PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
IEC – Instituto de Educação Continuada
Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data

Melkson Felix Correia da Silva
Raphael São José Rabelo
Romeu Teixeira Rebello
Thiago Ramos Bezerra da Silva

PROPOSTA DE MODELO PARA AUXILIAR NA TOMADA DE DECISÃO DE PROJETOS COM MAIOR VIABILIDADE

Belo Horizonte
2022

Melkson Felix Correia da Silva
Raphael São José Rabelo
Romeu Teixeira Rebello
Thiago Ramos Bezerra da Silva

PROPOSTA DE MODELO PARA AUXILIAR NA TOMADA DE DECISÃO DE PROJETOS COM MAIOR VIABILIDADE

Belo Horizonte
2022

SUMÁRIO

1. Introdução **5**
1.1. Contextualização 5
1.2. O problema proposto 5
1.3. Objetivos 6
2. Coleta de Dados **6**
2.1 Informações da Base 7
3. Processamento/Tratamento de Dados **9**
3.1. Dados Nulos 9
3.2. Dados que não serão utilizados 10
3.3. Cálculo da duração do negócio 11
3.4. Processamento da target 11
4. Análise e Exploração dos Dados **13**
4.1. Análise Estatísticas 13
4.1.1. Análise Univariada e Multivariada das Colunas 14
4.1.1.1. Coluna Escritório 14
4.1.1.1.1. Análise Univariada 14
4.1.1.1.2. Análise Multivariada 14
4.1.1.2. Coluna Data Início 17
4.1.1.2.1. Análise Univariada 17
4.1.1.2.2. Análise Multivariada 17
4.1.1.3. Coluna Tipo 18
4.1.1.3.1. Análise Univariada 18
4.1.1.3.2. Análise Multivariada 19
4.1.1.4. Coluna Duração Negócio 20
4.1.1.4.1. Análise Univariada 20
4.1.1.4.2. Análise Multivariada 20
4.1.1.5. Coluna Valor Mensal 22
4.1.1.5.1. Análise Univariada 22
4.1.1.5.2. Análise Multivariada 22
4.1.1.6. Coluna Valor Fee Sucesso 24
4.1.1.6.1. Análise Univariada 24
4.1.1.6.2. Análise Multivariada 25

4.1.1.7. Coluna Valor da Transação 26
4.1.1.7.1. Análise Univariada 26
4.1.1.7.2. Análise Multivariada 27
4.2. Insights com a target 28
5. Criação de Modelos de Machine Learning 35
6. Interpretação dos Resultados 43
7. Apresentação dos Resultados 45
8. LINKS 46
APÊNDICE 47

1. Introdução

1.1. Contextualização

A empresa objeto do estudo, atualmente, necessita de semanas em diversas reuniões para os envolvidos classificarem os negócios por níveis de interesse.
Este método, além de demorado, tem como base a discussão entre os participantes acerca dos principais negócios da empresa baseando-se no *know-how* dos colaboradores que, muitas vezes, pode ser tendenciosa e incorrer em erros.

1.2. O problema proposto

O problema da empresa objeto do estudo inicia devido à alta demanda por analisar as propostas para novos negócios que muitas vezes requerem muito tempo de análise e dependem da subjetividade dos indivíduos envolvidos.

A alta dependência do recurso humano é um dos problemas. É comum em organizações a detenção do conhecimento por parte de uma ou poucas pessoas que são responsáveis pela análise e possuem mais familiaridade com os negócios considerados mais importantes. No entanto, na empresa em questão, não se vê uma padronização ou automatização através de modelos que possam direcionar para os negócios mais relevantes, baseados em lições aprendidas e padrões que serão alimentados no modelo.

O acúmulo de análises devido a alta demanda, a quantidade de parâmetros, e o período curto de tempo para análise, podem gerar perda de negócios e consequentemente perdas financeiras irreversíveis.
Os dados em questão são de uma empresa privada do ramo de consultoria que serão analisados no período entre 2018 até a atualidade.

1.3. Objetivos

O objetivo principal dessa análise é mostrar, através de modelos de dados, um conjunto de soluções mais produtivas, com foco em prever negócios com maior interesse em um menor tempo de análise. Espera-se também agilizar o processo de elegibilidade dos negócios, reduzindo recursos com pessoal e custos operacionais inerentes.

2. Coleta de Dados

O modelo trabalhará com uma classificação relacionada ao nível de interesse do negócio para a empresa. Utilizaremos um modelo de classificação absorvendo os parâmetros extraídos da base de dados para gerar o resultado esperado.

Os campos são elencados conforme o *dataset* abaixo:

Nome da coluna/campo	Descrição	Tipo
nome	Nome do negócio	Texto
escritorio	Local onde o negócio ocorreu	Texto
data_inicio	Data do início do negócio	Data
area	Área da empresa responsável pelo negócio	Texto
tipo	Tipo do negócio	Texto
palavras_chaves	Palavras chaves do negócio	Texto
origem	Origem do negócio	Texto
iniciais_originadores	Iniciais dos nomes das pessoas que originaram o negócio	Texto
valor_da_transacao	Valor da transação do negócio	Número
valor_fee_successo	Valor que a empresa irá receber em caso de sucesso do negócio	Número
valor_fee_mensal	Valor que a empresa irá receber mensalmente	Número
expectativa_fechamento	Data no qual é esperado em que o negócio conclua com sucesso	Data

status_atual	Status atual do negócio na empresa	Texto
data_conclusao_conforme_status	Data em que ocorreu o respectivo status	Data
qtde_negocios_com_o_cliente	Quantidade de negócios passados ou atuais, com o cliente do respectivo negócio	Número
ult_interacao	Data em que houve a última interação com o respectivo negócio	Data
interesse(target)	Nível de interesse que a empresa possui com o negócio (1 – Maior interesse)	Número

Teremos os seguintes níveis de Interesse, classificados em:
Alto: 1
Médio: 2 a 3
Baixo: 4 a 5

A fonte de dados está no formato CSV, e possui campos com dados ausentes. Os dados sensíveis foram previamente tratados a fim de serem preservados.

2.1 Informações da Base:

A base escolhida possui 1961 linhas e 17 colunas (figuras 1 e 2). Devido à grande quantidade de itens nulos (Figura 3), foi feita tratativa para que alguns campos fossem melhorados com dados mais consistentes para o propósito do trabalho.

```
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   nome                            1961 non-null   object
 1   data_inicio                     1961 non-null   object
 2   escritorio                      1586 non-null   object
 3   area                            1961 non-null   object
 4   tipo                            1961 non-null   object
 5   palavras_chaves                 416 non-null    object
 6   origem                          54 non-null     object
 7   iniciais_originadores           1640 non-null   object
 8   valor_da_transacao              610 non-null    float64
 9   valor_fee_sucesso               538 non-null    float64
 10  valor_fee_mensal                353 non-null    float64
 11  expectativa_fechamento          1009 non-null   object
 12  status_atual                    1961 non-null   object
 13  data_conclusao_conforme_status  816 non-null    object
 14  qtde_negocios_com_o_cliente     1961 non-null   int64
 15  ult_interacao                   1961 non-null   object
 16  interesse(target)               1058 non-null   float64
dtypes: float64(4), int64(1), object(12)
memory usage: 260.6+ KB
```

Figura 1 - Informações da base

	valor_da_transacao	valor_fee_sucesso	valor_fee_mensal	qtde_negocios_com_o_cliente	interesse(target)
count	610.00	538.00	353.00	1961.00	1058.00
mean	96468748.24	1727343.46	53071731.61	2.33	2.59
std	274766815.22	2636951.06	666565332.17	3.61	1.42
min	0.00	0.00	441.94	0.00	1.00
25%	16283588.75	500000.00	90000.00	1.00	1.00
50%	40000000.00	1000000.00	128000.00	1.00	2.00
75%	100000000.00	2000000.00	240000.00	2.00	4.00
max	5000000000.00	40000000.00	12000000000.00	29.00	5.00

Figura 2 – Descrição da base

```
nome                              0
data_inicio                       0
escritorio                      375
area                              0
tipo                              0
palavras_chaves                1545
origem                         1907
iniciais_originadores           321
valor_da_transacao             1351
valor_fee_sucesso              1423
valor_fee_mensal               1608
expectativa_fechamento          952
status_atual                      0
data_conclusao_conforme_status 1145
qtde_negocios_com_o_cliente       0
ult_interacao                     0
interesse(target)               903
dtype: int64
```

Figura 3 – Descrição de itens nulos para posterior tratativa

3. Processamento/Tratamento de Dados

3.1 Dados Nulos

A base possuía muitos campos nulos, aos quais tivemos que dar algumas tratativas de reclassificação. Nota-se que nem todos os campos puderam ser alterados, para que não houvesse distorção dos dados originais da base, conforme descrito abaixo:

Campos Nulos em “Escritório”: substituídos por “Araujo Fontes BH”. Por falta de atualização dos usuários com relação ao volume de negócios deste escritório, criou-se uma cultura do não preenchimento deste campo pois, para eles, estava subentendida esta informação;

Campos Nulos em “Origem”: substituídos por “Interna”; Pelo mesmo motivo do item anterior, culturalmente os usuários não preenchiam a origem, devido ao grande volume de negócios de origem interna e a escassez de externos.

Campos Nulos em “Valor da transação”, “Valor fee sucesso” e “valor fee mensal”: substituídos por “0”;

Campos Nulos em “Interesse”: substituídos por “5”, ou seja, de menor interesse. Em conversa com os usuários, nos foi dito que os negócios de pouco interesse para a empresa sequer eram atualizados os dados, o que nos levou a substituir os nulos por 5;

Campos Nulos em “Data_conlusão”, caso o status seja:

1. “Concluído com sucesso”: pegar dado da coluna “ult interação” (última interação);
2. “Concluído sem sucesso”: pegar dado da coluna “ult interação” (última interação);
3. Campos Nulos em “Expectativa de fechamento”, caso o status seja:
4. “Concluído com sucesso”: pegar dado da coluna data de conclusão conforme status;
5. “Concluído sem sucesso”: pegar dado da coluna data de conclusão conforme status.

Resultado após a trativa dos nulos:

```
df_base.isnull().sum()
```

```
nome                                0
data_inicio                         0
escritorio                          0
area                                0
tipo                                0
palavras_chaves                  1544
origem                              0
iniciais_originadores             321
valor_da_transacao                  0
valor_fee_sucesso                   0
valor_fee_mensal                    0
expectativa_fechamento            530
status_atual                        0
data_conclusao_conforme_status    903
qtde_negocios_com_o_cliente         0
ult_interacao                       0
interesse(target)                   0
dtype: int64
```

Figura 4 - Resultado das tratativas dos campos nulos

3.2 Dados que não serão utilizados

Os negócios que possuem o tipo “institucional” foram removidos, tendo em vista a não aderência aos objetivos do trabalho. Tais tipos são projetos internos, os quais não são relevantes uma vez que é mais fácil decidir a relevância e não ter envolvimento de clientes por sua natureza interna.

Para tanto, foi feito um *drop* na base onde o *index* do tipo Institucional estava presente

```
df_base.drop(df_base.loc[df_base['tipo']=='Institucional'].index, inplace=True)
```

Figura 5 - Exclusão de coluna não relevante

```
df_base.describe()
```

	valor_da_transacao	valor_fee_sucesso	valor_fee_mensal	qtde_negocios_com_o_cliente	interesse(target)
count	610.00	538.00	353.00	1959.00	1056.00
mean	96468748.24	1727343.46	53071731.61	2.33	2.59
std	274766815.22	2636951.06	666565332.17	3.61	1.42
min	0.00	0.00	441.94	0.00	1.00
25%	16283588.75	500000.00	90000.00	1.00	1.00
50%	40000000.00	1000000.00	128000.00	1.00	2.00
75%	100000000.00	2000000.00	240000.00	2.00	4.00
max	5000000000.00	40000000.00	12000000000.00	29.00	5.00

Figura 6 - Resultado da Exclusão dos campos desnecessários

3.3 Cálculo da duração do negócio

Esta informação traz um novo *insight* de uma possível tendência de sucesso ou não de um determinado negócio considerando o tempo que levaria entre a data de início e sua conclusão.

Para isso, usamos a coluna “data_de_inicio” para aplicarmos no cálculo da diferença entre ela e a coluna “data_de_conclusão_conforme_status”.

Foi necessária a conversão destas duas colunas para o tipo data, conforme o código abaixo:

```
df_base["data_conclusao_conforme_status"]=pd.to_datetime(df_base["data_conclusao_conforme_status"])
df_base["data_inicio"]=pd.to_datetime(df_base["data_inicio"])
```

Figura 7 – “data_conclusão_conforme_status” e “data_inicio”

Para mensurar o tempo, foi feito um cálculo simples de subtração entre o campo “data_de_conclusão_conforme_status” e “data_de_inicio”, cujo resultado foi definido em nova coluna chamada “diferença_dias_inclusão_conclusão”:

```
days = df_base["data_conclusao_conforme_status"] - df_base["data_inicio"]
days_diff = days.dt.days
df_base["diferenca_dias_inclusao_conclusao"]= days_diff
```

Figura 8 – Cálculo de tempo entre duas variáveis do tipo data

3.4 Processamento da target

Foi criada nova coluna “interesse_name” onde rotulamos os níveis de interesse conforme mostrado abaixo:

- Alto: 1
- Médio: 2 e 3
- Baixo: 4 e 5

O código e o exemplo abaixo mostram o resultado pretendido:

```
for i,row in df_base.iterrows():
    if  row['interesse(target)'] == 1:
        df_base.at[i,'class_interesse'] = 'Alto'
    elif  row['interesse(target)'] >=2 and  row['interesse(target)'] <= 3:
         df_base.at[i,'class_interesse'] = 'Médio'
    elif   row['interesse(target)'] >= 4:
         df_base.at[i,'class_interesse'] = 'Baixo'
```

```
df_base[['interesse(target)', 'class_interesse']]
```

	interesse(target)	class_interesse
0	5.0	Baixo
1	2.0	Médio
2	5.0	Baixo
3	5.0	Baixo
4	5.0	Baixo
...	...	...
1956	4.0	Baixo
1957	1.0	Alto
1958	1.0	Alto
1959	1.0	Alto
1960	4.0	Baixo

Figura 9 – Criação de nova coluna de acordo com os níveis de interesse e tabela dos resultados

4. Análise e Exploração dos Dados

4.1 Análises Estatísticas

Antes de realizar as análises estatísticas, foi necessário transformar os dados categóricos em dados numéricos para analisarmos as correlações entre eles.

```
df_base['escritorio_cat']=df_base['escritorio'].astype('category').cat.codes
df_base['area_cat']=df_base['area'].astype('category').cat.codes
df_base['tipo_cat']=df_base['tipo'].astype('category').cat.codes
df_base['origem_cat']=df_base['origem'].astype('category').cat.codes
df_base['status_atual_cat']=df_base['status_atual'].astype('category').cat.codes
df_base['class_interesse_cat']=df_base['class_interesse'].astype('category').cat.codes
```

```
base_corr = df_base[['escritorio_cat', 'area_cat', 'tipo_cat', \
                     'origem_cat', 'status_atual_cat', 'qtde_negocios_com_o_cliente', \
                     'diferenca_dias_inclusao_conclusao', 'class_interesse_cat']]
base_corr.corr()

plt.figure(figsize=(16,9))
sns.heatmap(base_corr.corr(),linewidth = 0.30, annot = True)
plt.show()
```

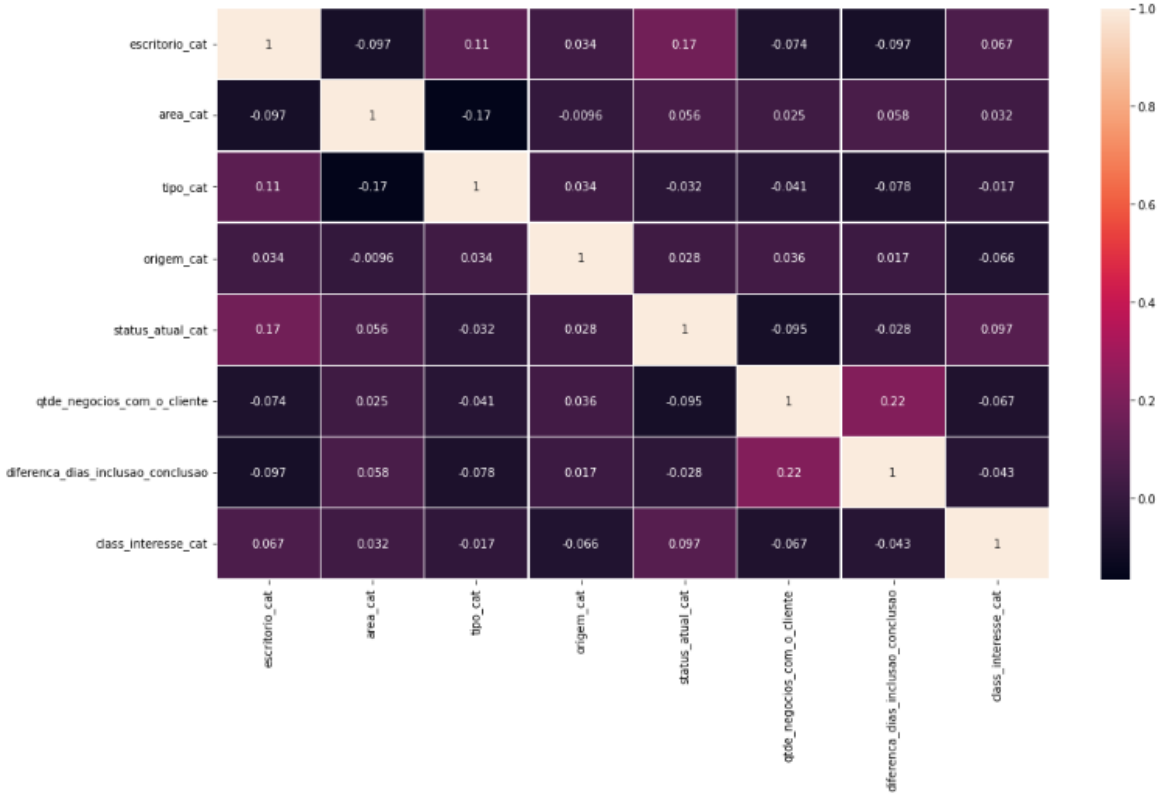


Figura 10 – Transformação de dados categóricos para análise de correlação e gráfico da correlação

4.1.1 Análises Univariadas e Multivariadas das Colunas

4.1.1.1 Coluna Escritório

4.1.1.1.1 Análise Univariada:

```
# informações da coluna
print('Informações da coluna:')
print("\n",df_base['escritorio'].describe())

# dados distintos
print("\nDados distintos")
print(df_base['escritorio'].unique())
```

```
Informações da coluna:

 count                 1959
unique                   4
top       Araujo Fontes BH
freq                  1651
Name: escritorio, dtype: object

Dados distintos
['Araujo Fontes BH' 'Araujo Fontes Goiânia' 'Araujo Fontes Rib. Preto'
 'Araujo Fontes São Paulo']
```

Figura 11 – Informações da coluna Escritório

4.1.1.1.2 Análise Multivariada

Quantidade de negócios por status por escritório (em percentual)

Cruzamos os dados de negócios em relação a seu status atual de negócios com a empresa por escritório para entender quais são as empresas mais envolvidas e seu nível de atividade. O resultado está no gráfico abaixo:

	escritorio	status_atual	T_Esc_Status_Neg	T_Esc_Neg	percentual
0	Araujo Fontes BH	Ativo	722	1651	43.73
1	Araujo Fontes BH	Concluído com sucesso	129	1651	7.81
2	Araujo Fontes BH	Concluído sem sucesso	703	1651	42.58
3	Araujo Fontes BH	Inativo	97	1651	5.88
4	Araujo Fontes Goiânia	Ativo	14	144	9.72
5	Araujo Fontes Goiânia	Concluído com sucesso	5	144	3.47
6	Araujo Fontes Goiânia	Concluído sem sucesso	124	144	86.11
7	Araujo Fontes Goiânia	Inativo	1	144	0.69
8	Araujo Fontes Rib. Preto	Ativo	15	57	26.32
9	Araujo Fontes Rib. Preto	Concluído com sucesso	6	57	10.53
10	Araujo Fontes Rib. Preto	Concluído sem sucesso	31	57	54.39

Figura 12 - Relação de negócios por escritório

Pode-se inclusive inferir que a diferença nos dados indica algum tipo de causa que desconhecemos mas claramente demonstra um diferente nível de resultado para diferentes escritórios. Diferentes teorias que pensamos, indicam uma qualidade melhor no atendimento, uma velocidade diferente na conclusão de etapas ou mesmo algum motivo econômico regional, mas para ir mais a fundo nesse quesito seria necessária uma análise focada no resultado.

Valor_da_transacao, valor_fee_sucesso, valor_mensal por escritorio

Agora vamos analisar o valor das transações frente ao seu valor de sucesso por mês para entender a representatividade de cada escritório na atividade mês a mês.

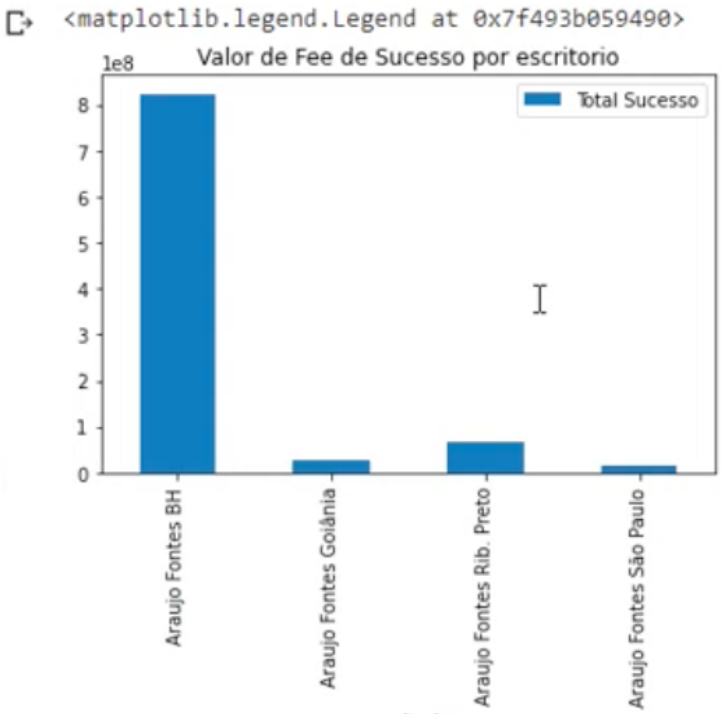


Figura 13 – Valor de fee de sucesso por escritório

Claramente os dados parecem apontar para uma maior atividade no Office (Branch) de Belo Horizonte e um valor bem maior nas fees relacionadas aos negócios, o que também indica que além de mais volume a precificação e lucratividade também são maiores nesta branch.

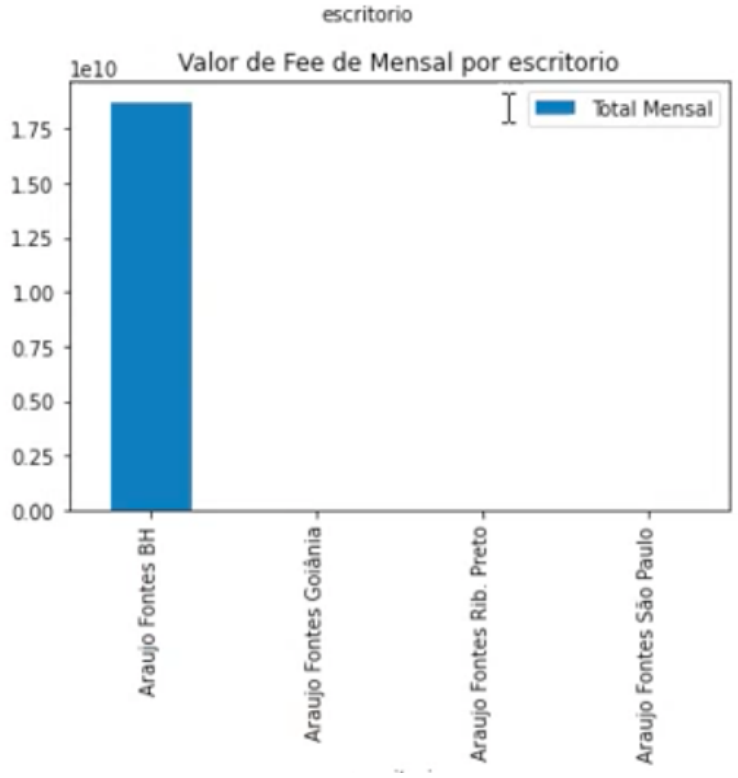


Figura 14 - Valor de fee mensal por escritório

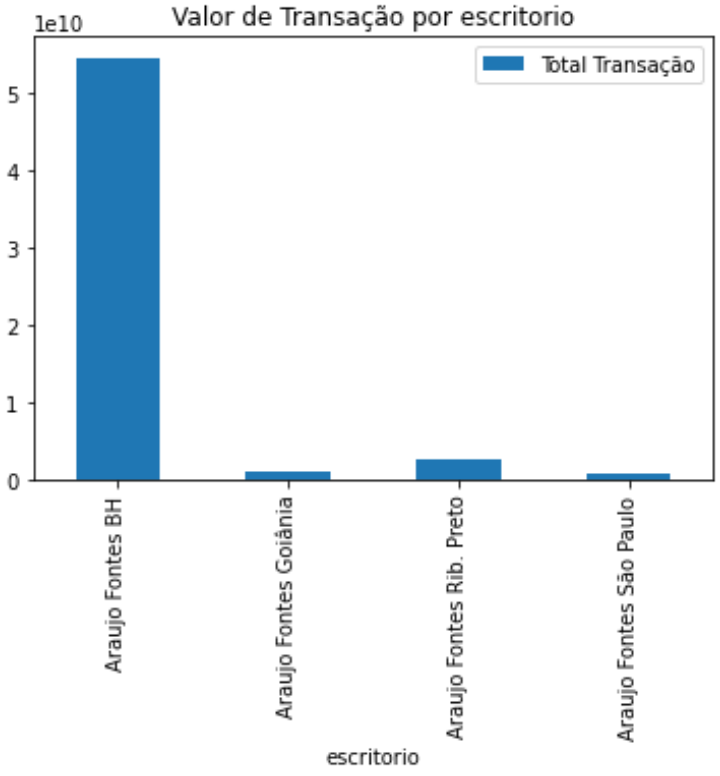


Figura 15 - valor de transação por escritório

4.1.1.2 Coluna Data Inicio

4.1.1.2.1 Análise Univariada

```
# informações da coluna
print('Informações da coluna:')
print("\n",df_base['data_inicio'].describe())
```

```
Informações da coluna:

 count                    1959
unique                    830
top       2018-12-12 00:00:00
freq                       50
first     2008-09-09 00:00:00
last      2022-11-01 00:00:00
Name: data_inicio, dtype: object
```

Figura 16 - Análise univariada da coluna 'data_inicio'

4.1.1.2.2 Análise Multivariada

Quantidade negócios mês/ano e por área

```
df_base.groupby([df_base['data_inicio'].dt.strftime('%Y'), df_base["area"]]).agg({'nome':'count'})
```

		nome
data_inicio	area	
2008	Gestão de Recursos	1
2009	Gestão de Recursos	3
2010	Gestão de Recursos	1
2013	Consultoria	1
	Gestão de Recursos	1
...	...	...
2022	Dívida Captação	11
	Gestão de Recursos	2
	Imobiliário	2
	M&A Sell Side	5
	Projetos Internos	1

84 rows × 1 columns

Figura 17 – Quantidade de negócios por área

Utilizamos o código abaixo para plotar o gráfico da análise multivariada entre as colunas “negocio”, “area” e “ano_inicio”:

```
grafico = df_base.groupby([df_base['data_inicio'].dt.strftime('%Y'), df_base["area"]]).agg({'nome':'count'})\
                 .sort_values(by='nome', ascending=False)\
                 .unstack().plot(figsize=(20,8), marker='o', colormap='viridis', grid=True)
grafico.set_title('Quantidade de Negocios x Area x Ano Inicio', fontsize=25)
grafico.set_xlabel('Ano Data de Inicio')
grafico.set_ylabel('Quantidade de Negocios')
handler, labels = grafico.get_legend_handles_labels()
editar_labels = [re.search(',\s(.+?)\)', label).group(1) for label in labels]
grafico.legend(editar_labels,bbox_to_anchor=(0.01,1), loc=2)
```

Figura 18 – Negócios por área e ano de início

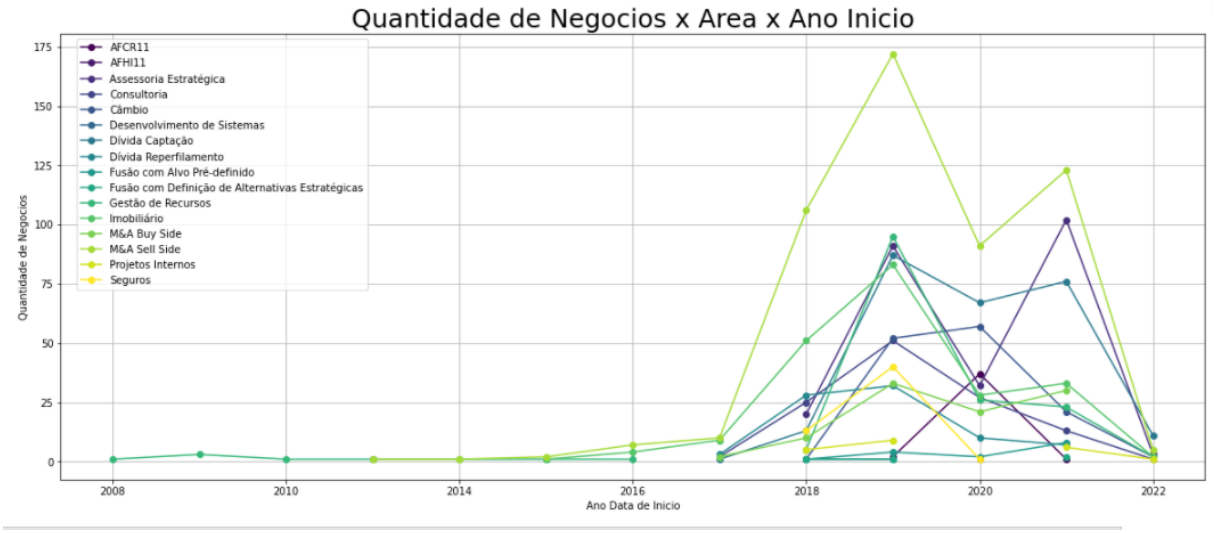


Figura 19 – Gráfico de negócios por área e ano de início

4.1.1.3 Coluna Tipo

4.1.1.3.1 Análise Univariada

A Coluna "tipo" possui quatro tipos únicos para o negócio, que seguem respectivamente a sequência Pré-Lead, Lead, Proposta, Projeto.

```
# informações da coluna
print('Informações da coluna:')
print("\n",df_base['tipo'].describe())

# dados distintos
print("\nDados distintos")
print(df_base['tipo'].unique())
```

```
Informações da coluna:

 count     1959
unique       4
top       Lead
freq       737
Name: tipo, dtype: object
```

Figura 20 - Análise univariada do campo "tipo".

4.1.1.3.2 Análise Multivariada

Quantidade de negócios por tipo

Aqui decidimos criar uma planilha para visualizar onde os negócios estão baseados em diferentes estágios de desenvolvimento. Lead seria um negócio já contactado, projeto um negócio que já estaria encaminhado esperando finalização, proposta seria a formalização e em fase de análise e Pré-lead uma oportunidade de negócio de que temos a possibilidade mas sem contato ou proposta.

tipo	Qtde
Lead	737
Projeto	496
Proposta	393
Pré-Lead	333

Figura 21 – Quantidade de negócios por tipo

Os dados mostram que a empresa possivelmente ainda precisa focar em sua área de relações e de contato com clientes porque existem muito mais projetos em Lead e Pré-Lead do que aqueles em andamento.

Percentual de negócio por tipo por status

Abaixo mostramos mais informações para entender essa relação entre lead e projeto, agora divididas em subcategorias específicas e suas porcentagens.

	tipo	T_Tipo_Neg	status_atual	T_Tipo_Status_Neg	Percentual
0	Lead	737	Ativo	334	45.32
1	Lead	737	Concluído com sucesso	7	0.95
2	Lead	737	Concluído sem sucesso	357	48.44
3	Lead	737	Inativo	39	5.29
4	Projeto	496	Ativo	152	30.65
5	Projeto	496	Concluído com sucesso	136	27.42

Figura 22 - Percentual de negócios por tipo e status

Claramente, o setor comercial precisa focar em transformar estas “Leads” em projetos. A quantidade de Leads ativas e inativas em relação aos projetos está muito alta. Algo não está correspondendo às expectativas, talvez haja muitas leads e oportunidades de negócio recentemente, o que pode ter causado inconsistência por causa da pausa abrupta causada pelo covid-19. Esta é mais uma boa frente de pesquisa para que a empresa possa otimizar este processo e aumentar a quantidade de negócios ou descartar leads inadequados e investir no setor de novos negócios. Vale lembrar que existe a possibilidade de uma *lead* se tornar um projeto, sem necessariamente, se tornar um proposta, oportunidades essas não aproveitadas para alavancar a conclusões com sucesso.

4.1.1.4 Coluna Duração Negócio

4.1.1.4.1 Análise Univariada

```
# informações da coluna
print('Informações da coluna:')
print("\n",df_base['diferenca_dias_inclusao_conclusao'].describe())
```

```
Informações da coluna:

 count   1056.00
mean     276.84
std      328.77
min     -331.00
25%       60.00
50%      218.50
75%      399.75
max     4347.00
Name: diferenca_dias_inclusao_conclusao, dtype: float64
```

Figura 23 - Análise univariada da coluna "diferenca_dias_inclusao_conclusao".

4.1.1.4.2 Análise Multivariada

Média de dias x área x status

```
df_base[df_base["status_atual"] != 'Inativo'][df_base["status_atual"] != 'Ativo']\
       .groupby([df_base['area'], df_base["status_atual"]])\
       .agg({'diferenca_dias_inclusao_conclusao':'mean'})
```

Figura 24 – Média dia por área e status

Logo abaixo, tabela da análise multivariada do item anterior:

		diferenca_dias_inclusao_conclusao
area	status_atual	
AFCR11	Concluído com sucesso	19.750000
	Concluído sem sucesso	145.000000
Assessoria Estratégica	Concluído com sucesso	409.666667
	Concluído sem sucesso	285.176471
Consultoria	Concluído com sucesso	360.419355
	Concluído sem sucesso	196.062500
Câmbio	Concluído sem sucesso	294.666667

Figura 25 - Tabela da relação média dias x área x status

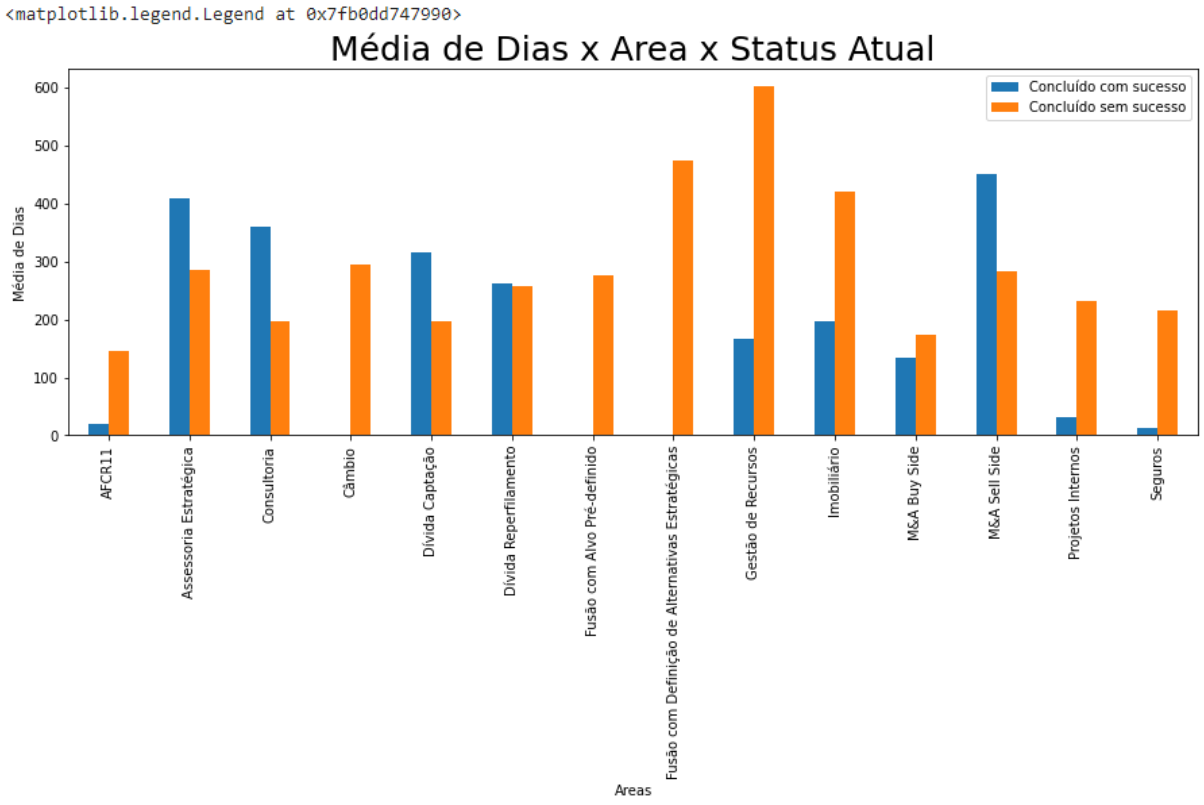


Figura 26 - Gráfico Média de Dias por área e Status Atual dos negócios

Com as informações acima, pudemos ter mais clareza sobre quais tipos de negócios levaram mais ou menos tempo para serem concluídos com ou sem sucesso. Por ter características próprias, cada tipo de negócio mostra oportunidades de análise de viabilidades futuras na relação tempo x retorno.

4.1.1.5 Coluna Valor Mensal

4.1.1.5.1 Análise Univariada

```
# informações da coluna
print('Informações da coluna:')
print("\n",df_base['valor_fee_mensal'].describe())

# dados distintos
print("\nSoma Total")
print(df_base['valor_fee_mensal'].sum())
```

```
Informações da coluna:

 count          1959.00
mean         9563206.36
std        283358660.15
min                0.00
25%                0.00
50%                0.00
75%                0.00
max     12000000000.00
Name: valor_fee_mensal, dtype: float64

Soma Total
18734321259.760002
```

Figura 27 - Descrição da coluna valor fee mensal

4.1.1.5.2 Análise Multivariada

Valor Mensal por tipo e área

	tipo	area	Total_Feed_Mensal
0	Lead	Dívida Captação	260000.00
1	Lead	Imobiliário	1680000.00
2	Projeto	AFCR11	1371205.85
3	Projeto	Assessoria Estratégica	260000.00
4	Projeto	Consultoria	2848220.95
5	Projeto	Câmbio	441.94

Figura 28 – Valor mensal de negócios por tipo e área

Percebe-se, com clareza, que as fases de negócio dos tipos Pré-Lead e Lead são as que menos trazem retorno. Em contrapartida, à medida que os negócios avançam, percebe-se o aumento do retorno financeiro nas fases mais avançadas de Proposta e Projeto. Esta análise é fundamental para ilustrar a importância do trabalho do

setor comercial em avançar nas etapas de negócios, desde o pré-lead até o projeto em si.

O mesmo padrão é percebido na análise seguinte que calcula a média dos valores mensais por tipo e área, conforme é mostrado abaixo:

	tipo	area	Média_Feed_Mensal
0	Lead	Dívida Captação	130000.00
1	Lead	Imobiliário	1680000.00
2	Projeto	AFCR11	1371205.85
3	Projeto	Assessoria Estratégica	86666.67
4	Projeto	Consultoria	81377.74
5	Projeto	Câmbio	441.94
6	Projeto	Dívida Captação	129456.73
7	Projeto	Dívida Reperfilamento	235076.19
8	Projeto	Fusão com Alvo Pré-definido	182500.00
9	Projeto	Fusão com Definição de Alternativas Estratégicas	240000.00
10	Projeto	Gestão de Recursos	780322.99

Figura 29 - Médias de valores por tipo e área

4.1.1.6 Coluna Valor Fee Sucesso

4.1.1.6.1 Análise Univariada

```
# informações da coluna
print('Informações da coluna:')
print("\n",df_base['valor_fee_sucesso'].describe())

# dados distintos
print("\nDados distintos")
print(df_base['valor_fee_sucesso'].unique())
```

```
Informações da coluna:

 count       1959.00
mean      474380.18
std      1581692.31
min            0.00
25%            0.00
50%            0.00
75%        95000.00
max     40000000.00
Name: valor_fee_sucesso, dtype: float64

Dados distintos
[0.00000000e+00 5.00000000e+05 5.00000000e+06 1.50000000e+06
 2.00000000e+06 6.00000000e+05 7.00000000e+06 8.00000000e+05
 1.00000000e+06 1.37500000e+06 3.75000000e+06 4.50000000e+05
 3.50000000e+06 9.00000000e+05 1.20000000e+06 7.00000000e+05
 4.60000000e+06 8.00000000e+06 1.05000000e+06 2.40000000e+04
 1.40000000e+06 5.25000000e+05 2.10000000e+06 5.70000000e+05
 6.60000000e+06 3.00000000e+06 1.75000000e+06 6.00000000e+06
 7.20000000e+06 7.50000000e+05 9.90000000e+05 1.22500000e+06
 1.32000000e+06 2.40000000e+06 8.75000000e+02 8.23683790e+05
 4.00000000e+06 3.30000000e+06 4.30000000e+05 4.01119630e+05
 1.00000000e+05 2.40000000e+05 1.25000000e+05 4.00000000e+05
 1.60000000e+07 9.60000000e+05 1.00000000e+00 1.87500000e+06
 2.00000000e+05 4.96941630e+05 7.98000000e+05 1.25000000e+06
 1.00142500e+06 1.62500000e+06 2.50000000e+05 3.50000000e+05
 3.36000000e+06 4.50000000e+06 6.86000000e+05 4.91514747e+06
```

Figura 30 – Valores de fee de sucesso

4.1.1.6.2 Análise Multivariada

Média dos valores Sucesso por tipo e área

Conforme mostrado abaixo, nota-se uma concentração de negócio (em termos financeiros) nas áreas de Dívida e Captação, Dívida e Reperfilamento, Fusão com Alvo Pré-Definido, Imobiliário e M&A Sell Side. Esta análise é fundamental na tomada de decisão no sentido do direcionamento e foco em novos negócios e aprimoramento dos negócios atuais. Mostra, também, quais áreas estão mais deficitárias, para que a empresa consiga perceber padrões negativos em cada tipo de negócio por área.

		valor_fee_sucesso
area	tipo	
Assessoria Estratégica	Projeto	1112333.33
	Proposta	1618437.50
Consultoria	Projeto	145890.13
	Proposta	80000.00
Câmbio	Projeto	43224.44
Dívida Captação	Projeto	1405673.60
	Proposta	1547000.00
Dívida Reperfilamento	Projeto	951889.13
	Proposta	1732976.43
Fusão com Alvo Pré-definido	Projeto	2035000.00
	Proposta	1850000.00
Fusão com Definição de Alternativas Estratégicas	Projeto	1200000.00
Imobiliário	Projeto	1650138.70

Figura 31 - Tabela com Média do valor de sucesso x tipo x área

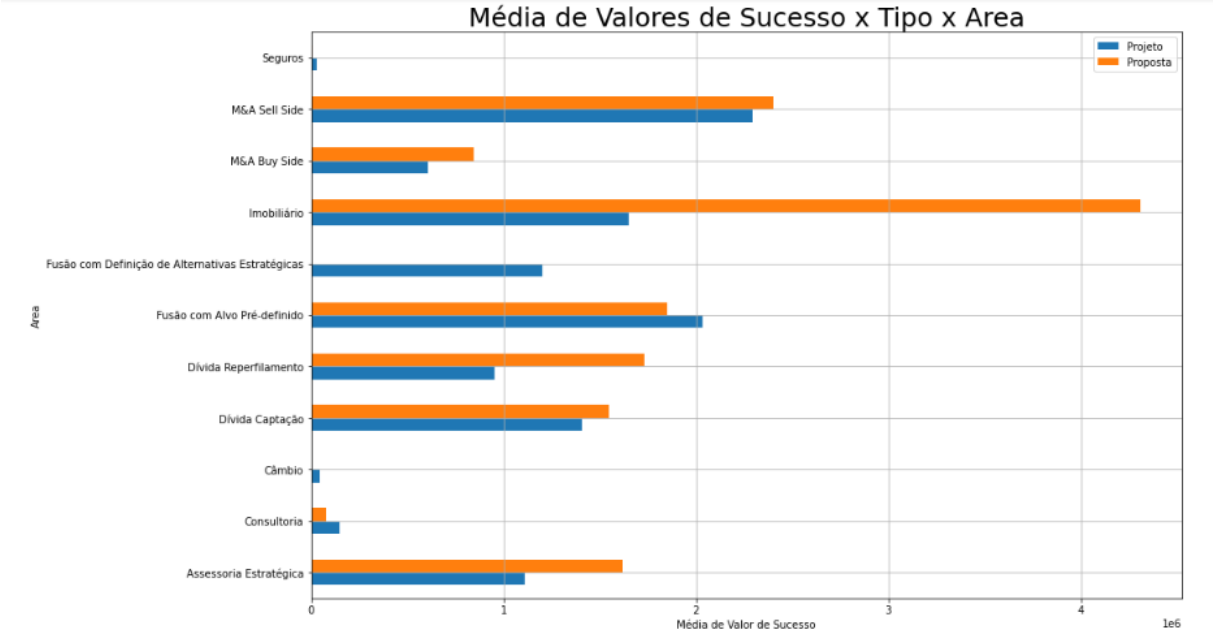


Figura 32 - Gráfico da relação Média do valor de sucesso x tipo x área

4.1.1.7 Coluna Valor da Transação

4.1.1.7.1 Análise Univariada

```
# informações da coluna
print('Informações da coluna:')
print("\n",df_base['valor_da_transacao'].describe())

# Valor total
print("\nValor Total")
print(df_base['valor_da_transacao'].sum())
```

```
Informações da coluna:

 count         1959.00
mean      30038762.85
std      159619516.27
min              0.00
25%              0.00
50%              0.00
75%       13000000.00
max     5000000000.00
Name: valor_da_transacao, dtype: float64

Valor Total
58845936427.5
```

Figura 33 - Análise univariada do campo Valor de Transação

4.1.1.7.2 Análise Multivariada

Valor de transação por tipo e área

Conclui-se que as maiores áreas de negócio da empresa são: imobiliário, assessoria estratégica, M&A Sell Side e dívida captação. Esta análise nos mostra em quais áreas a empresa está deficitária de ações e atenção aos negócios, o que nos leva a outro *insight:* o custo operacional x retorno financeiro ainda mostra se ainda é viável a continuidade de alguns segmentos da empresa, como Consultoria, Câmbio ou se estas áreas poderiam ser fundidas com outras de interesses e processos similares, visando a redução com diversos tipos de recursos não produtivos.

	tipo	area	Total_Valor_Transação
31	Pré-Lead	Imobiliário	300000000.00
30	Pré-Lead	Dívida Captação	100000000.00
28	Proposta	M&A Sell Side	6847355000.00
23	Proposta	Dívida Captação	2198000000.00
26	Proposta	Imobiliário	1206000000.00
21	Proposta	Assessoria Estratégica	864500000.00
24	Proposta	Dívida Reperfilamento	841790000.00
27	Proposta	M&A Buy Side	315000000.00
25	Proposta	Fusão com Alvo Pré-definido	200000000.00
22	Proposta	Consultoria	30040000.00
29	Proposta	Seguros	300000.00
19	Projeto	M&A Sell Side	24776302463.93
12	Projeto	Dívida Captação	3253040063.16
13	Projeto	Dívida Reperfilamento	1937697043.07
14	Projeto	Fusão com Alvo Pré-definido	1375000000.00

Figura 34 - Valores de transação por tipo e área

4.2 Insights com a target

1- Se o tempo de duração dos projetos recentes afeta o interesse do cliente ao longo do tempo.

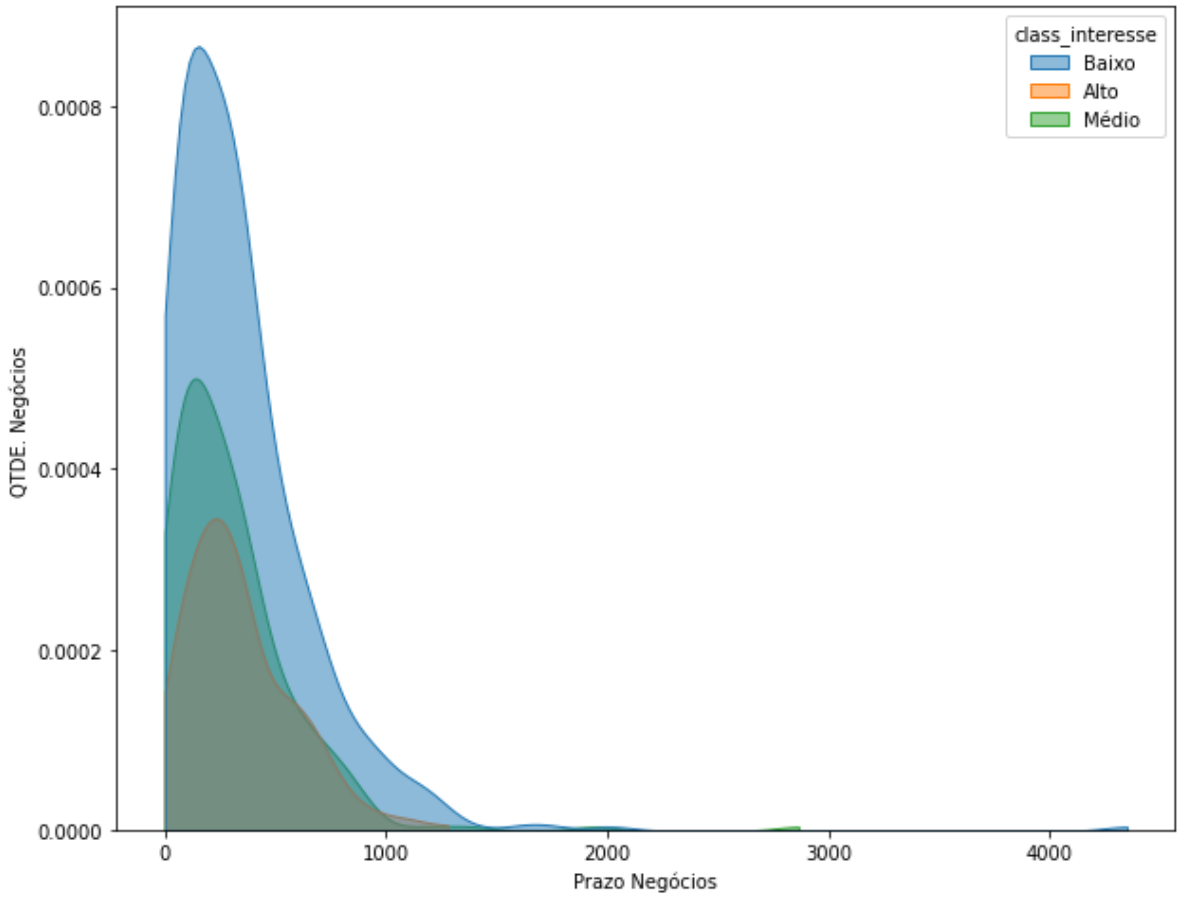


Figura 35 - Relação tempo x interesse x quantidade projetos

O gráfico aponta para uma situação em que o interesse continua forte e rígido nos primeiros 200 e de 200-500 se mantém embora comece a oscilar levemente e após 500 se mantém mais robusto e com menos variações. Ou seja, seria um bom objetivo manter a duração até 400 para que o cliente continue engajado ou mesmo para aumentar a chance de futuros negócios.

2- Buscar entender se o interesse é alterado de acordo com o valor das transações. Ou seja: Será que negócios com interesses mais baixos serão os com valor de transação menores?

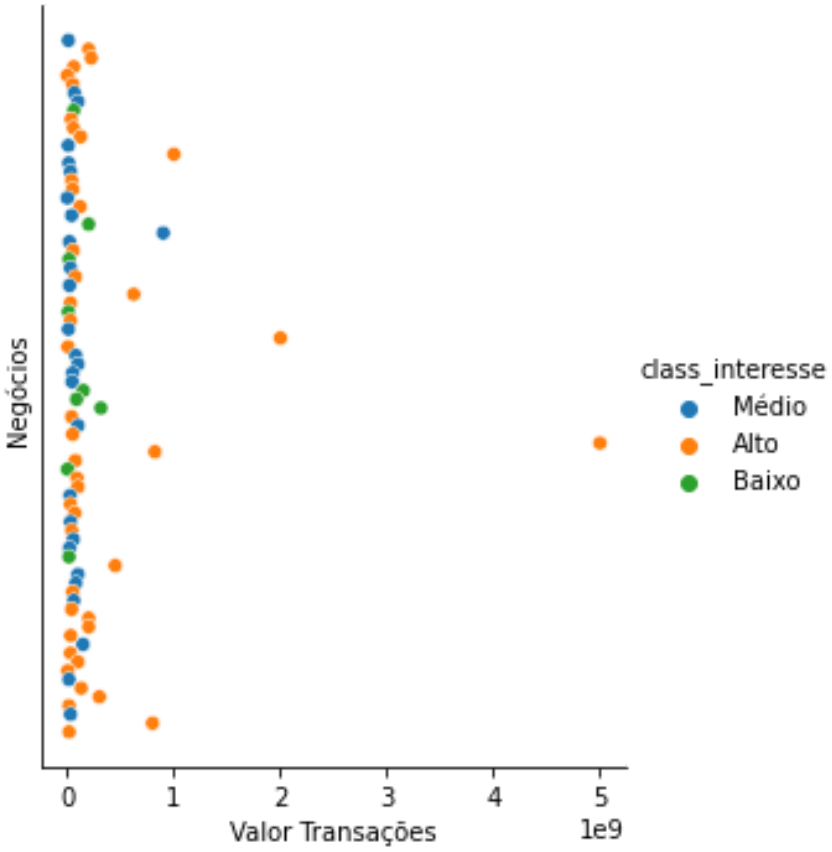


Figura 36 - Relação entre volume de negócios por transações baseada em interesse

Inicialmente uma pequena variação no valor da transação parece alterar o interesse mas não demonstra uma relação contínua. Com o aumento progressivo dos valores, especialmente chegando em R $300 milhões e em diante o comprometimento do cliente com o negócio se estabiliza em alta. Portanto, sim, valores muito altos retém a atenção do cliente, mas mudanças em valores baixos parecem não ser o fator decisor na medida de interesse.

3- Agora vamos comparar o interesse baseado em diferentes áreas da empresa. O objetivo é tentar entender se o nível de interesse condiz, proporcionalmente, com o número de negócios.

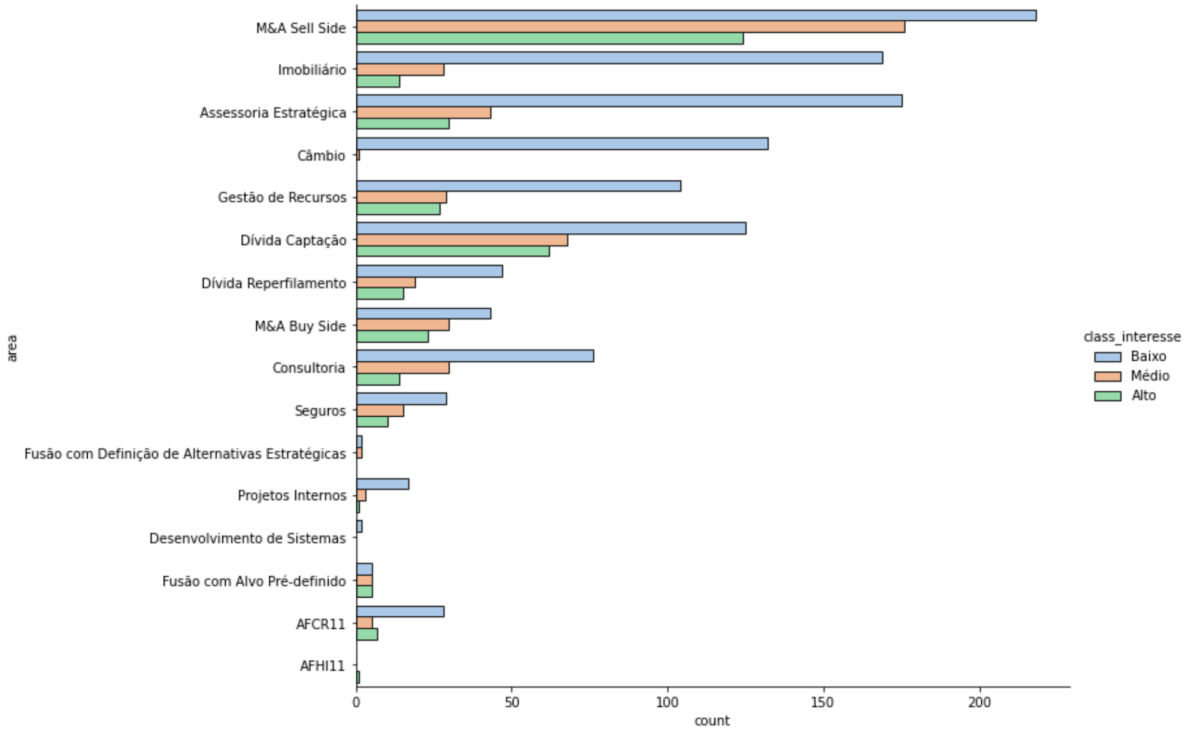


Figura 37 - Contagem de negócios por interesse por cada empresa

Esta foi a análise que retornou os dados mais preocupantes sobre como a empresa está tratando dos seus negócios. Percebe-se uma quantidade absurda de negócios com níveis de interesse baixo e, ao mesmo tempo, pouquíssimos negócios com interesse alto para a empresa. Nos leva a perguntar o porquê de a empresa estar fechando tantos negócios que não são interessantes e vantajosos: Falta de análise?, análises tendenciosas?

Partindo disto, podemos criar modelos que nos mostram quais são os fatores de para conclusão de negócio com sucesso e bom retorno financeiro, bem como mudar o viés decisório atual, mostrando quais os padrões levaram a decidir pelo fechamento de negócios, agora, com resultados baseados em maior favorecimento à empresa.

4- Comparando status para compreender o interesse. Será que negócios com maior interesse tendem a ser concluídos com sucesso?

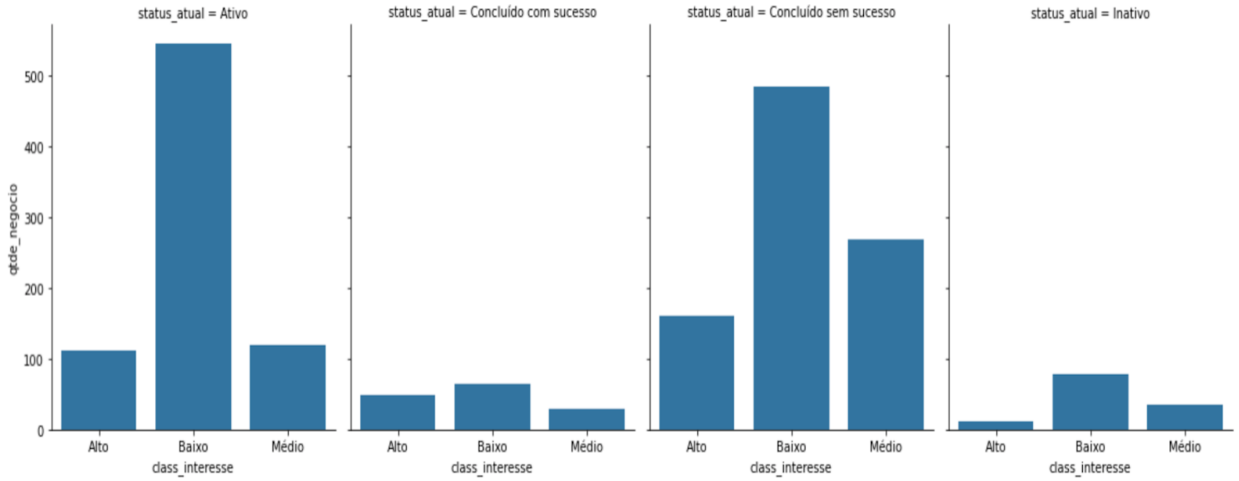


Figura 38 - Status por negócios e interesse

Percebe-se, novamente, que muitos dos negócios que tem status como “Baixo Interesse” estão sendo concluídos com sucesso, bem como estão se mantendo ativos. A partir desse ponto, podem os negócios que foram classificados com baixo interesse ser reclassificados para médio ou alto interesse, uma vez que o nível de conclusão com sucesso e status ativo tem surpreendido?

Do mesmo modo, negócios com o status “Alto” parecem estar estáveis em todos os níveis de conclusão, mas merecem mais atenção de qualquer forma ou uma reanálise questionando se, de fato, a classificação inicial de interesses vista na base tem reflexo na realidade.

5- Comparação do interesse ao longo dos meses e anos. Nosso objetivo é entender se existe alguma variação na busca pela empresa baseada em época do ano ou datas específicas.

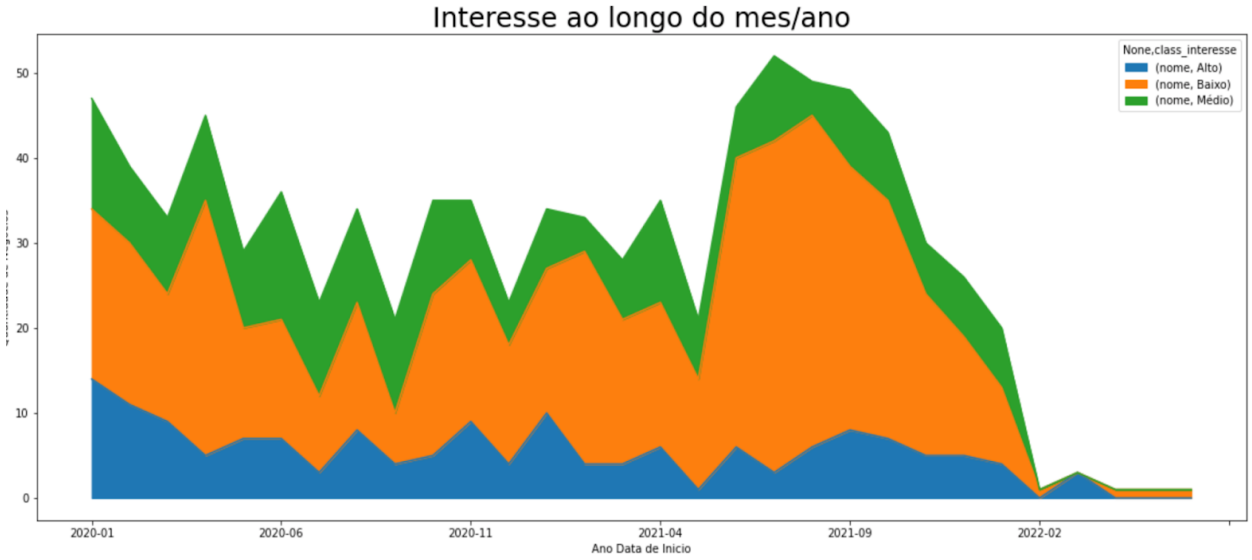


Figura 39 – Relação de interesse ao longo dos meses

Inicialmente, percebe-se que existe uma sazonalidade típica na relação entre meses e interesse ao longo dos anos. A cada mês, em média, existe um padrão de queda e elevação no interesse, ou seja, em um determinado mês, existe uma elevação no interesse e, no mês seguinte, a queda. Esse padrão se repete por vezes, até mudar entre junho e julho de 2021, onde existe uma retomada da elevação do interesse a um nível recorde. O declínio visto ao final do gráfico é apenas por não existirem mais dados coletados, por volta de fevereiro de 2022.

6- Aqui vamos analisar o interesse de acordo com a quantidade geral de negócios com o cliente. Será que quanto mais negócios com um cliente, maior será o interesse?

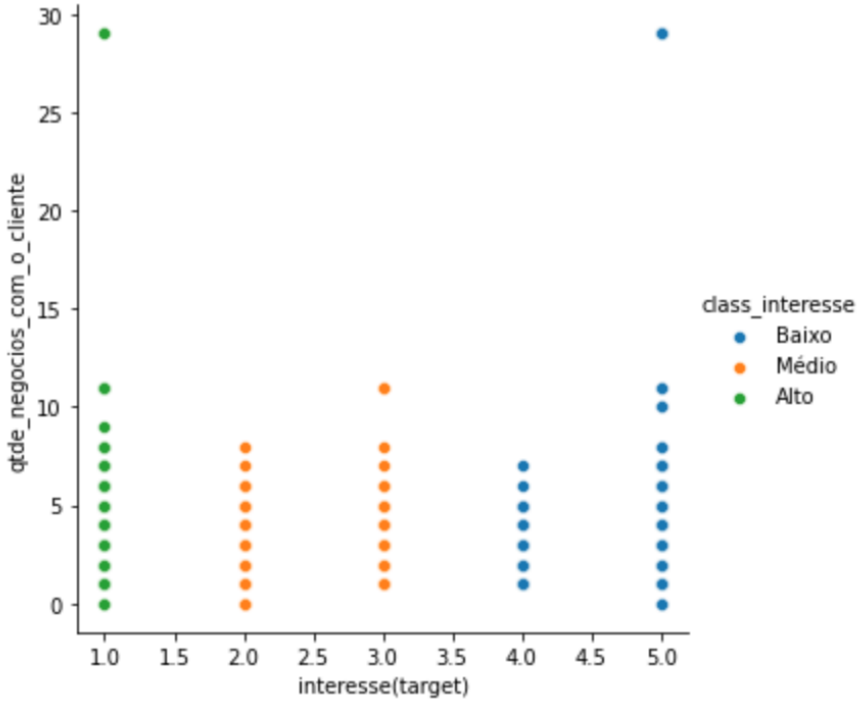


Figura 40 - Relação quantidade de negócios por interesse

Independentemente do nível de interesse, parece haver continuidade do negócio. Seria interessante analisar o porquê de clientes com baixo interesse não subirem de status para alto interesse mesmo tendo negócios constantes. Aparentemente existe uma concomitância entre diferentes níveis por volume também. Existem dois "outliers" que representam clientes que têm uma quantidade de negócios alto e similar porem nível de interesse diferentes e independentes (não sabemos explicar o porquê nesta variância).

7- Comparação entre valor da transação e o valor independente da fee por sucesso. Aqui está uma comparação entre o valor calculado da relação fees de sucesso com o valor total da transação.

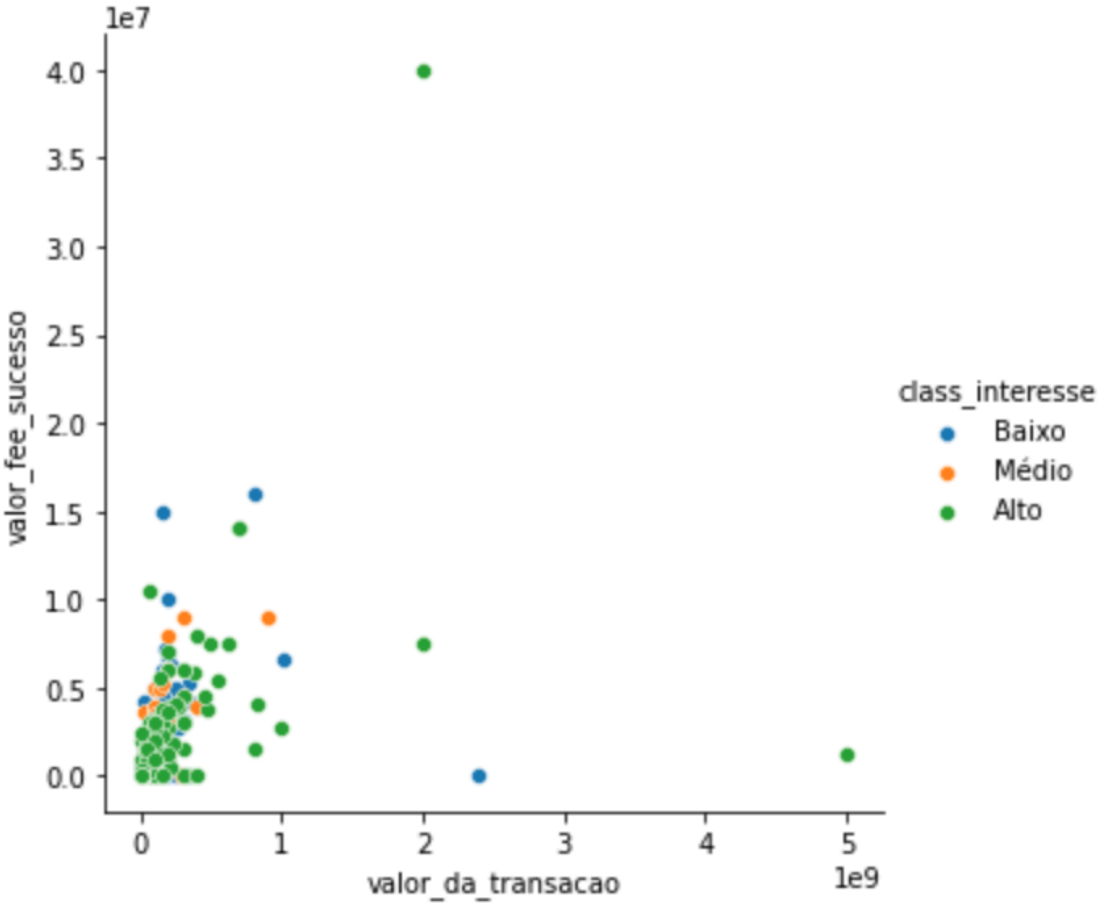


Figura 41 – Gráfico da relação dos fees de sucesso por valor de transação

Os negócios de maior interesse da empresa estão visivelmente relacionados com transações de valor baixo, não possibilitando, consequentemente e proporcionalmente, valores altos de fee de sucesso.

Os negócios categorizados na base como baixo interesse, são os que tem trazido maiores valores de fee de sucesso. Isso mostra como a empresa tem insistido em uma análise enviesada de suas oportunidades de negócio.

5. Criação de Modelos de Machine Learning

Para a criação dos modelos, inicialmente, consideramos relevantes colunas da base categorizadas para as quais atribuímos valores numéricos para o processamento.

- escritorio_cat
- area_cat
- tipo_cat
- class_interesse_cat
- qtde_negocios_com_o_cliente
- valor_da_transação

	escritorio	escritorio_cat
0	Araujo Fontes BH	0
1	Araujo Fontes Goiânia	1
2	Araujo Fontes Rib. Preto	2
3	Araujo Fontes São Paulo	3

Figura 42 - Descrição da categoria escritório

0	AFCR11	0
1	AFHI11	1
2	Assessoria Estratégica	2
3	Consultoria	3
4	Câmbio	4
5	Desenvolvimento de Sistemas	5
6	Dívida Captação	6
7	Dívida Reperfilamento	7
8	Fusão com Alvo Pré-definido	8
9	Fusão com Definição de Alternativas Estratégicas	9
10	Gestão de Recursos	10
11	Imobiliário	11
12	M&A Buy Side	12
13	M&A Sell Side	13
14	Projetos Internos	14
15	Seguros	15

Figura 43 - Descrição da categoria áreas

	tipo	tipo_cat
0	Lead	0
1	Projeto	1
2	Proposta	2
3	Pré-Lead	3

Figura 44 - Descrição da categoria tipo

A coluna class_interesse_cat, que antes continha os valores 0, 1 e 2, foi reclassificada e atribuída nova coluna class_interesse_alto, cujos valores 0 e 2 (antes médio e baixo interesse), receberam valores 0 (interesse baixo) e o valor 1 continuou com valor 1 (interesse alto). Esse ajuste foi necessário pois o modelo retornava baixa acurácia (em torno de 50% a 60%), enquanto estávamos considerando 3 classes de target - baixo, médio e alto interesse. Após a reclassificação, percebemos que o processamento ficou mais fluido e a acurácia imediatamente saltou para 83% em praticamente todos os modelos. Isso se explica pois, por se tratar de um modelo de classificação, onde a abordagem da target é booleana, o algoritmo não conseguia realizar uma predição com boa acurácia com mais de 02 classes.

	class_interesse_cat	class_interesse_alto
0	0	1.00
1	1	0.00
2	2	0.00

Figura 45 - Reclassificação da coluna class_interesse_cat

Ressaltamos que tal ajuste foi feito após diversas tentativas de aprimoramento da acurácia, muito embora esteja sendo já mostrado para atualizar tais ajustes das colunas de forma a evitar repetições neste trabalho.

As features escolhidas, dentre as diversas oriundas da base, são as que mais tem relevância, de acordo com o modelo de negócio e nossa proposta de modelo para auxiliá-los na tomada de decisão.

Após a escolha das *features*, partimos para a preparação do dataframe para o modelo, chamando aquelas colunas explicitadas acima e definindo a coluna class_interesse_alto como nossa *target* e definimos que 20% da base seria usada para testes.

```
#Preparando DataFrame para o modelo

df_base_modelo                                                    =
df_base[['escritorio_cat','area_cat','tipo_cat','class_interesse_alto',
'qtde_negocios_com_o_cliente','valor_da_transacao']]

df_base_modelo['class_interesse_alto']                            =
df_base_modelo["class_interesse_alto"].astype(int)

X=
df_base_modelo[['escritorio_cat','area_cat','tipo_cat','qtde_negocios_c
om_o_cliente','valor_da_transacao']]

Y= df_base_modelo[['class_interesse_alto']]

X_train, X_test, y_train, y_test = train_test_split(X,Y,test_size=0.2)
```

Preparada a base, partimos para a escolha dos modelos para a realização dos testes e posteriores ajustes para a melhoria da acurácia. Foram escolhidos modelos aprendidos em sala de aula na disciplina de *Machine Learning* dessa Pós-graduação em Ciência de Dados e Big Data, bem como são os melhores modelos de classificação. São eles:

- XGBoost
- Decision Tree
- Random Forest
- LightGBM

A parametrização dos modelos foi uma etapa extensa e repetitiva, com muitos ajustes, tendo em vista a baixa acurácia retornada por cada modelo. As configurações que entendemos como mais adequadas para o problema proposto são estas:

```
classifiers=[]
# MODELO 1 - XGB
param_grid1 = {'n_estimators': [400, 700, 1000],
               'colsample_bytree': [0.7, 0.8],
               'max_depth': [15,20,25],
               'reg_alpha': [1.1, 1.2, 1.3],
               'reg_lambda': [1.1, 1.2, 1.3],
               'subsample': [0.7, 0.8, 0.9]
              }
model1 = xgboost.XGBClassifier()
classifiers.append(tuple([model1,param_grid1]))

## MODELO 2 - Decision Tree
param_grid2 = {'criterion': ['entropy', 'gini'],
                'max_depth':range(1,20),
                'min_samples_leaf': range(1,4)
               }
model2 = tree.DecisionTreeClassifier()
classifiers.append(tuple([model2,param_grid2]))

## MODELO 3 - Random Forest
param_grid3 = {'bootstrap': [True, False],
```

```
                'criterion': ['entropy', 'gini'],
                'max_depth': range(1,20),
                'max_features': ['auto'],
                'min_samples_leaf': [1, 2, 4],
                'min_samples_split': [2, 5, 10],
                'n_estimators': [90,100,115,130]
               }
model3 = RandomForestClassifier()
classifiers.append(tuple([model3,param_grid3]))

## MODELO 4 - LGBM
param_grid4 ={'num_leaves': range(1, 20),
              'boosting_type': ['gbdt'],
              'max_depth': range(1,20),
              'n_estimators':[90,100,115,130],
              'min_data_in_leaf':[20]
             }
model4 = lgb.LGBMClassifier()
classifiers.append(tuple([model4,param_grid4]))
```

A partir disso, pudemos realizar a passagem dos parâmetros nos modelos de acordo com os *classifiers* criados, bem como realizar o treinamento da base e a predição:

```
mod=[]
for modelo, parametros in classifiers:
  print("-------------- MODELO - %s --------\n"%(modelo))
  gs = GridSearchCV(
            estimator=modelo,
            param_grid=parametros,
            cv=3,
            n_jobs=-1,
            verbose=2)
  fitted_model = gs.fit(X_train, y_train)

  print("#Melhor Acurácia do modelo:\n ",fitted_model.best_score_)
      print("#Melhores      Parametros      para      o      modelo:\n
",fitted_model.best_params_)

  y_pred= fitted_model.predict(X_test)
  acc = accuracy_score(y_test, y_pred)
  print("#Acurácia:              \n %s"%(acc))

          mod.append({'modelo':modelo,     'melho_modelo':fitted_model,
'y_pred':y_pred})

  cm = confusion_matrix(y_test, y_pred)
  tn, fp, fn, tp = cm.ravel()
  cm_reorganizada = np.array([[tp, fn], [fp, tn]])
  print("#Matriz de Confusão:              \n %s"%(cm_reorganizada))
```

Nesse processo, obtivemos como resultado a melhor acurácia do teste, melhores parâmetros para o modelo, a acurácia do modelo, a matriz de confusão, a visualização gráfica da matriz de confusão e a visualização do relatório de classificação, conforme mostrado abaixo para cada modelo:

---------- MODELO - XGBClassifier() --------
Fitting 3 folds for each of 486 candidates, totalling 1458 fits
#Melhor Acurácia do modelo: 0.8391793586954134
#Melhores Parametros para o modelo: {'colsample_bytree': 0.7, 'max_depth': 15, 'n_estimators': 400, 'reg_alpha': 1.3, 'reg_lambda': 1.3, 'subsample': 0.9}
#Acurácia: **0.8061224489795918**
#Matriz de Confusão: [[16 66] [10 300]]

#Visualização da importância de cada campo para o modelo:

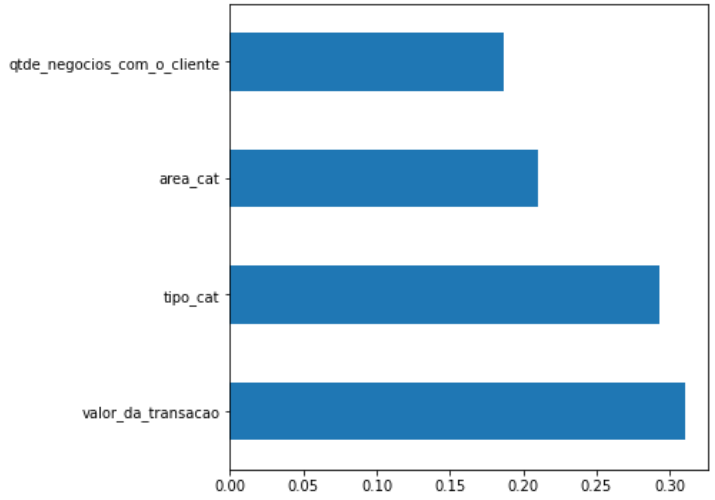


Figura 46 - Gráfico da importância das features para modelo XGBClassifier

#Visualização Gráfica da Matriz de Confusão

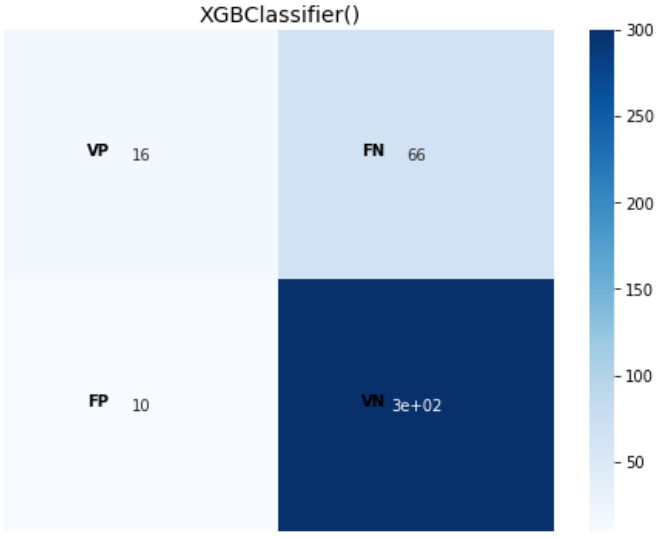


Figura 47 - Gráfico da matriz de confusão do modelo XGBClassifier

#Visualização do relatório de classificação

	precision	recall	f1-score	support
0	0.82	0.97	0.89	310
1	0.62	0.20	0.30	82
accuracy			0.81	392

---------- MODELO - Decision Tree Classifier() --------
Fitting 3 folds for each of 114 candidates, totalling 342 fits
#Melhor Acurácia do modelo: 0.8398215912226593
#Melhores Parametros para o modelo: {'criterion': 'entropy', 'max_depth': 1, 'min_samples_leaf': 1}
#Acurácia: **0.7908163265306123**
#Matriz de Confusão: [[0 82][0 310]]

#Visualização da importância de cada campo para o modelo:

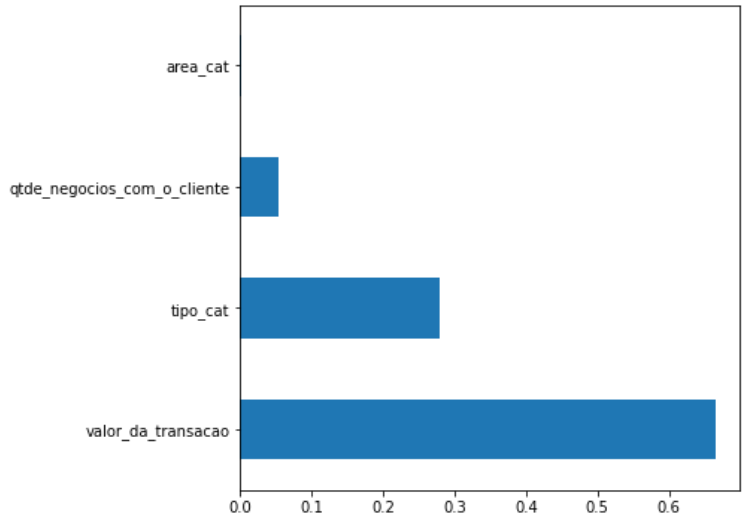


Figura 48 - Gráfico da importância das features para modelo árvore de decisão

#Visualização Gráfica da Matriz de Confusão

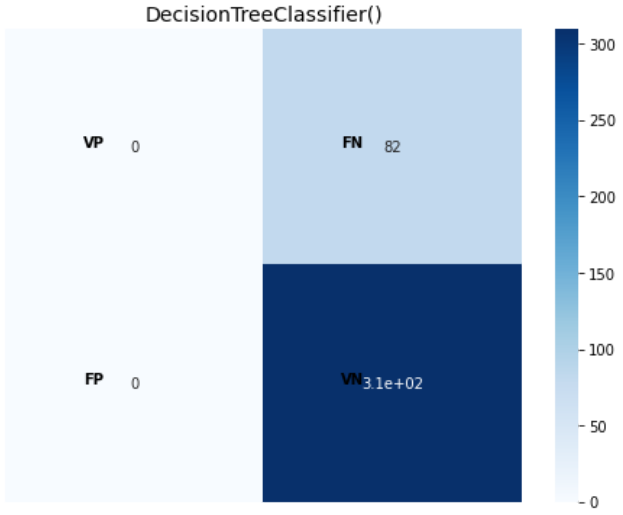


Figura 49 - Gráfico da matriz de confusão do modelo árvore de decisão

#Visualização do relatório de classificação

	precision	recall	f1-score	support
0	0.79	1.00	0.88	310
1	0.00	0.00	0.00	82
accuracy			0.79	392

---------- MODELO - RandomForestClassifier() --------
Fitting 3 folds for each of 2736 candidates, totalling 8208 fits
#Melhor Acurácia do modelo: 0.847473437702224
#Melhores Parametros para o modelo: {'bootstrap': True, 'criterion': 'gini', 'max_depth': 7, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 90}
#Acurácia: **0.7959183673469388**
#Matriz de Confusão: [[9 73][7 303]]

#Visualização da importância de cada campo para o modelo:

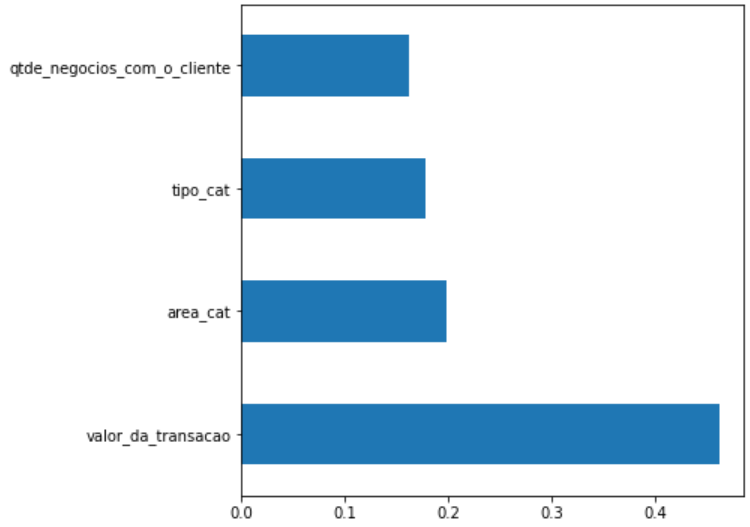


Figura 50 - Gráfico da importância das features para modelo Random Forest

#Visualização Gráfica da Matriz de Confusão

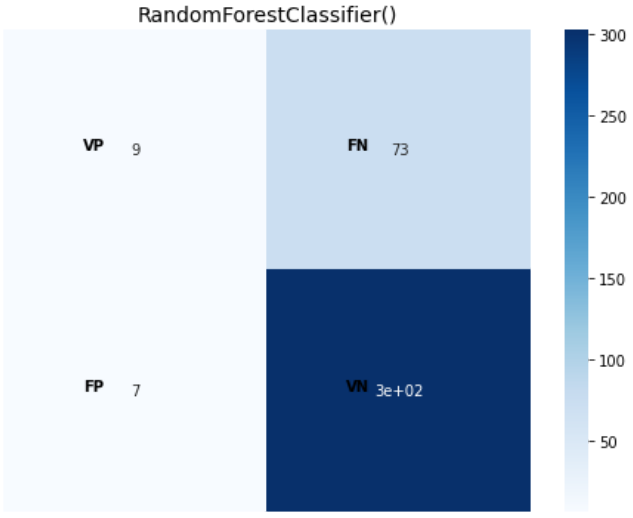


Figura 51 - Gráfico da matriz de confusão do modelo Random Forest

#Visualização do relatório de classificação

	precision	recall	f1-score	support
0	0.81	0.98	0.88	310
1	0.56	0.11	0.18	82
accuracy			0.80	392

---------- MODELO - LGBMClassifier() --------
Fitting 3 folds for each of 1444 candidates, totalling 4332 fits
#Melhor Acurácia do modelo: 0.8449228222090358
#Melhores Parametros para o modelo: {'boosting_type': 'gbdt', 'max_depth': 2, 'min_data_in_leaf': 20, 'n_estimators': 130, 'num_leaves': 3}
#Acurácia: **0.8137755102040817**
#Matriz de Confusão: [[12 70][3 307]]

#Visualização da importância de cada campo para o modelo:

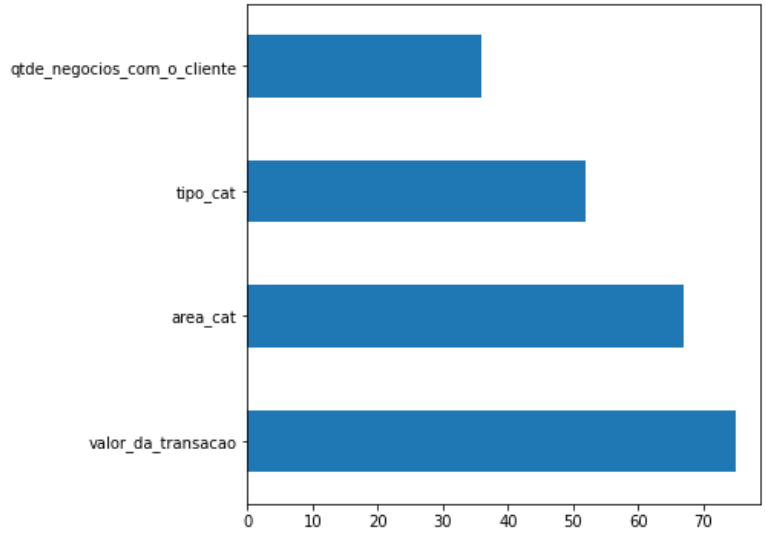


Figura 52 - Gráfico da importância das features para modelo LGBMClassifier

#Visualização Gráfica da Matriz de Confusão

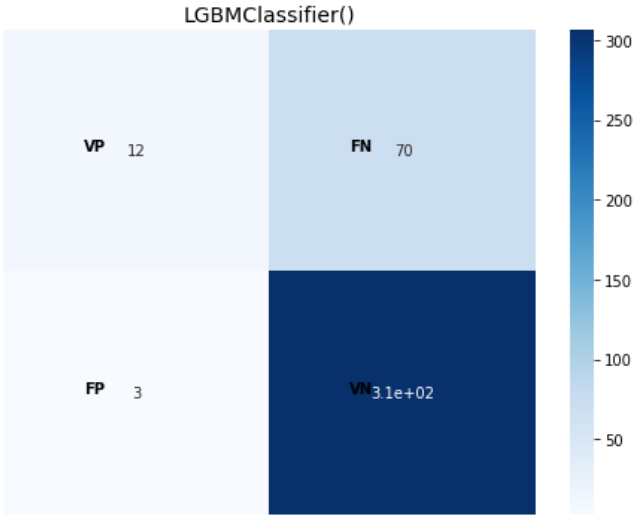


Figura 53 - Gráfico da matriz de confusão do modelo LGBMClassifier

#Visualização do relatório de classificação

	precision	recall	f1-score	support
0	0.81	0.99	0.89	310
1	0.80	0.15	0.25	82
accuracy			0.81	392

6. Interpretação dos Resultados

Na etapa de treinamento dos modelos, percebemos alguns padrões e tivemos alguns *insights* com os quais percebemos, inicialmente, que deveríamos mostrar os escritórios com uma das colunas para ser processada no modelo, porque nos mostra um leque de padrões relacionados à quantidade de negócios e as respectivas tendências de conclusão de sucesso, retorno financeiro etc. Muito embora alguns escritórios possuam um volume global de negócios concluídos com sucesso, proporcionalmente, escritórios menores podem possuir taxas de sucesso maiores. Partindo de uma análise segmentada, a escolha dessa coluna escritório traria ao modelo mais características a serem processadas e, consequentemente, uma análise enviesada para o segmentado.

Após testes com alguns modelos e parâmetros, percebemos que a coluna escritório não era muito relevante para os modelos, conforme informado pelo próprio resultado do uso das *features*. Muito embora a acurácia dos modelos tenha chegado à marca dos 83%, a decisão de criar um novo modelo sem essa *feature* foi necessária, porque estaríamos criando um viés analítico e de treinamento apenas por escritórios, quando, na verdade, gostaríamos que houvesse um viés dos negócios de maneira global, para melhor atender as dinâmicas da empresa.

Consideramos interessante também a inclusão das colunas: área (de negócio), tipo (de negócio), que deram ao modelo mais padrões por segmento. Por exemplo: a área imobiliária tem características próprias, com vários tipos de níveis de negócio (pré-lead, lead, proposta ou projeto), com diferentes resultados atrativos ou não às expectativas da empresa.

Por uma questão de falta de atualização da base, percebemos que a empresa possui muitos negócios que ela considera pouco atrativos (baixo interesse) e poucos negócios que ela considera mais atrativos (alto interesse). Com uma constante atualização da base, os modelos escolhidos posteriormente poderão mostrar dados mais decisórios, enquanto, no momento, nos mostram dados de alerta para a correção de padrões organizacionais.

Dentro desse cenário levantado, concluímos que:

Os modelos XGB e LGBM foram os modelos que mais balancearam os campos em relação à importância. O campo valor_da_transação foi o mais importante e a quantidade_de_negocios_com_o_cliente em ambos os modelos.

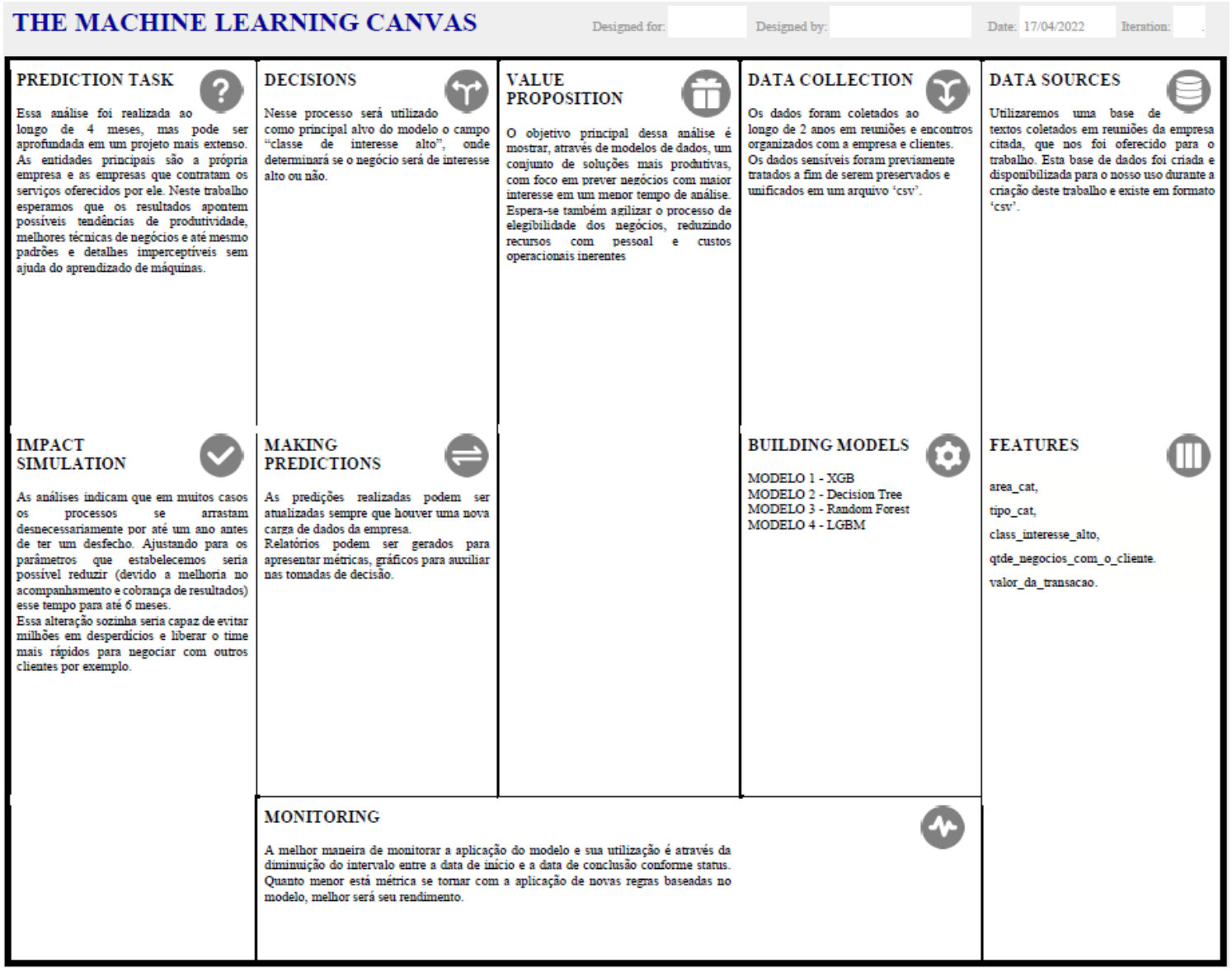
Já os modelos Decision Tree e Random Forest, foram os que mais tiveram discrepância no balanceamento das importâncias dos campos. Curiosamente, no modelo Decision Tree, o campo area_cat teve importância igual a zero, enquanto que no Random Forest, quantidade_de_negocios_com_o_cliente foi novamente o que teve menos importância, seguindo a linha dos modelos XGB e LGBM.

Fica claro que a coluna valor_da_transação foi a coluna mais importante em todos os modelos, o que talvez sugira a linha de decisão que a empresa usa para determinar a importância de seus negócios.

Em termos de acurácia do modelo, todos os 04 modelos apresentaram resultados muito parecidos. O XGB e Decision Tree, obtiveram 80,36% de acurácia. O LGBM obteve 80,31% de acurácia, enquanto que o Random Forest obteve 79,85% de acurácia.

Num contexto geral, LGBM foi o modelo escolhido para auxiliar a empresa no processo de tomada de decisão, levando os processos da empresa a um nível mais alinhado à sua realidade de dados.

7. Apresentação dos Resultados



THE MACHINE LEARNING CANVAS

Designed for: | Designed by: | Date: 17/04/2022 | Iteration: .

PREDICTION TASK

Essa análise foi realizada ao longo de 4 meses, mas pode ser aprofundada em um projeto mais extenso. As entidades principais são a própria empresa e as empresas que contratam os serviços oferecidos por ele. Neste trabalho esperamos que os resultados apontem possíveis tendências de produtividade, melhores técnicas de negócios e até mesmo padrões e detalhes imperceptíveis sem ajuda do aprendizado de máquinas.

DECISIONS

Nesse processo será utilizado como principal alvo do modelo o campo "classe de interesse alto", onde determinará se o negócio será de interesse alto ou não.

VALUE PROPOSITION

O objetivo principal dessa análise é mostrar, através de modelos de dados, um conjunto de soluções mais produtivas, com foco em prever negócios com maior interesse em um menor tempo de análise. Espera-se também agilizar o processo de elegibilidade dos negócios, reduzindo recursos com pessoal e custos operacionais inerentes

DATA COLLECTION

Os dados foram coletados ao longo de 2 anos em reuniões e encontros organizados com a empresa e clientes. Os dados sensíveis foram previamente tratados a fim de serem preservados e unificados em um arquivo 'csv'.

DATA SOURCES

Utilizaremos uma base de textos coletados em reuniões da empresa citada, que nos foi oferecido para o trabalho. Esta base de dados foi criada e disponibilizada para o nosso uso durante a criação deste trabalho e existe em formato 'csv'.

IMPACT SIMULATION

As análises indicam que em muitos casos os processos se arrastam desnecessariamente por até um ano antes de ter um desfecho. Ajustando para os parâmetros que estabelecemos seria possível reduzir (devido a melhoria no acompanhamento e cobrança de resultados) esse tempo para até 6 meses.
Essa alteração sozinha seria capaz de evitar milhões em desperdícios e liberar o time mais rápidos para negociar com outros clientes por exemplo.

MAKING PREDICTIONS

As predições realizadas podem ser atualizadas sempre que houver uma nova carga de dados da empresa.
Relatórios podem ser gerados para apresentar métricas, gráficos para auxiliar nas tomadas de decisão.

BUILDING MODELS

MODELO 1 - XGB
MODELO 2 - Decision Tree
MODELO 3 - Random Forest
MODELO 4 - LGBM

FEATURES

area_cat,
tipo_cat,
class_interesse_alto,
qtde_negocios_com_o_cliente.
valor_da_transacao.

MONITORING

A melhor maneira de monitorar a aplicação do modelo e sua utilização é através da diminuição do intervalo entre a data de início e a data de conclusão conforme status. Quanto menor está métrica se tornar com a aplicação de novas regras baseadas no modelo, melhor será seu rendimento.

8. Links

Link para o repositório:
https://github.com/MelksonF/ProjetoIntegrado_CDBG_PUC.git

APÊNDICE

Programação/Scripts

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
import numpy as np
import re
import seaborn as sns
from sklearn.metrics import
precision_recall_curve,precision_score,recall_score,f1_score,accuracy_score,classification_repor
t,confusion_matrix
from sklearn.ensemble import RandomForestClassifier
from sklearn import svm, tree
import xgboost
import lightgbm as lgb
from sklearn.model_selection import
train_test_split,StratifiedKFold,cross_validate,GridSearchCV
from scipy.stats import uniform as sp_uniform
from matplotlib.pyplot import figure
from mlxtend.plotting import plot_confusion_matrix
import warnings
warnings.filterwarnings('ignore')
```

Importação da base

```
from google.colab import drive
drive.mount('/content/drive')

# Caminho do drive onde está localizada a base
mypath = '/content/drive/MyDrive/Colab Notebooks/Projeto integrado de ciencia de dados/'
Mounted at /content/drive
```

```
# importanto o csv da base com o pandas
df_base = pd.read_csv(mypath+'dataset_projeto_integrado.csv')
```

Análise Iniciais da Base

```
#Formatar campos de valores numericos
pd.options.display.float_format = '{:.2f}'.format
```

Shape (tamanho) da base

```
df_base.shape
```

Pré-Processamento da base

Dados que não serão utilizados

- remover os negócios que possui o tipo Institucional

```
df_base.drop(df_base.loc[df_base['tipo']=='Institucional'].index, inplace=True)
```

Registros Nulos

- os escritórios que tiverem null, considerar como Araújo Fontes BH
- as origens nulas, consideram como Interno
- valor_da_transacao, valor_fee_sucesso e valor_fee_mensal, considerar como 0 os nulos
- Interesse, quando for null, considerar como 5
- Data conclusão conforme status quando for null, caso status seja:
    - Concluído com sucesso: pegar o dado da coluna de ult interação
    - Concluído sem sucesso: pegar o dado da coluna de ult interação
- Expectativa de fechamento quando for null, caso status seja:
    - Concluído com sucesso: pegar o dado da coluna data conclusão conforme status
    - Concluído sem sucesso: pegar o dado da coluna data conclusão conforme status

```
df_base["escritorio"].fillna("Araujo Fontes BH", inplace=True)
df_base["origem"].fillna("Interno", inplace=True)
df_base["valor_da_transacao"].fillna(0, inplace=True)
df_base["valor_fee_sucesso"].fillna(0, inplace=True)
df_base["valor_fee_mensal"].fillna(0, inplace=True)
df_base["interesse(target)"].fillna(5, inplace=True)
df_base.loc[(df_base['data_conclusao_conforme_status'].isna()) & ((df_base['status_atual']
=='Concluído sem sucesso') ^ (df_base['status_atual'] =='Concluído com sucesso')),
'data_conclusao_conforme_status'] = df_base["ult_interacao"]
df_base.loc[(df_base['expectativa_fechamento'].isna()) & ((df_base['status_atual'] =='Concluído
sem sucesso') ^ (df_base['status_atual'] =='Concluído com sucesso')), 'expectativa_fechamento']
= df_base["data_conclusao_conforme_status"]
```

Calcular duração do negócio para os status Concluído com sucesso e Sem sucesso

- calcular a diferença de dias entre data de início e data conclusão conforme status

```
df_base["data_conclusao_conforme_status"]=pd.to_datetime(df_base["data_conclusao_conform
e_status"])
df_base["data_inicio"]=pd.to_datetime(df_base["data_inicio"])


days = df_base["data_conclusao_conforme_status"] - df_base["data_inicio"]
days_diff = days.dt.days
df_base["diferenca_dias_inclusao_conclusao"]= days_diff


r = df_base[df_base["data_conclusao_conforme_status"].notna()]

r[['nome', 'data_inicio', 'data_conclusao_conforme_status',
'diferenca_dias_inclusao_conclusao']].shape
```

Processamento da target

- criar uma nova coluna , chamada interesse_name, onde:
    - Alto: 1
    - Médio: 2 e 3
    - Baixo: 4 e 5

```
for i,row in df_base.iterrows():
   if  row['interesse(target)'] == 1:
      df_base.at[i,'class_interesse'] = 'Alto'
   elif  row['interesse(target)'] >=2 and  row['interesse(target)'] <= 3:
       df_base.at[i,'class_interesse'] = 'Médio'
   elif   row['interesse(target)'] >= 4:
       df_base.at[i,'class_interesse'] = 'Baixo'
```

```
df_base[['interesse(target)', 'class_interesse']]
```

Análises Estatísticas

Categorizando as colunas

```
df_base['escritorio_cat']=df_base['escritorio'].astype('category').cat.codes
df_base['area_cat']=df_base['area'].astype('category').cat.codes
df_base['tipo_cat']=df_base['tipo'].astype('category').cat.codes
df_base['origem_cat']=df_base['origem'].astype('category').cat.codes
df_base['status_atual_cat']=df_base['status_atual'].astype('category').cat.codes
df_base['class_interesse_cat']=df_base['class_interesse'].astype('category').cat.codes
```

Correlação dos dados

```
base_corr = df_base[['escritorio_cat', 'area_cat', 'tipo_cat', \
                     'origem_cat', 'status_atual_cat', 'qtde_negocios_com_o_cliente', \
                     'diferenca_dias_inclusao_conclusao', 'class_interesse_cat']]
base_corr.corr()
```

```
plt.figure(figsize=(16,9))
sns.heatmap(base_corr.corr(),linewidth = 0.30, annot = True)
plt.show()
```

Coluna Escritório

Univariada

```
# informações da coluna
print('Informações da coluna:')
print("\n",df_base['escritorio'].describe())

# dados distintos
print("\nDados distintos")
print(df_base['escritorio'].unique())
```

Multivariada

Quantidade de negocios por status por escritorio (em percentual)

```
negocios_status_escritorio =
pd.DataFrame(df_base.groupby(['escritorio','status_atual']).agg({'nome': 'count'}).reset_index())
negocios_status_escritorio.rename(columns={'nome': 'T_Esc_Status_Neg'}, inplace=True)

negocios_escritorio = pd.DataFrame(df_base.groupby(['escritorio']).agg({'nome':
'count'}).reset_index())
negocios_escritorio.rename(columns={'nome': 'T_Esc_Neg'}, inplace=True)

g = pd.merge(negocios_status_escritorio, negocios_escritorio, how = 'inner', on = 'escritorio')

g['percentual'] = (g['T_Esc_Status_Neg'] /
                   g['T_Esc_Neg']) *100

g
```

Valor da transação, Fee Sucesso e Fee Mensal por escritorio

```
fee_escritorio_sucesso = df_base.groupby(['escritorio']).agg({
                        'valor_fee_sucesso': [('Total Sucesso', 'sum')]
                        })
fee_escritorio_sucesso_graph = pd.concat([fee_escritorio_sucesso],axis=0).plot.bar()
fee_escritorio_sucesso_graph.set_title('Valor de Fee de Sucesso por escritorio')
handler, labels = fee_escritorio_sucesso_graph.get_legend_handles_labels()
editar_labels = [re.search(',\s(.+?)\)', label).group(1) for label in labels]
fee_escritorio_sucesso_graph.legend(editar_labels,bbox_to_anchor=(1,1), loc=0)

fee_escritorio_mensal = df_base.groupby(['escritorio']).agg({
                        'valor_fee_mensal': [('Total Mensal', 'sum')]
                        })
fee_escritorio_mensal_graph = pd.concat([fee_escritorio_mensal],axis=1).plot.bar()
fee_escritorio_mensal_graph.set_title('Valor de Fee de Mensal por escritorio')
handler, labels = fee_escritorio_mensal_graph.get_legend_handles_labels()
editar_labels = [re.search(',\s(.+?)\)', label).group(1) for label in labels]
fee_escritorio_mensal_graph.legend(editar_labels,bbox_to_anchor=(1,1), loc=0)

fee_escritorio_transacao = df_base.groupby(['escritorio']).agg({
                        'valor_da_transacao': [('Total Transação', 'sum')]
                        })

fee_escritorio_transacao_graph = pd.concat([fee_escritorio_transacao],axis=1).plot.bar()
fee_escritorio_transacao_graph.set_title('Valor de Transação por escritorio')
handler, labels = fee_escritorio_transacao_graph.get_legend_handles_labels()
editar_labels = [re.search(',\s(.+?)\)', label).group(1) for label in labels]
fee_escritorio_transacao_graph.legend(editar_labels,bbox_to_anchor=(1,1), loc=0)
```

Coluna Data Início

Univariada

```
# informações da coluna
print('Informações da coluna:')
print("\n",df_base['data_inicio'].describe())
```

Multivariada

- quantidade de negocios ano e por area

```
df2 = df_base.groupby([df_base['data_inicio'].dt.strftime('%Y'),
df_base["area"]]).agg({'nome':'count'}).reset_index()
df2.rename(columns={'nome': 'T_Neg'}, inplace=True)
df2

grafico = df_base.groupby([df_base['data_inicio'].dt.strftime('%Y'),
df_base["area"]]).agg({'nome':'count'})\
            .sort_values(by='nome', ascending=False)\
            .unstack().plot(figsize=(20,8), marker='o', colormap='viridis', grid=True)
grafico.set_title('Quantidade de Negocios x Area x Ano Inicio', fontsize=25)
grafico.set_xlabel('Ano Data de Inicio')
grafico.set_ylabel('Quantidade de Negocios')
handler, labels = grafico.get_legend_handles_labels()
editar_labels = [re.search(',\s(.+?)\)', label).group(1) for label in labels]
grafico.legend(editar_labels,bbox_to_anchor=(0.01,1), loc=2)
```

Coluna Tipo

Univariada

```
# informações da coluna
print('Informações da coluna:')
print("\n",df_base['tipo'].describe())

# dados distintos
print("\nDados distintos")
print(df_base['tipo'].unique())
```

Informações da coluna:

```
count     1959
```

```
unique       4
top       Lead
freq       737
Name: tipo, dtype: object

Dados distintos
['Lead' 'Projeto' 'Proposta' 'Pré-Lead']
```

Multivariada

- quantidade de negócios por tipo

```
df3 = df_base.groupby(['tipo']).agg({'nome':'count'}).reset_index()
df3.rename(columns={'nome': 'T_Tipo_Neg'}, inplace=True)
df3
```

- percentual de negócios por tipo e status

```
df4 = negocios_tipo_status =
df_base.groupby(['tipo','status_atual']).agg({'nome':'count'}).reset_index()
df4.rename(columns={'nome': 'T_Tipo_Status_Neg'}, inplace=True)

g2 = pd.merge(df3, df4, how = 'inner', on = 'tipo')

g2['Percentual'] = (g2['T_Tipo_Status_Neg'] /
                g2['T_Tipo_Neg']) *100

g2
```

Coluna Duração negócio

Univariada

```
# informações da coluna
print('Informações da coluna:')
print("\n",df_base['diferenca_dias_inclusao_conclusao'].describe())
```

Multivariada

- média de dias por area e status

```
df_base[df_base["status_atual"] != 'Inativo'][df_base["status_atual"] != 'Ativo']\
    .groupby([df_base['area'], df_base["status_atual"]])\
    .agg({'diferenca_dias_inclusao_conclusao':'mean'}).reset_index()

grafico1 = df_base[df_base["status_atual"] != 'Inativo'][df_base["status_atual"] != 'Ativo']\
    .groupby([df_base['area'], df_base["status_atual"]])\
    .agg({'diferenca_dias_inclusao_conclusao':'mean'}).unstack().plot(figsize=(15,5), kind="bar")
grafico1.set_title('Média de Dias x Area x Status Atual', fontsize=25)
grafico1.set_xlabel('Areas')
grafico1.set_ylabel('Média de Dias')
plt.xticks(rotation=90)
handler, labels = grafico1.get_legend_handles_labels()
editar_labels1 = [re.search(',\s(.+?)\)', label).group(1) for label in labels]
grafico1.legend(editar_labels1,bbox_to_anchor=(1,1), loc=0)
```

Coluna Valor Mensal

Univariada

```
# informações da coluna
print('Informações da coluna:')
print("\n",df_base['valor_fee_mensal'].describe())

# dados distintos
print("\nSoma Total")
print(df_base['valor_fee_mensal'].sum())
```

Multivariada

- valor mensal por tipo e area

```
valor_mensal_por_tipo_area =
df_base[df_base['valor_fee_mensal']>0].groupby(['tipo','area']).agg({
                        'valor_fee_mensal':'sum'}).reset_index()

valor_mensal_por_tipo_area.rename(columns={'valor_fee_mensal': 'Total_Feed_Mensal'},
inplace=True)
valor_mensal_por_tipo_area
```

- média dos valores mensais por tipo e area

```
media_valor_mensal_por_tipo_area =
df_base[df_base['valor_fee_mensal']>0].groupby(['tipo','area']).agg({
                        'valor_fee_mensal':'mean'}).reset_index()

media_valor_mensal_por_tipo_area.rename(columns={'valor_fee_mensal':
'Média_Feed_Mensal'}, inplace=True)
media_valor_mensal_por_tipo_area
```

Coluna Valor Fee Sucesso

Univariada

```
# informações da coluna
print('Informações da coluna:')
print("\n",df_base['valor_fee_sucesso'].describe())
```

```
Informações da coluna:

 count       1959.00
mean      474380.18
std      1581692.31
min            0.00
25%            0.00
50%            0.00
75%        95000.00
max     40000000.00
Name: valor_fee_sucesso, dtype: float64
```

Multivariada

- média dos valores sucesso por tipo e area

```
df6 = df_base[df_base['valor_fee_sucesso'] > 0]\
      .groupby([df_base['area'], df_base["tipo"]])\

.agg({'valor_fee_sucesso':'mean'}).round(2).reset_index().sort_values(['area','valor_fee_sucesso']
,ascending=False)
df6.rename(columns={'valor_fee_sucesso': 'Média_Feed_Sucesso'}, inplace=True)
df6
grafico2 = df_base[df_base["tipo"] != 'Lead'][df_base["tipo"] !=
'Pré-Lead'][df_base['valor_fee_sucesso'] > 0]\
      .groupby([df_base['area'], df_base["tipo"]])\
      .agg({'valor_fee_sucesso':'mean'}).round(2)\
      .unstack().plot(figsize=(15,10), kind='barh', grid=True)
grafico2.set_title('Média de Valores de Sucesso x Tipo x Area', fontsize=25)
grafico2.set_xlabel('Média de Valor de Sucesso')
grafico2.set_ylabel('Area')
plt.xticks(rotation=0)
handler, labels = grafico2.get_legend_handles_labels()
editar_labels2 = [re.search(',\s(.+?)\)', label).group(1) for label in labels]
grafico2.legend(editar_labels2,bbox_to_anchor=(1,1), loc=0)
```

- Total de Valor Mensal por Tipo e Area

```
df7 = df_base[df_base['valor_fee_mensal'] > 0]\
      .groupby([df_base['area'], df_base["tipo"]])\

.agg({'valor_fee_mensal':'sum'}).round(2).sort_values(['area','valor_fee_mensal'],ascending=Fals
e).reset_index()

df7.rename(columns={'valor_fee_mensal': 'Total_Feed_Sucesso'}, inplace=True)
df7

grafico3 = df_base[df_base["tipo"] != 'Lead'][df_base["tipo"] !=
'Pré-Lead'][df_base['valor_fee_mensal'] > 0]\
```

```
    .groupby([df_base['area'], df_base["tipo"]])\
    .agg({'valor_fee_mensal':'count'})\
    .unstack().plot(figsize=(15,10), kind='barh')
grafico3.set_title('Total de Valores Mensais x Tipo x Area', fontsize=25)
grafico3.set_xlabel('Valor Mensal')
grafico3.set_ylabel('Area')
plt.xticks(rotation=0)
handler3, labels3 = grafico3.get_legend_handles_labels()
editar_labels3 = [re.search(',\s(.+?)\)', label).group(1) for label in labels3]
grafico3.legend(editar_labels3,bbox_to_anchor=(1,1), loc=0)
```

Coluna Valor Transação

Univariada

```
# informações da coluna
print('Informações da coluna:')
print("\n",df_base['valor_da_transacao'].describe())

# Valor total
print("\nValor Total")
print(df_base['valor_da_transacao'].sum())
```

```
Valor Total
58845936427.5
```

Multivariada

- valor de transação por tipo e area

```
df8 = df_base[df_base['valor_da_transacao']>0].groupby(['tipo','area']).agg({
                    'valor_da_transacao':
'sum'}).reset_index().sort_values(['tipo','valor_da_transacao'],ascending=False)
```

```
df8.rename(columns={'valor_da_transacao': 'Total_Valor_Transação'}, inplace=True)
df8
```

- média dos valores de transação por tipo e area

```
df8 = df_base[df_base['valor_da_transacao']>0].groupby(['tipo','area']).agg({

'valor_da_transacao':'mean'}).reset_index().sort_values(['tipo','valor_da_transacao'],ascending=False)

df8.rename(columns={'valor_da_transacao': 'Média_Valor_Transação'}, inplace=True)
df8
```

Insights com a target

- insight da base, ex:
- Duração dos negócios que não estão inativos é um fator decisório na classificação do interesse

```
base_g0 = df_base.query("diferenca_dias_inclusao_conclusao > 0 & status_atual != 'Inativo'")
plt.rcParams["figure.figsize"]=10,8
sns.kdeplot(data=base_g0, x="diferenca_dias_inclusao_conclusao", hue="class_interesse",
fill=True, cut=0, alpha=.5)
plt.ylabel('QTDE. Negócios')
plt.xlabel('Prazo Negócios')
```

- Tendência do valor de transação conforme o interesse (ex: negócios com interesses mais baixos são aqueles com valor de transação menores)

```
base_g1 = df_base.where((df_base['valor_da_transacao']>0) & (df_base['status_atual'] != 'Ativo')
& (df_base['status_atual'] != 'Inativo')).dropna().sort_values('data_inicio', ascending=False)
sns.relplot(data=base_g1, x="valor_da_transacao", y="nome", hue="class_interesse" ,
kind="scatter", height=5)
plt.yticks([])
plt.ylabel('Negócios')
```

```
plt.xlabel('Valor Transações')
```

- interesse de acordo com a area (se determinada area tende a ter um maior interesse)

```
df_ultimos_negocios_area = df_base[['area','interesse(target)', 'data_inicio', 'class_interesse',
'class_interesse_cat']].sort_values(by='data_inicio', ascending=False).head(100)
sns.catplot(y="area", hue="class_interesse", kind="count",
        palette="pastel", edgecolor=".10",size=8, aspect = 1.5,
        data=df_base)
```

- como o status afeta o interesse (ex: negócios com maior interesse tendem a ser concluídos com sucesso)

```
grafico4 = pd.DataFrame(df_base.groupby([df_base['status_atual'],
df_base["class_interesse"]]).agg({'nome':'count'})\
            .sort_values(by='status_atual', ascending=False)).reset_index()
grafico4.rename(columns={'nome': 'T_Status_Class_Neg'}, inplace=True)
print(grafico4)

#O status afeta o nível de interesse?
grafico4 = df_base.groupby([df_base['status_atual'], df_base["class_interesse"]],
as_index=False).agg({'nome':'count'}).sort_values(by='class_interesse')
grafico4.columns=['status_atual', 'class_interesse','qtde_negocio']
g = sns.FacetGrid(grafico4, col='status_atual', height=5, aspect=0.8, )
g = g.map(sns.barplot ,'class_interesse','qtde_negocio')
```

- interesse ao longo do mês/ano (se o tempo é um fator decisório na classificação do interesse.....ex: ao longos dos anos o interesse veio aumentando ou diminuindo)

```
##Interesse ao longo do mês/ano
grafico5 = df_base[df_base['data_inicio'] >'2020-01-01']\
.groupby([df_base['data_inicio'].dt.strftime('%Y-%m'), df_base['class_interesse']])\
.agg({'nome':'count'})\
            .sort_values(by='data_inicio', ascending=True)\
            .unstack().plot(figsize=(20,8), kind='area')
grafico5.set_title('Interesse ao longo do mês/ano', fontsize=25)
grafico5.set_xlabel('Ano Data de Inicio')
```

```
grafico5.set_ylabel('Quantidade de Negócios')
handler, labels = grafico5.get_legend_handles_labels()
editar_labels5 = [re.search(',\s(.+?)\)', label).group(1) for label in labels]
grafico.legend(editar_labels5,bbox_to_anchor=(1,1), loc=0)
```

- interesse de acordo com qtde_negocios_com_o_cliente (se quanto maior a qtde de negócios com o cliente o interesse também é maior)

```
#interesse de acordo com qtde_negocios_com_o_cliente
g = sns.FacetGrid(df_base, hue='class_interesse', size=5)
g.map(sns.scatterplot,'interesse(target)','qtde_negocios_com_o_cliente')
g.add_legend()
```

```
#interesse segue a tendencia entre valor de transação e valor fee de sucesso
g = sns.FacetGrid(df_base, hue='class_interesse', size=5, )
g.map(sns.scatterplot,'valor_da_transacao','valor_fee_sucesso')
g.add_legend()
```

Construção, validação e avaliação do Modelo

Colunas para modelo:
escritorio_cat,area_cat,tipo_cat,class_interesse_cat,qtde_negocios_com_o_cliente,
valor_da_transacao

```
#Coluna Escritório
df_base[['escritorio','escritorio_cat']].groupby('escritorio').agg({'escritorio_cat':'max'}).reset_index()

#Coluna Área
df_base[['area','area_cat']].groupby('area').agg({'area_cat':'max'}).reset_index()

#Coluna Tipo
df_base[['tipo','tipo_cat']].groupby('tipo').agg({'tipo_cat':'max'}).reset_index()
```

- Reclassificar a coluna class_interesse_cat para class_interesse_alto quando class_interesse_cat for igual a 1, caso contrário classificar com zero, repectivamente True e False

```
df_base[['class_interesse','class_interesse_cat']].groupby('class_interesse').agg({'class_interesse_cat':'max'}).reset_index()

for i,row in df_base.iterrows():
    if  row['class_interesse_cat'] == 0:
        df_base.at[i,'class_interesse_alto'] = int(1)
    else:
      df_base.at[i,'class_interesse_alto'] = int(0)


df_base[['class_interesse_cat','class_interesse_alto']].groupby('class_interesse_cat').agg({'class_interesse_alto':'max'}).reset_index()

#Preparando DataFrame para o modelo
df_base_modelo =
df_base[['escritorio_cat','area_cat','tipo_cat','class_interesse_alto','qtde_negocios_com_o_cliente','valor_da_transacao']]
df_base_modelo['class_interesse_alto'] = df_base_modelo["class_interesse_alto"].astype(int)

X2= df_base_modelo[['area_cat','tipo_cat','qtde_negocios_com_o_cliente','valor_da_transacao']]
Y2= df_base_modelo[['class_interesse_alto']]

X_train, X_test, y_train, y_test = train_test_split(X2,Y2,test_size=0.2)

classifiers=[]
# MODELO 1 - XGB
param_grid1 = {'n_estimators': [400, 700, 1000],
               'colsample_bytree': [0.7, 0.8],
               'max_depth': [15,20,25],
               'reg_alpha': [1.1, 1.2, 1.3],
               'reg_lambda': [1.1, 1.2, 1.3],
               'subsample': [0.7, 0.8, 0.9]
              }
```

```
model1 = xgboost.XGBClassifier()
classifiers.append(tuple([model1,param_grid1]))

## MODELO 2 - Decision Tree
param_grid2 = {'criterion': ['entropy', 'gini'],
               'max_depth':range(1,20),
               #'n_estimators':[90,100,115,130],
               'min_samples_leaf': range(1,4)
              }
model2 = tree.DecisionTreeClassifier()
classifiers.append(tuple([model2,param_grid2]))

## MODELO 3 - Random Forest
param_grid3 = {'bootstrap': [True, False],
               'criterion': ['entropy', 'gini'],
               'max_depth': range(1,20),
               'max_features': ['auto'],
               'min_samples_leaf': [1, 2, 4],
               'min_samples_split': [2, 5, 10],
               'n_estimators': [90,100,115,130]
              }
model3 = RandomForestClassifier()
classifiers.append(tuple([model3,param_grid3]))

## MODELO 4 - LGBM
param_grid4 ={'num_leaves': range(1, 20),
              'boosting_type': ['gbdt'],
              'max_depth': range(1,20),
              'n_estimators':[90,100,115,130],
              'min_data_in_leaf':[20]
             }
model4 = lgb.LGBMClassifier()
classifiers.append(tuple([model4,param_grid4]))



classifiers

Out[ ]:
```

```
[(XGBClassifier(),
  {'colsample_bytree': [0.7, 0.8],
   'max_depth': [15, 20, 25],
   'n_estimators': [400, 700, 1000],
   'reg_alpha': [1.1, 1.2, 1.3],
   'reg_lambda': [1.1, 1.2, 1.3],
   'subsample': [0.7, 0.8, 0.9]}),
 (DecisionTreeClassifier(),
  {'criterion': ['entropy', 'gini'],
   'max_depth': range(1, 20),
   'min_samples_leaf': range(1, 4)}),
 (RandomForestClassifier(),
  {'bootstrap': [True, False],
   'criterion': ['entropy', 'gini'],
   'max_depth': range(1, 20),
   'max_features': ['auto'],
   'min_samples_leaf': [1, 2, 4],
   'min_samples_split': [2, 5, 10],
   'n_estimators': [90, 100, 115, 130]}),
 (LGBMClassifier(),
  {'boosting_type': ['gbdt'],
   'max_depth': range(1, 20),
   'min_data_in_leaf': [20],
   'n_estimators': [90, 100, 115, 130],
   'num_leaves': range(1, 20)})]

mod=[]
for modelo, parametros in classifiers:
  print("---------- MODELO - %s --------\n"%(modelo))
  gs = GridSearchCV(
          estimator=modelo,
          param_grid=parametros,
          cv=3,
          n_jobs=-1,
          verbose=2)
  fitted_model = gs.fit(X_train, y_train)

  print("#Acurácia do Teste:          \n ",fitted_model.best_score_)
```

```
  print("#Melhores Parâmetros para o modelo:\n ",fitted_model.best_params_)

  y_pred= fitted_model.predict(X_test)
  acc = accuracy_score(y_test, y_pred)
  print("#Acurácia do Modelo:              \n %s"%(acc))

  mod.append({'modelo':modelo, 'melho_modelo':fitted_model, 'y_pred':y_pred})

  cm = confusion_matrix(y_test, y_pred)
  tn, fp, fn, tp = cm.ravel()
  cm_reorganizada = np.array([[tp, fn], [fp, tn]])
  print("#Matriz de Confusão:              \n %s"%(cm_reorganizada))
```

```
---------- MODELO - XGBClassifier() --------

Fitting 3 folds for each of 486 candidates, totalling 1458 fits
#Acurácia do Teste:
  0.8404662656986782
#Melhores Parâmetros para o modelo:
  {'colsample_bytree': 0.8, 'max_depth': 15, 'n_estimators': 400, 'reg_alpha': 1.3, 'reg_lambda':
1.2, 'subsample': 0.8}
#Acurácia do Modelo:
 0.8035714285714286
#Matriz de Confusão:
 [[ 12  62]
 [ 15 303]]
---------- MODELO - DecisionTreeClassifier() --------

Fitting 3 folds for each of 114 candidates, totalling 342 fits
#Acurácia do Teste:
  0.8442915784512671
#Melhores Parâmetros para o modelo:
  {'criterion': 'gini', 'max_depth': 3, 'min_samples_leaf': 1}
#Acurácia do Modelo:
 0.8035714285714286
#Matriz de Confusão:
 [[  4  70]
 [  7 311]]
```

---------- MODELO - RandomForestClassifier() --------

Fitting 3 folds for each of 2736 candidates, totalling 8208 fits
#Acurácia do Teste:
0.8468421939444554
#Melhores Parâmetros para o modelo:
{'bootstrap': True, 'criterion': 'gini', 'max_depth': 10, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 5, 'n_estimators': 90}
#Acurácia do Modelo:
0.798469387755102
#Matriz de Confusão:
[[6 68]
[11 307]]
---------- MODELO - LGBMClassifier() --------

Fitting 3 folds for each of 1444 candidates, totalling 4332 fits
#Acurácia do Teste:
0.8462060662891414
#Melhores Parâmetros para o modelo:
{'boosting_type': 'gbdt', 'max_depth': 2, 'min_data_in_leaf': 20, 'n_estimators': 115, 'num_leaves': 3}
#Acurácia do Modelo:
0.8010204081632653
#Matriz de Confusão:
[[5 69]
[9 309]]

Interpretação dos Resultados

Resultados obtidos na análise

Exploração de dados

#Exploração dos dados coletados dos 4 modelos analisados.

```
for i in mod:
  print('MODELO -',i['modelo'])
  print('#Melhores Parâmetros do Modelo: ',i['melho_modelo'].best_params_)
  print('#Acurácia do Teste:           ',round(i['melho_modelo'].best_score_*100, 2),'%')
  print('#Acurácia do Modelo:          ',round(accuracy_score(y_test, i['y_pred'])*100, 2),'%')
  print("\n#Visualização da importância de cada campo para o modelo:")
  best_rf = i['melho_modelo'].best_estimator_
  feat_importances = pd.Series(best_rf.feature_importances_, index=X_train.columns)
  feat_importances.nlargest(10).plot(kind='barh', figsize=(6, 6))
  plt.show()

  print("\n#Visualização da Matriz de Confusão:")
  conf_matrix = confusion_matrix(y_true=y_test, y_pred=i['y_pred'])
  tn, fp, fn, tp = conf_matrix.ravel()
  cm_reorganizada = np.array([[tp, fn], [fp, tn]])
  fig, ax = plot_confusion_matrix(conf_mat=cm_reorganizada, figsize=(6, 6))
  plt.xticks([])
  plt.yticks([])
  plt.annotate('VP', (0.05,0.02), fontweight='bold')
  plt.annotate('FN', (1.06,0.02), fontweight='bold')
  plt.annotate('FP', (0.05,1.02), fontweight='bold')
  plt.annotate('VN', (1.06,1.02), fontweight='bold')
  plt.xlabel('Predição', fontsize=18)
  plt.ylabel('Real', fontsize=18)
  plt.show()

  print('\n#Visualização do relatório de classificação:')
  target_names = ['0 - Não Altos', '1 - Altos']
  print(classification_report(y_test, i['y_pred'], target_names=target_names),'\n')

  print('-------------------------------------------------------------------------------------------------------\n')
```

```
MODELO - XGBClassifier()
#Melhores Parâmetros do Modelo:  {'colsample_bytree': 0.8, 'max_depth': 15, 'n_estimators': 400,
'reg_alpha': 1.3, 'reg_lambda': 1.2, 'subsample': 0.8}
#Acurácia do Teste:           84.05 %
#Acurácia do Modelo:          80.36 %
```

#Visualização da importância de cada campo para o modelo:
#Visualização da Matriz de Confusão:
#Visualização do relatório de classificação:

	precision	recall	f1-score	support
0 - Não Altos	0.83	0.95	0.89	318
1 - Altos	0.44	0.16	0.24	74
accuracy			0.80	392
macro avg	0.64	0.56	0.56	392
weighted avg	0.76	0.80	0.76	392

MODELO - DecisionTreeClassifier()
#Melhores Parametros do Modelo: {'criterion': 'gini', 'max_depth': 3, 'min_samples_leaf': 1}
#Acurácia do Teste: 84.43 %
#Acurácia do Modelo: 80.36 %

#Visualização da importância de cada campo para o modelo:
#Visualização da Matriz de Confusão:
#Visualização do relatório de classificação:

	precision	recall	f1-score	support
0 - Não Altos	0.82	0.98	0.89	318
1 - Altos	0.36	0.05	0.09	74
accuracy			0.80	392
macro avg	0.59	0.52	0.49	392
weighted avg	0.73	0.80	0.74	392

MODELO - RandomForestClassifier()
#Melhores Parâmetros do Modelo: {'bootstrap': True, 'criterion': 'gini', 'max_depth': 10, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 5, 'n_estimators': 90}

#Acurácia do Teste: 84.68 %
#Acurácia do Modelo: 79.85 %

#Visualização da importância de cada campo para o modelo:
#Visualização da Matriz de Confusão:
#Visualização do relatório de classificação:

	precision	recall	f1-score	support
0 - Não Altos	0.82	0.97	0.89	318
1 - Altos	0.35	0.08	0.13	74
accuracy			0.80	392
macro avg	0.59	0.52	0.51	392
weighted avg	0.73	0.80	0.74	392

MODELO - LGBMClassifier()
#Melhores Parâmetros do Modelo: {'boosting_type': 'gbdt', 'max_depth': 2, 'min_data_in_leaf': 20, 'n_estimators': 115, 'num_leaves': 3}
#Acurácia do Teste: 84.62 %
#Acurácia do Modelo: 80.1 %

#Visualização da importância de cada campo para o modelo:
#Visualização da Matriz de Confusão:
#Visualização do relatório de classificação:

	precision	recall	f1-score	support
0 - Não Altos	0.82	0.97	0.89	318
1 - Altos	0.36	0.07	0.11	74
accuracy			0.80	392
macro avg	0.59	0.52	0.50	392
weighted avg	0.73	0.80	0.74	392
