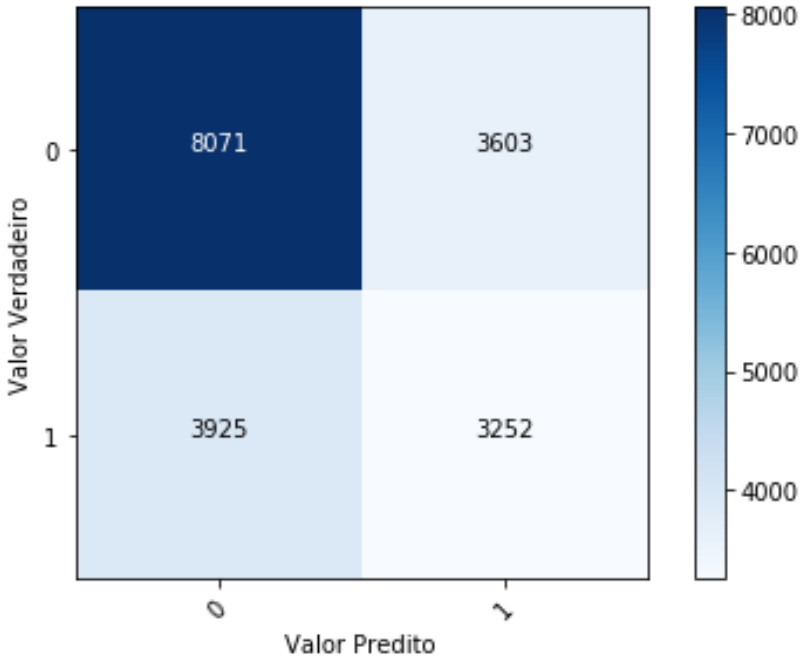

Figura 24 - Rede Neural com Random Under Sampler

- Taxa de acerto: 60,1%
- Taxa de acerto na segmentação positiva: 47,4%
- Quantidade de acertos na segmentação positiva: 3252

Tabela 2 - Comparação dos resultados dos algoritmos

Algoritmo	Tx Acerto	Tx Seg. Positiva	Qtd Seg. Positiva
Árvore de decisão com SMOTE	58,6%	44,8%	2701
Árvore de decisão com Random Under Sampler	58,0%	44,3%	2910
Random Forest com SMOTE	58,0%	44,3%	2863
Random Forest com Random Under Sampler	56,9%	43,6%	3177
Extra Trees com SMOTE	58,6%	44,8%	2710
Extra Trees com Random Under Sampler	57,9%	44,3%	2929
K-Nearest Neighbors com SMOTE	56,9%	42,3%	2616

K-Nearest Neighbors com Random Under Sampler	61,8%	49,4%	1112
AdaBoost com SMOTE	58,4%	45,6%	3450
AdaBoost com Random Under Sampler	59,6%	46,8%	3206
Bagging com SMOTE	58,5%	44,9%	2872
Bagging com Random Under Sampler	56,9%	43,5%	3163
Regressão Logística com SMOTE	59,0%	46,2%	3344
Regressão Logística com Random Under Sampler	59,7%	46,9%	3215
Gradient Boosting com SMOTE	60,2%	47,5%	3102
Gradient Boosting com Random Under Sampler	59,6%	46,9%	3029
Rede Neural com SMOTE	60,7%	48,1%	3031
Rede Neural com Random Under Sampler	60,1%	47,4%	3252

A maior parte dos algoritmos apresentou um desempenho similar, sendo que o K-Nearest Neighbors com Random Under Sampler se destacou e apresentou ótimas taxas de acerto, porém segmentou uma parte pequena da lista como positiva, o que faz com que haja poucos nomes para serem entrados em contato, por esse motivo, foi escolhido o algoritmo de Rede Neural com amostragem por Random Under Sampler, que também apresentou taxas acima da média e segmentou uma boa parte da lista como positiva.

Apesar de o algoritmo não possuir uma taxa de acerto muito elevada ele ainda consegue classificar melhor do que o caso de segmentar toda a lista como positiva, que acertaria apenas 38% dos casos. E também consegue ser um pouco mais eficiente do que a segmentação pela regra criada pela área de *Command Post*,

que tem uma taxa de acerto na segmentação positiva de 46,2% e uma quantidade de acertos na segmentação positiva de 2245, a vantagem do algoritmo é justamente o tamanho da lista que ele consegue segmentar, pois com mais nomes é possível ser tentado contato com mais pessoas propensas.

Sendo assim, com a estrutura de dados e o script definidos é possível proporcionar uma melhoria no processo de segmentação dos *leads* mais propensos à comprar o cartão de crédito e pode trazer um resultado melhor para a operação.

Além disso, é facilitado o processo de segmentação das listas, por ele ser realizado pelo algoritmo, auxiliando a área de *Command Post* e é assistida também a área operacional a alcançar as metas pretendidas, principalmente de número de vendas.

A partir da apresentação dos resultados às pessoas responsáveis pela operação, foi identificado que seria interessante utilizar as regras já existentes pela área de *Command Post* e em paralelo utilizar essa segmentação do algoritmo fazendo um complemento de uma lista com a outra, analisando a intersecção das listas e para cada perfil que cada lista pode ser mais assertiva, e assim é possível um melhor aproveitamento de todos os *leads* disponibilizados pelo banco.

Como a implementação do projeto foi recente, ainda está sendo rodado em apenas uma operação piloto e ainda não substitui o processo que já acontecia na área. Porém, de início, é possível perceber o impacto que ele pode ter, há uma lista maior na segmentação o que possibilita encaminhar mais *leads* mais propensos à compra aos melhores vendedores, aumentando assim a quantidade de vendas realizadas.

CAPÍTULO 7: CONSIDERAÇÕES FINAIS E SUGESTÕES

O trabalho foi implementado com sucesso, sendo estabelecido um processo adaptado para a predição dos *leads* mais propensos à compra, utilizando as ferramentas desenvolvidas no decorrer do projeto.

Com a implementação foi possível que a área de *Command Post* passasse a ser mais orientada por dados, permitindo que a tomada de decisão e elaboração de estratégias fosse mais baseada em resultados e indicadores.

Com os resultados obtidos com o projeto será possível diminuir o trabalho manual da área e pode ser focado os esforços em outras atividades que podem trazer um maior resultado.

Deixaria como sugestão, para a implementação de projetos futuros, a busca por operações que possuem mais dados do *lead*, pois com mais variáveis o algoritmo pode associar mais informações e possivelmente ser mais assertivo na predição.

O presente projeto trouxe, a cada etapa, um conhecimento muito grandioso para o aluno, que pode colocar em prática os conhecimentos que obteve nas disciplinas realizadas na Universidade Federal de Santa Catarina.

A partir do desenvolvimento deste trabalho, pode-se analisar que os resultados e as aplicações de Machine Learning trazem resultados satisfatórios. Com o impacto positivo do modelo, abre-se a possibilidade da criação de um modelo para a predição do melhor dia e horário para entrar em contato com um *lead*, assim é aumentada a taxa de contato, diminui-se o esforço para discar para diversas pessoas ao mesmo tempo e consequentemente a operação passa a ser mais eficiente.

Este projeto, até o momento, foi o mais complexo e desafiador que o aluno desenvolveu durante toda a sua vida acadêmica. Isto se deve, ao grau de especificidade e inovação do trabalho. Não existem informações de empresas deste ramo que possuem processos semelhantes, visto o grau de sigilo deste tipo de informação, uma vez que trata de um diferencial estratégico para a empresa.

Outra dificuldade enfrentada foi quanto ao pré-processamento e a limpeza dos dados, tendo em vista que não há muitas variáveis disponíveis e não há muitas informações sobre como lidar em situações assim. O projeto foi desenvolvido em

etapas incrementais, que juntas dão forma ao projeto final implementado, sendo uma dependente da outra.

Por fim, o projeto atingiu parcialmente as expectativas iniciais, ao mesmo tempo que ele foi mais eficiente que o processo anteriormente existente ainda não se obteve um ganho de efetividade tão grande quanto o esperado. Porém o acadêmico conseguiu adquirir conhecimentos técnicos, sem contar a evolução das suas habilidades, a experiência de mercado e os contatos profissionais que foram realizados no decorrer do projeto.

REFERÊNCIAS

[1] M. A. Ponti e G. B. P. d. Costa, "Como funciona o Deep Learning” [Online]. Disponível em: http://conteudo.icmc.usp.br/pessoas/moacir/papers/Ponti_Costa_Como-funciona-o-Deep-Learning_2017.pdf. [Acesso em 16 06 2018].

[2] N. V. Chawla, N. Japkowicz, A. Kotcz. “Editorial: Special Issue on Learning from Imbalanced Data Sets”. SIGKDD Explorations, v. 6, n. 1, p.1-6, 2004.

[3] N. V. Chawla, “Data mining for imbalanced datasets: an overview”. Em: O. Maimon, L. Rokach, Data Mining and Knowledge Discovery Handbook, p. 853-867, 2005

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall e W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique” Journal of Artificial Intelligence Research, v. 16, p. 321-357, 2002.

[5] L. N. d. Castro e F. J. V. Zuben, "Redes Neurais Artificiais" [Online]. Disponível em: ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia006_03/topico5_03.pdf. [Acesso em 16 06 2018].

[6] [Online]. Disponível em: https://www.python.org/psf/. [Acesso em 16 06 2018].

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, “Scikit-learn: Machine Learning in Python” [Online]. Disponível em: http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf. [Acesso em 09 07 2018].

[8] [Online]. Disponível em: https://www.scipy.org/about.html. [Acesso em 16 06 2018].

[9] [Online]. Disponível em: https://pandas.pydata.org/about.html. [Acesso em 16 06 2018].

[10] S. B. Kotsiantis, D. Kanellopoulos, P. E. Pintelas, “Data Preprocessing for Supervised Leaning”. International Journal of Computer Science. v. 1, p. 111-117, 2006.

[11] S. Sinharay, H. S. Stern, D. W. Russell, “The Use of Multiple Imputation for the Analysis of Missing Data”. Psychological Methods, v. 6, n. 4, p. 317-329, 2001.

[12] S. J. Yen, Y. S. Lee, “Cluster-based under-sampling approaches for imbalanced data distributions”. Expert Systems with Applications, v. 36, n. 3, p. 5718-5727, 2009.

APÊNDICE A – SCRIPT EM PYTHON

```
########################################## Data Preprocessing ##############################

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd


# Importing the dataset
dataset = pd.read_csv('Banco.csv', sep = ";", decimal =",")
X = dataset.iloc[:, 4:11].values
y_sale = dataset.iloc[:, 11].values
y_time = dataset.iloc[:, 12:14].values


# Encoding categorical data
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_X_1 = LabelEncoder()
X[:, 1] = labelencoder_X_1.fit_transform(X[:, 1])
labelencoder_X_2 = LabelEncoder()
X[:, 2] = labelencoder_X_2.fit_transform(X[:, 2])
labelencoder_X_3 = LabelEncoder()
X[:, 3] = labelencoder_X_3.fit_transform(X[:, 3])
labelencoder_X_4 = LabelEncoder()
X[:, 4] = labelencoder_X_4.fit_transform(X[:, 4])
labelencoder_X_5 = LabelEncoder()
X[:, 5] = labelencoder_X_5.fit_transform(X[:, 5])
labelencoder_X_6 = LabelEncoder()
X[:, 6] = labelencoder_X_6.fit_transform(X[:, 6])
```

```
onehotencoder = OneHotEncoder(categorical_features = [1,2,3,4,5,6])
X = onehotencoder.fit_transform(X).toarray()


# Splitting the dataset into the training set and test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y_sale, test_size = 0.2, random_state = 0)


################################## Possible sampling algorithms ##################################

# Part 1 - Choose the sampling algorithm


"""
# SMOTE - Oversampler
from imblearn.over_sampling import SMOTE
smote = SMOTE(random_state=12, ratio = 'minority')
X_train, y_train = smote.fit_sample(X_train, y_train)
"""

# Random Under Sampler
from imblearn.under_sampling import RandomUnderSampler
randomundersampler = RandomUnderSampler(random_state=0)
X_train, y_train = randomundersampler.fit_sample(X_train, y_train)


# Standardize The data
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
```

```
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)


################################## Possible machine learning algorithms ##################################

# Part 2 - Choose the machine learning algorithm


"""
# Decision Tree
from sklearn import tree
model = tree.DecisionTreeClassifier()
model = model.fit(X_train, y_train)
y_pred = model.predict(X_test)


# Random Forest
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model = model.fit(X_train, y_train)
y_pred = model.predict(X_test)


# Extra Trees
from sklearn.ensemble import ExtraTreesClassifier
model = ExtraTreesClassifier()
model = model.fit(X_train, y_train)
y_pred = model.predict(X_test)


# KNN
```

```
from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier()
model = model.fit(X_train, y_train)
y_pred = model.predict(X_test)


# AdaBoost
from sklearn.ensemble import AdaBoostClassifier
model = AdaBoostClassifier()
model = model.fit(X_train, y_train)
y_pred = model.predict(X_test)


# Bagging
from sklearn.ensemble import BaggingClassifier
model = BaggingClassifier()
model = model.fit(X_train, y_train)
y_pred = model.predict(X_test)


# Logistic Regression
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model = model.fit(X_train, y_train)
y_pred = model.predict(X_test)


# Gradient Boosting
from sklearn.ensemble import GradientBoostingClassifier
model = GradientBoostingClassifier()
model = model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

```
"""

# Artificial Neural Network(ANN)
from keras.models import Sequential
from keras.layers import Dense

# Initialising the ANN
model = Sequential()

# Adding the input layer and the first hidden layer
model.add(Dense(activation = "relu", input_dim = 55, units = 6, kernel_initializer = "uniform"))

# Adding the second hidden layer
model.add(Dense(activation="relu", units=6, kernel_initializer="uniform"))

# Adding the output layer
model.add(Dense(units = 1, kernel_initializer = "uniform", activation = "sigmoid"))

# Compiling the ANN
model.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])

# Fitting the ANN to the Training set
model.fit(X_train, y_train, epochs = 5)

y_pred = model.predict(X_test)
y_pred = (y_pred > 0.5)


############################## Making and ploting the Confusion Matrix ##############################
```

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)

# Creating the function to plot the confusion matrix

def plot_confusion_matrix(cm, classes):
    import itertools
    plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    fmt = 'd'
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], fmt),
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('Valor Verdadeiro')
    plt.xlabel('Valor Predito')

# Ploting confusion matrix
plt.figure()
plot_confusion_matrix(cm, classes = [0,1])
plt.show()
```