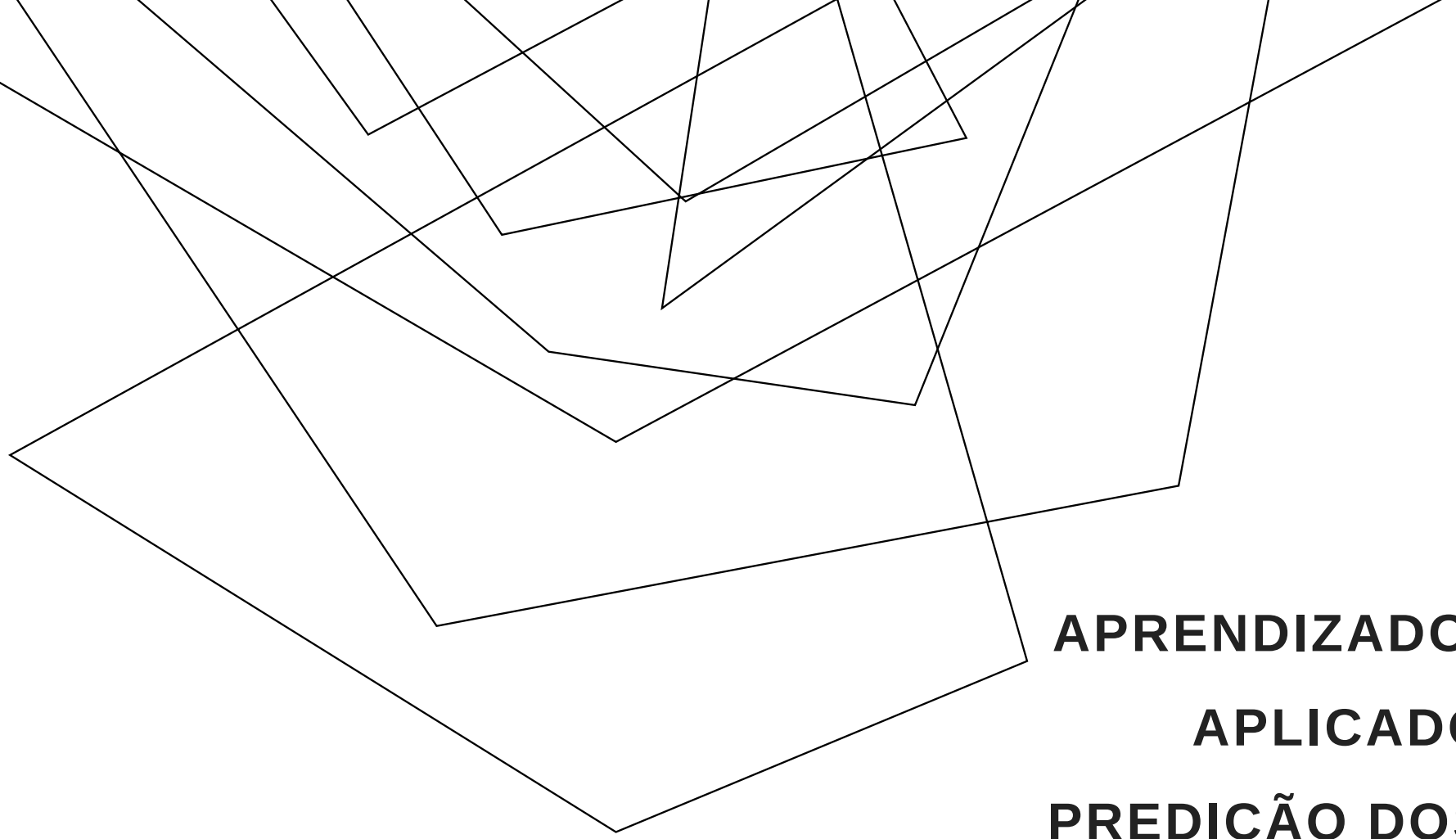


APRENDIZADO DE MAQUINA APLICADO PARA A PREDIÇÃO DOS LEADS MAIS PROPENSOS À COMPRA

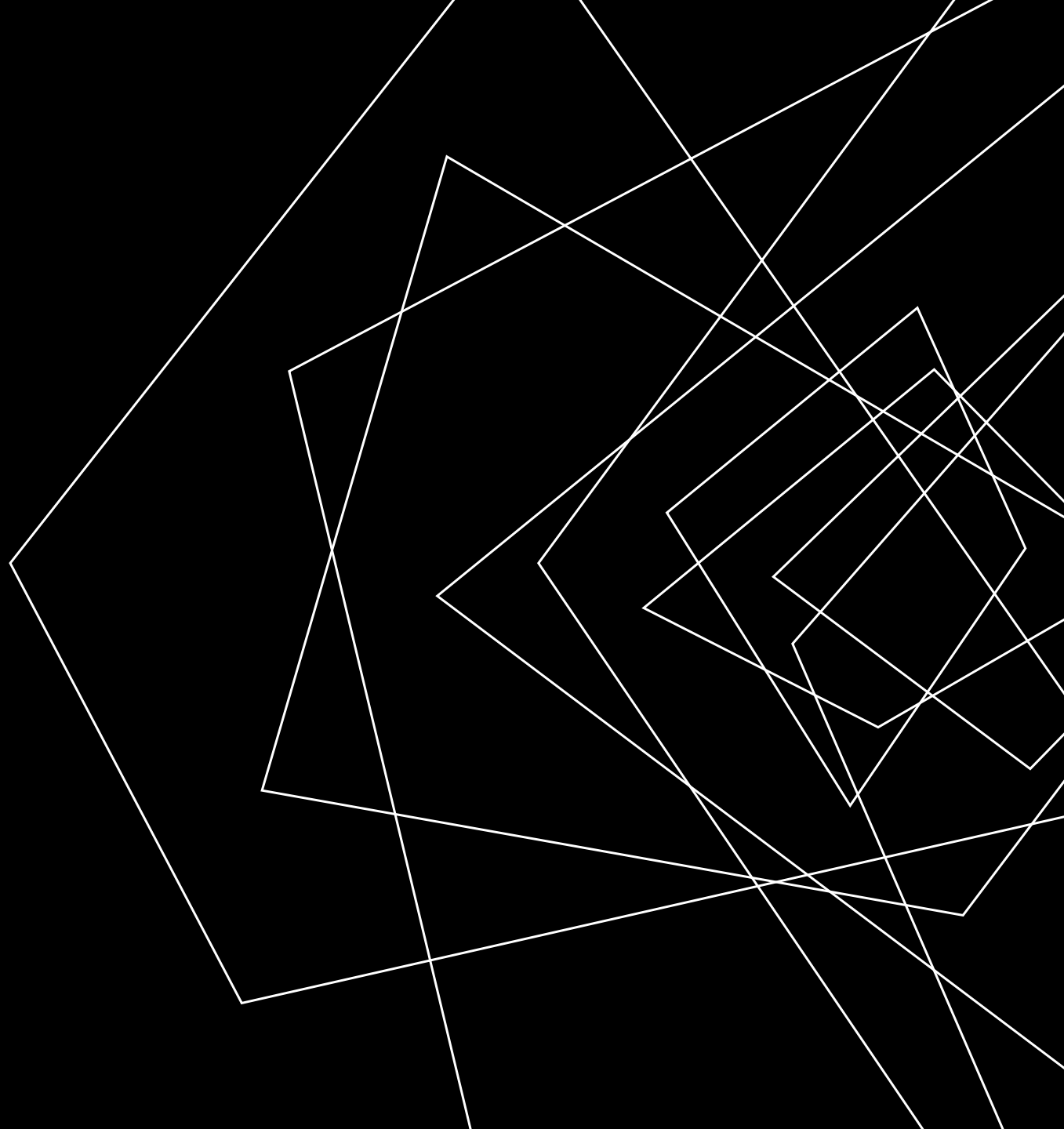
Adriano Fonseca

Belo Horizonte

2022

DEFINIÇÃO DO PROBLEMA

Propor uma abordagem preditiva para identificar a qual classe um Lead pertence, com o objetivo de prever se o Lead estaria ou não interessado na compra de planos de internet. Um Lead é considerado como uma oportunidade de negócio para a organização, é alguém que já demostrou interesse no produto.



POR QUE? — Com a solução, os times de vendas, planejamento e marketing esperam conseguir priorizar os Leads com maior interesse nos planos de internet.

QUEM? — Os dados analisados são de uma empresa privada.

O QUE? — O objetivo é analisar o comportamento preditivo dos *Leads* através das suas características de recargas e serviços enriquecidas.

ONDE? — O estudo é realizado com *Leads* espalhados em todo território brasileiro.

QUANDO? — Podemos considerar o terceiro e quarto trimestre do ano 2021.

5 W's[1]: why (por que), who (quem), what (o que), when (quando)

COLETA DE DADOS

O conjunto de dados utilizado neste projeto, foi obtido via Python e linguagem Structured Query Language (SQL), através de conexão no banco de dados MySQL, carregando os Leads da base de dados do produto chat da minha organização.

```
def get_contact(contact):
    # Get contact
    query = """
        SELECT DISTINCT
            l.id_contact,
            l.plan_type
        FROM
            lead.leads AS l
        WHERE l.is_chat = 1
            AND l.id_campaign = 25
            AND l.id_contact = {}
            AND l.created_at BETWEEN '2021-08-19 00:00:00' AND '2021-10-31 23:59:59';
    """.format(contact)
    data = pd.read_sql_query(query, db.get_connection('con_mysql'))
    return data
```

PROCESSAMENTO E TRATAMENTO DOS DADOS

Estatísticas do conjunto de dados

Número de variáveis	47
Número de observações	53257
Células ausentes	1474378
Células ausentes (%)	58,9%
Linhas duplicadas	0
Linhas duplicadas (%)	0.0%

Tipos de variáveis

Numérico	45
Categórico	2

Variável alvo venda

Value	Count	Frequency (%)
0	40490	76.0%
1	12767	24.0%

PROCESSAMENTO E TRATAMENTO DOS DADOS

Os valores ausentes, são referentes as variáveis enriquecidas de recargas e serviços e não é descartado inconsistências na imputação de dados, no enriquecimento dos dados, visto que com êxito, valores zerados para as variáveis de recargas e serviços foram recuperados na base original.

recharge_frequency	rec_online_10	rec_online_35_b5	rec_online_15	...	prezao_mensal	prezao_quinzenal	prezao_semanal	recarga_sos	servicos_operadora
5.0	3.0	0.0	0.0	...	0.0	0.0	0.0	0.0	6.0
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
4.0	0.0	0.0	1.0	...	0.0	0.0	2.0	0.0	0.0
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN

PROCESSAMENTO E TRATAMENTO DOS DADOS

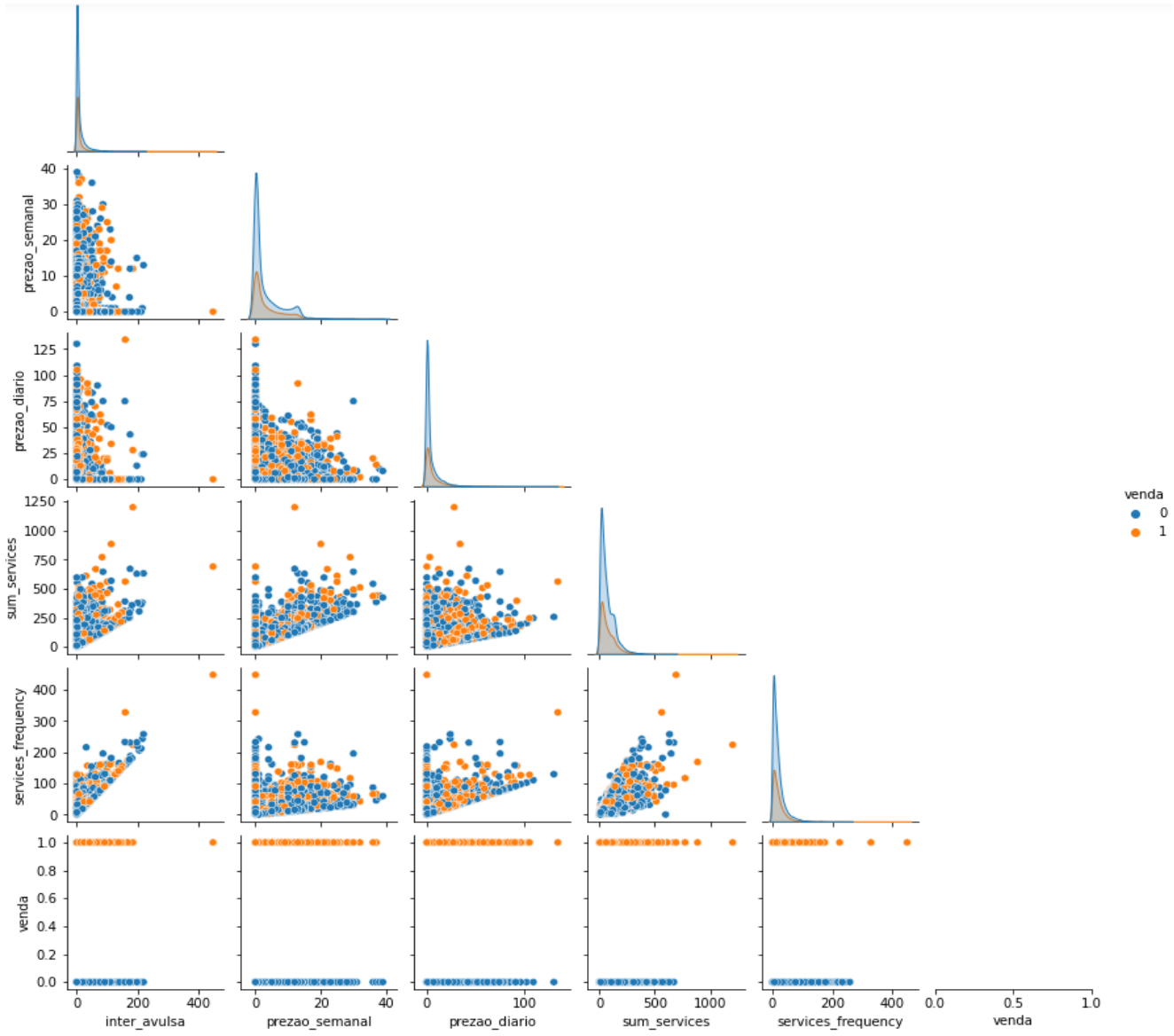
```
data.select_dtypes(include='object')
```

	regional	plan_type
0	BASE	CONTROLE
13	SP	PRE PAGO
14	RJES	PRE PAGO
17	SP	CONTROLE
19	RJES	PRE PAGO
...	...	...
53242	SP	PRE PAGO
53250	NE	PRE PAGO
53253	SP	CONTROLE
53254	SP	PRE PAGO
53256	SP	PRE PAGO

16376 rows × 2 columns

```
# Transformar rótulos não numéricos (desde que sejam laváveis e comparáveis) em rótulos numéricos.
var_cat=data.select_dtypes('object')
for col in var_cat:
    data[col] = LabelEncoder().fit_transform(data[col].astype('str'))
```

ANÁLISE E EXPLORAÇÃO DOS DADOS

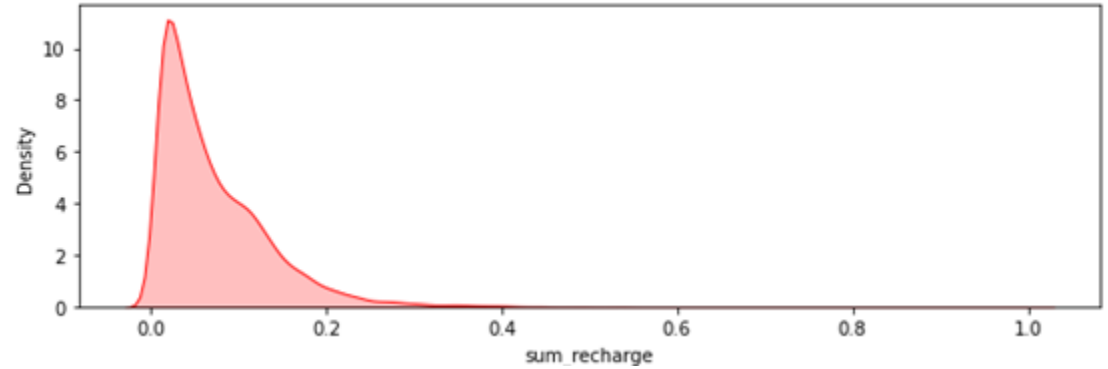


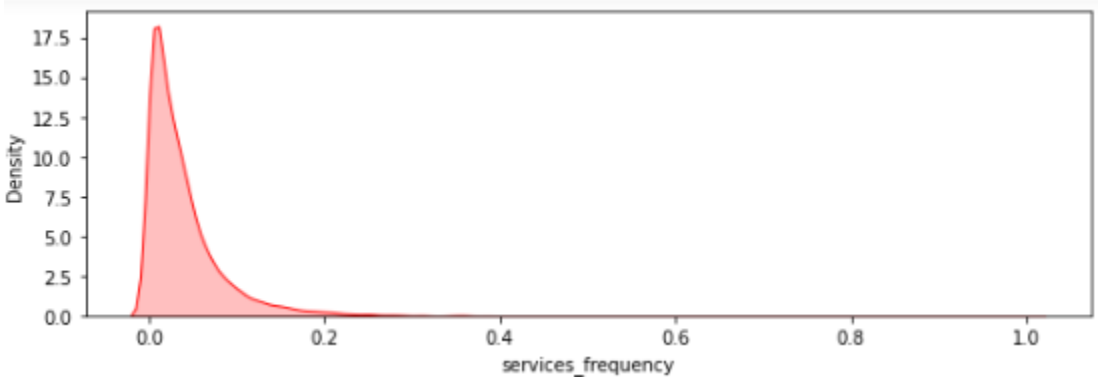
ANÁLISE E EXPLORAÇÃO DOS DADOS

Para tratar as faixas de valores diferentes e reduzir a influência dos pesos dos coeficientes, vamos trabalhar melhor os atributos, realizando a normalização *Min-Max* dos dados das variáveis de recargas e serviços.

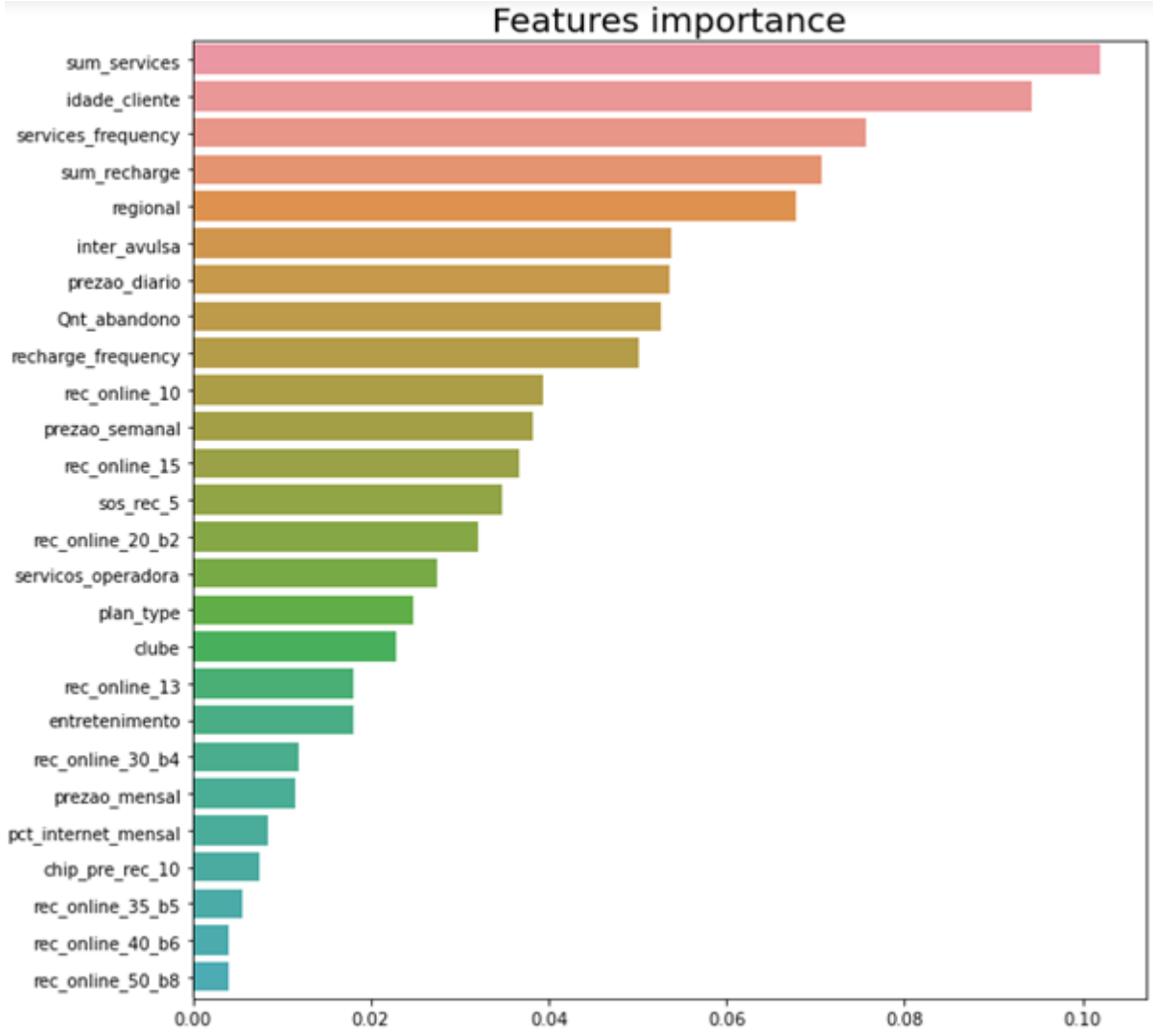
```
# Retirando a alta dimensionalidade
# Normalização Min-Max dos dados
cols = ['regional', 'idade_cliente', 'plan_type', 'Qnt_abandono',
        'sum_recharge', 'recharge_frequency', 'rec_online_10',
        'rec_online_35_b5', 'rec_online_15', 'sos_rec_5', 'rec_online_20_b2',
        'chip_pre_rec_10', 'chip_pre_rec_20', 'rec_online_13',
        'rec_online_50_b8', 'rec_online_30_b4', 'rec_online_40_b6',
        'pct_rec_1190', 'pct_rec_690', 'rec_online_100_b18', 'pct_rec_sos_5',
        'sos_rec_3', 'rec_online_8', 'sum_services', 'services_frequency',
        'inter_avulsa', 'antivirus', 'app_educacao', 'app_emprego', 'app_saude',
        'clube', 'pre_mix_giga', 'entretenimento', 'games',
        'pct_internet_mensal', 'prezao_diario', 'prezao_mensal',
        'prezao_quinzenal', 'prezao_semanal', 'recarga_sos',
        'servicos_operadora', 'sms_cobrar', 'sms_internacional',
        'transf_entre_regionais', 'truecaller']
for col in cols:
    # Ajustar aos dados e transformá-los.
    data[col] = MinMaxScaler().fit_transform(data[col].values.reshape(-1,1))
```

ANÁLISE E EXPLORAÇÃO DOS DADOS

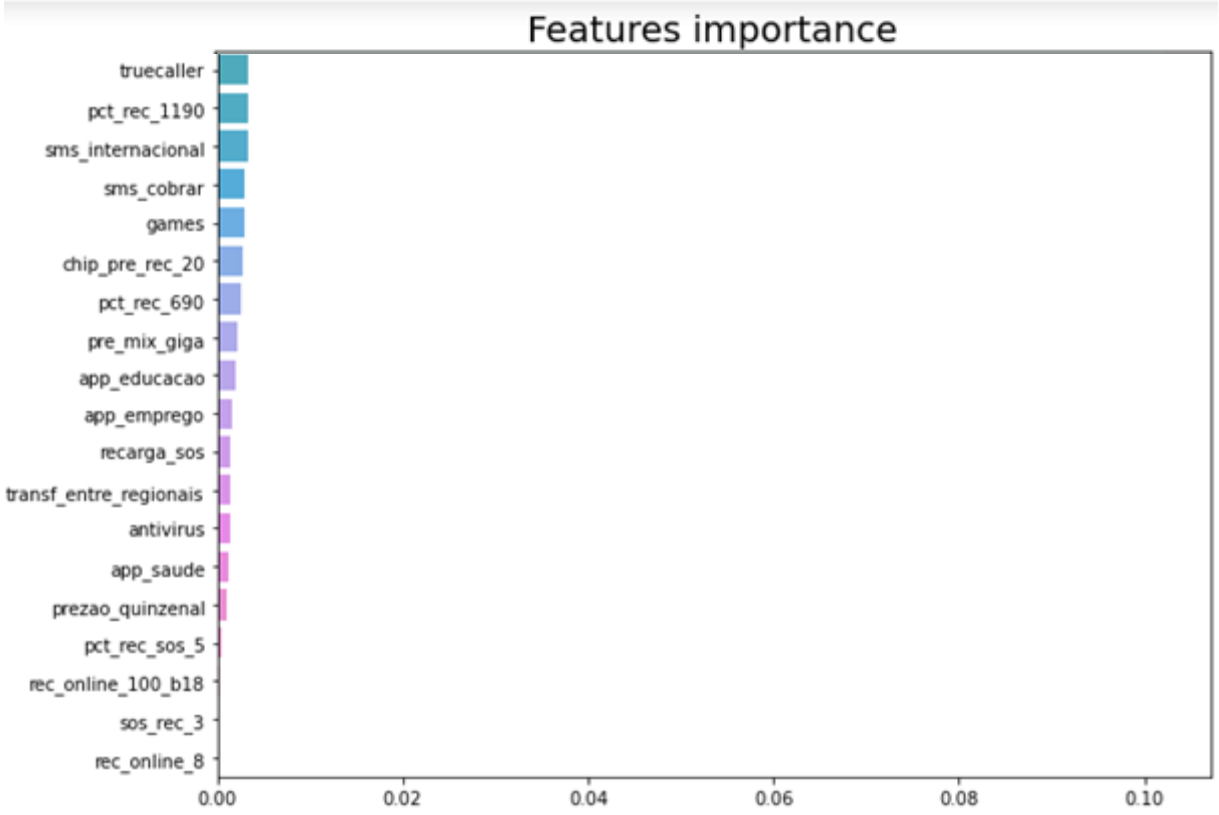




ANÁLISE E EXPLORAÇÃO DOS DADOS



ANÁLISE E EXPLORAÇÃO DOS DADOS




```
# Drop variáveis
to_drop=['rec_online_8', 'sos_rec_3', 'rec_online_100_b18', 'pct_rec_sos_5',
         'prezao_quinzenal', 'antivirus', 'app_saude', 'app_emprego', 'recarga_sos',
         'transf_entre_regionais', 'pre_mix_giga', 'app_educacao', 'pct_rec_690',
         'chip_pre_rec_20', 'games', 'sms_cobrar', 'pct_rec_1190','truecaller',
         'sms_internacional']
data.drop(to_drop, axis=1, inplace=True)
```

RESULTADO DO AJUSTE DO MODELO ML

Ajuste do modelos Suport Vector Machine (svm):

```
Fitting 5 folds for each of 64 candidates, totalling 320 fits
The best parameters for using this model SVM is {'C': 100, 'gamma': 1, 'kernel': 'sigmoid'}
```

Ajuste do modelos *Gaussian Naive Bayes* (gnb):

```
Fitting 5 folds for each of 100 candidates, totalling 500 fits
The best parameters for using this model GNB is {'var_smoothing': 5.3366992312063123e-05}
```

Ajuste do modelos RandomForestClassifier (rfc):

```
Fitting 5 folds for each of 2880 candidates, totalling 14400 fits
The best parameters for using this model RFC is {'class_weight': 'balanced_subs
ample', 'criterion': 'entropy', 'max_depth': 5, 'max_features': 'log2', 'min_sa
mples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 200}
```

```
clfs={

    'svm_best_param': SVC(C=100, gamma=1, kernel='sigmoid', probability=True, random_state=seed),

    'gnb_best_param': GaussianNB(var_smoothing=5.3366992312063123e-05),

    'rfc_best_param': RandomForestClassifier(class_weight='balanced_subsample', criterion='entropy',
                                             max_depth=5, max_features='log2', min_samples_leaf=2,
                                             min_samples_split=5, n_estimators=200, random_state=seed,
                                             n_jobs=-1),


}
```

INTERPRETAÇÃO DOS RESULTADOS

```
{'svm_best_param': 0.636030534351145,
 'gnb_best_param': 0.6665648854961832,
 'rfc_best_param': 0.6516030534351145}
```

Figura 125 – Resultado da acurácia dos modelos

```
{'svm_best_param': 0.37901498929336186,
 'gnb_best_param': 0.43,
 'rfc_best_param': 0.4417055296469021}
```

Figura 126 – Resultado da precisão dos modelos

```
{'svm_best_param': 0.36645962732919257,
 'gnb_best_param': 0.40062111801242234,
 'rfc_best_param': 0.6863354037267081}
```

Figura 127 – Resultado da revocação dos modelos

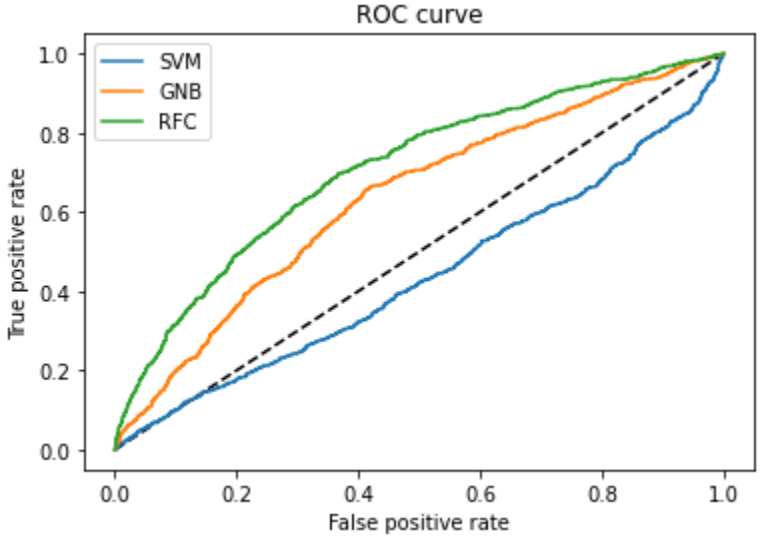
```
{'svm_best_param': 0.37263157894736837,
 'gnb_best_param': 0.41479099678456594,
 'rfc_best_param': 0.5374949331171464}
```

Figura 128 – Resultado da F1 score dos modelos

```
{'svm_best_param': 0.4351556202348,
 'gnb_best_param': 0.6349678591379309,
 'rfc_best_param': 0.7075544699963328}
```

Figura 129 – Resultado AUC dos modelos

INTERPRETAÇÃO DOS RESULTADOS



O valor ideal para AUC é 1 e AUC, , mas um bom classificador terá AUC acima dos 0.90% ou próximo e AUC para os modelos SVM está com valor 0.44, GNB com 0.63 e o RFC é de 0.71.

O RandomForestClassifier é o classificador que chega mais próximo de um bom classificador.

INTERPRETAÇÃO DOS RESULTADOS

- *RandomForestClassifier*, modelo que manteve o melhor resultado no *tuning* de parâmetros:

```
Confusion matrix:
[[1471  838]
 [ 303  663]]
```

Temos um suporte de 2309 não venda, onde o modelo *RandomForestClassifier* conseguiu prever corretamente 1471 não venda (TP) e os demais 838 previu como venda (FN). Para a venda, temos um suporte de 966, onde o modelo *RandomForestClassifier* conseguiu prever 303 como não venda (FP) e mais 663 como venda (TN).

INTERPRETAÇÃO DOS RESULTADOS

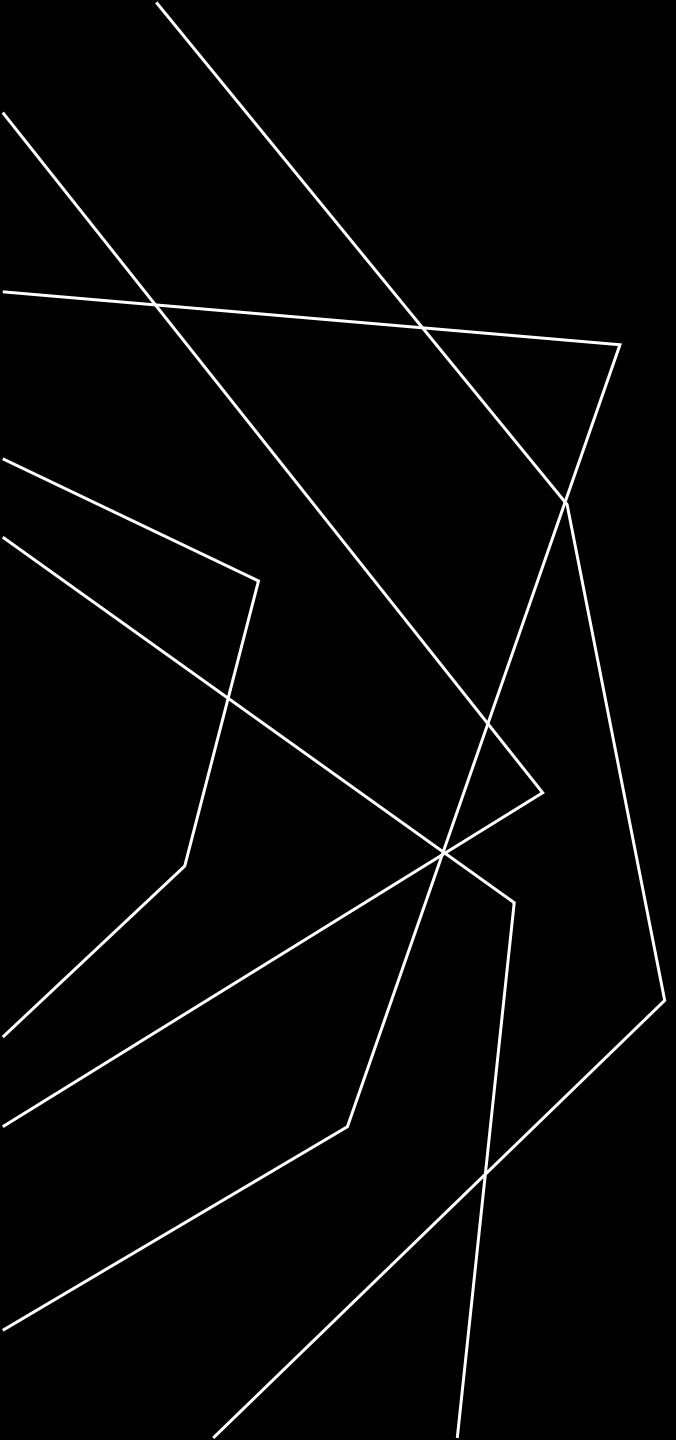
- *RandomForestClassifier*, modelo que manteve o melhor resultado no *tuning* de parâmetros:

```
Classification report:
              precision    recall  f1-score   support

           0       0.83      0.64      0.72      2309
           1       0.44      0.69      0.54       966

    accuracy                           0.65      3275
   macro avg       0.64      0.66      0.63      3275
weighted avg       0.71      0.65      0.67      3275
```

Para auxiliar no entendimento da matriz de confusão, podemos observar o *classification report*, que nos confirma toda interpretação acima, realizada sobre a predição do modelo *RandomForestClassifier* para as classes da variável venda, onde a não venda é (Classe sem fraude “0”) e venda é (Classe fraudulenta “1”).



OBRIGADO!