Máquinas de Vetores de Suporte e Regressão Logística aplicados ao problema de análise de crédito

Evelin Heringer Manoel Krulikovski*
Universidade Federal do Paraná

Novembro de 2016

Resumo

Neste trabalho, abordamos duas técnicas de aprendizagem de máquina supervisionada, as Máquinas de Vetores de Suporte (SVM, do Inglês *Support Vector Machine*) e Regressão Logística, para o problema de análise de crédito, que é um problema de classificação binária. Através destas técnicas, encontramos uma função de decisão, que prediz se um cliente é um "bom" ou "mau" pagador. O objetivo geral foi realizar um estudo prático sobre o SVM, pois queríamos verificar a fundamentação matemática na aplicação citada acima, a qual envolve um problema de otimização quadrático e convexo. Realizamos uma comparação entre dois tipos de funções Kernel: Polinomial (homogênea e não homogênea) e Gaussiana, usadas para resolver o problema quando não for possível encontrar uma função de decisão no espaço de entrada. Por fim, fizemos uso do método de Regressão Logística, para compararmos com os resultados obtidos pelo SVM. Os resultados encontrados demonstram uma superioridade do SVM em relação à outra técnica. A validação do modelo foi realizada por meio do método *cross-validation*, ou seja, a subdivisão da amostra original em duas partes: uma para a definição do modelo e outra para a sua validação.
Palavras-chaves: Análise de crédito. Support Vector Machine. Regressão Logística.

1 Introdução

O presente trabalho é resultado da aplicação das técnicas Support Vector Machine (SVM) e Regressão Logística ao problema de prever se determinado solicitante é um "bom" ou "mau" pagador, para fins de fornecimento de crédito.

*evehmano@gmail.com

As máquinas de Vetores de Suporte (SVM) constituem uma das técnicas de aprendizagem mais utilizadas. Muitas são as aplicações, como na classificação de caracteres, bioinformática e análise de imagens. Entretanto na literatura existem poucas aplicações desta na Análise de crédito. Já a Regressão Logística está presente em diversas aplicações para o problema de Análise de crédito.

2 Análise de crédito

A primeira manifestação histórica de concessão de crédito ocorreu no Sul da Alemanha em 1946 e foi denominada

Associação do pão

criada pelo Pastor Raiffeinsem após um rigoroso inverno que deixou agricultores locais endividados e na dependência de agiotas.

A partir de então, surgiram mundialmente inúmeras associações para fornecimento de crédito e a análise de crédito passou a ser uma das mais importantes atividades bancárias. Assim, precisando do desenvolvimento de modelos de tratamento do risco de crédito, procurando uma otimização deste problema.

A importância do crédito na atividade econômica tornou-se fundamental. Sem obter financiamento bancário as pequenas e médias empresas acabam parando a sua capacidade produtiva e comprometendo a geração de emprego e de rendimento. Isto provoca falência de muitas pequenas e médias empresas e outras nem chegam a concretizar-se. Um outro fator importante é a melhoria das condições de vida das famílias, podendo antecipar a obtenção de bens, como por exemplo casa, carro, estudos, etc.

O fornecimento de crédito acorda-se entre as partes interessadas (credor e devedor) à utilização de um determinado montante de dinheiro durante um período de tempo. A análise de crédito busca uma probabilidade de o tomador não cumprir com o pagamento de tal montante.

Um dos principais objetivos para as empresas fornecedoras de crédito é obter um modelo para medir o risco de inadimplência de seus futuros clientes, através do *credit score*.

O *credit score* consiste numa análise à qualidade de crédito, relacionando o incumprimento de empréstimos com as características dos clientes, permitindo a construção de um modelo onde as características contribuem para estimar a probabilidade final de inadimplência.

Dentre as vantagens da utilização da pontuação temos que ela pode ser monitorada ao longo do tempo e pessoas sem o conhecimento técnico em estatística ou economia pode utilizá-la facilmente para tomar decisões.

Num estudo realizado por Fazenda (2008) concluiu-se que na análise de incumprimento, as características pessoais não são tão importantes quanto as características dos créditos concedidos.

3 Técnicas de aprendizagem supervisionada

A aprendizagem de máquina busca produzir ferramentas e métodos voltados para a compreensão de dados. Fizemos uso dela para o problema de análise de crédito. Assim, comparamos a capacidade preditiva das técnicas de aprendizagem de máquina: Support Vector Machine e Regressão Logística.

A aprendizagem da máquina é uma das áreas que têm fornecido técnicas úteis para prever eventos de ganho ou perca. A qual obtém-se conclusões genéricas a partir de um conjunto particular de exemplos.

O aprendizado da máquina pode ser comparado ao dos seres humanos, no sentido que ele vem com a experiência. Ela precisa experimentar o problema, várias vezes, e criar seu próprio modelo da solução. Essencialmente, isso significa reconhecer padrões. Quanto mais informação for fornecida, ou seja, quanto mais experiência obter, melhor vai ser o aprendizado, pois será mais fácil de reconhecer o padrão.

Por exemplo, dada uma sequência, temos que dizer o próximo número. Se for uma sequência de um ou dois números, por exemplo $(3, 6, \ldots)$, dificilmente saberemos qual é a regra que gera aquela sequência e assim conseguir obter o próximo valor. Mas, quanto mais números termos da sequência, mais fácil é de encontrar o padrão, como $(3, 6, 9, 12, \ldots)$, onde o padrão é uma P.A. de razão 3.

Este é um tipo de aprendizado, o supervisionado, onde temos alguns casos solucionados e a partir disso podemos deduzir o caso desconhecido. Foi este tipo de aprendizado que abordamos aqui.

Esse tipo é muito usado, por exemplo, para problemas como reconhecimento facial. Vamos supor que queremos que a máquina reconheça um professor. No caso do aprendizado supervisionado, o que temos que fazer é alimentar a máquina, com várias fotos do professor e colocar um rótulo nessas imagens, que na prática significa dizer para ela: essas imagens são o professor.

O que vai ser feito é uma coleta de características, de cada uma daquelas fotos, para criar um padrão, um modelo de características do rosto do professor. Feito isso, toda vez que for apresentado uma nova imagem e perguntar quem é essa pessoa, basta comparar o modelo de características que ela criou a partir dos outros dados, com as características da nova imagem e se as características coincidirem significa que provavelmente essa nova imagem é o professor.

Os métodos de aprendizagem tradicionalmente empregados para tratamento do tipo de problema que propomos, a análise de crédito, atuam de forma

a modelar um conjunto de variáveis (características) de forma tal que essas ofertem uma saída - como uma previsão - para uma nova entrada.

Assim, dado um conjunto de exemplos rotulados na forma (x^i, y_i), em que x^i representa um exemplo e y_i um rótulo, deve-se produzir um classificador capaz de prever o rótulo para novos dados. Essa fase de obtenção do classificador a partir de uma amostra de dados é denominada fase de treinamento. E para estimar a taxa de acerto ou acurácia, temos a fase de teste, que dado um conjunto com saídas conhecidas, testamos se o classificador obtido na fase anterior realmente fornece a saída igual a conhecida.

Os rótulos ou classes representam as possíveis respostas para o dado problema. Neste trabalho, considera-se uma classificação binária, isto é, o dado x pertence ao conjunto (X_1), dos dados da classe positiva ou pertence ao conjunto (X_2), dos pontos da classe negativa. Por exemplo, é ou não uma imagem do professor respectivamente. Em que, os rótulos assumem valores discretos -1 $(x \in X_2)$ ou 1 $(x \in X_1)$ para a técnica SVM e valores discretos 0 $(x \in X_2)$ e 1 $(x \in X_1)$ para a técnica de Regressão Logística. Cada exemplo, também referenciado por dado ou caso, será representado por um vetor de características. A qual cada característica, pode ser nominal (por exemplo, estado civil) e contínuo (por exemplo, idade).

3.1 Support Vector Machine

No problema proposto, usamos o chamado Truque do Kernel, pois não foi possível separar perfeitamente o conjunto de dados em duas classes (classe dos maus e a dos bons pagadores) por um hiperplano de máxima margem.

Dentre as principais vantagens do SVM para o problema de análise de crédito, destacam-se o bom desempenho de generalização dos dados, boa fundamentação teórica e a existência de poucos parâmetros livres para ajuste.

A seguir, abordaremos brevemente somente dois tipos de classificador: o linear com margem flexível e o não-linear respectivamente. O linear/margem rígida, não iremos tratar pois está restrito à poucas aplicações práticas.

Para os dois tipos de classificador, iremos considerar o conjunto de treinamento

$$S = \{(x^1, y_1), \ldots, (x^m, y_m), x^i \in \mathbb{R}^n, y_i \in \{-1, 1\}\}.$$

O objetivo da técnica SVM, é obter um hiperplano separador

$$w^T x + b = 0,$$

tal que, dado um novo caso x diremos que,

$$x \text{ é da classe positiva, se } w^T x + b > 0 \text{ e}$$

$$x \text{ é classe negativa, se } w^T x + b < 0$$

Além disso, precisamos obter uma máxima margem, acrescentando as seguintes restrições

$$w^T x + b \geq 1 \text{ para todo } x \in X_1,$$

$$w^T x + b \leq -1 \text{ para todo } x \in X_2$$

ou de forma compacta

$$y(w^T x + b) \geq 1 \text{ para todo } x \in X_1 \cup X_2.$$

3.1.1 C-SVM

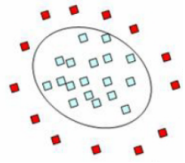
É comum aplicações que os m dados não sejam linearmente separáveis. Quando isso ocorre, mas não de forma severa, introduz-se m variáveis de folga ($\xi_i \geq 0$), para penalizar a função objetivo e dar uma folga as restrições. Portanto, temos o seguinte problema

$$\begin{aligned} \min_{(w,b,\xi)} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \\ s.a \quad & y_i(w^T x^i + b) \geq 1 - \xi_i,\ i = 1, \ldots, m \\ & \xi_i \geq 0,\ i = 1, \ldots, m \end{aligned} \tag{1}$$

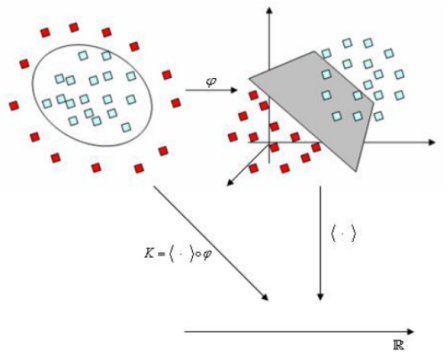
com $C > 0$ uma constante de penalização, definida pelo usuário. Esta técnica também é conhecida como SVM com margens flexíveis.

3.1.2 C-SVM com Kernel

Existem muitos casos, nos quais o conjunto de dados está "longe" de ser linearmente separável. Por exemplo,



A ideia para resolver este problema é conhecida como Truque do Kernel, que consiste de mapear os pontos para um espaço de dimensão ($\mathbb{R}^N$), maior do que o espaço de entrada ($\mathbb{R}^n$), para obter um hiperplano separador em tal espaço e depois retornar ao espaço de entrada, conforme podemos acompanhar na figura a seguir



Logo, o problema anterior torna-se

$$\begin{aligned} \min_{(w,b,\xi)} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \\ s.a \quad & y_i(w^T\Phi(x^i)+b) \geq 1-\xi_i,\ i=1,\ldots,m \\ & \xi_i \geq 0,\ i=1,\ldots,m \end{aligned} \tag{2}$$

onde $\Phi : \mathbb{R}^n \to \mathbb{R}^N$ é a função mapeamento.

Entretanto, iremos usar a teoria do Lagrangiano para obter a forma dual do SVM, que fornece um problema com restrições mais simples, obtendo o seguinte problema dual,

$$\begin{aligned} \max_{\alpha} \quad & -\frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j \Phi(x^i)^T\Phi(x^j) + \sum_{i=1}^{m}\alpha_i \\ s.a \quad & 0 \leq \alpha_i \leq C,\ i=1,\ldots,m \\ & \sum_{i=1}^{m}\alpha_i y_i = 0 \end{aligned} \tag{3}$$

Através do Truque do Kernel, podemos substituir o produto interno $(x^i)^T x^j$ por uma função Kernel ($K(x^i,x^j) = \Phi(x^i)^T\Phi(x^j)$), obedecendo o Teorema de Mercer, para resolvermos problemas não linearmente separáveis.

Levando em consideração que $\max f(x)$ é equivalente a $\min -f(x)$, resolveremos o seguinte problema

$$\begin{aligned} \min_{\alpha} \quad & -\sum_{i=1}^{m}\alpha_i + \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j K(x^i,x^j) \\ s.a \quad & 0 \leq \alpha_i \leq C,\ i=1,\ldots,m \\ & \sum_{i=1}^{m}\alpha_i y_i = 0 \end{aligned}$$

o que equivale a

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha^T H\alpha - e^T\alpha \\ s.a \quad & y^T\alpha = 0 \\ & 0 \leq \alpha \leq C \end{aligned}$$

com $H(i,j) = y_i y_j K(x^i, x^j)$ e $e_j = 1, j = 1, \ldots, m$

3.2 Regressão Logística

O modelo de regressão logística é uma das técnicas mais utilizadas para a área de análise de crédito. Sua diferença da técnica SVM está na falta das restrições, ou seja, não precisamos impor uma máxima margem.

Tal modelo consiste em estimar a probabilidade de ocorrência de um evento com base em um conjunto de características, ou seja, a probabilidade de um dado pertencer a uma determinada classe.

Como por exemplo, a classe dos clientes x adimplentes, pertencentes ao conjunto X_1 ou a classe dos clientes x inadimplentes, pertencentes ao conjunto X_2. Consideremos a variável dependente ou rótulo $y \in \{1, 0\}$, onde:

$y = 1$ se x pertence a "classe positiva", é adimplente, e

$y = 0$ se x pertence a "classe negativa", é inadimplente

Então o modelo de regressão logística consiste em tomar a seguinte função, que calcula a probabilidade de termos $y = 1$, considerando que x ocorreu e utilizando o parâmetro θ.

$$P_a = g(\theta^T x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n)}}$$

Em que P_a representa a probabilidade de um dado assumir um certo valor (neste caso de uma cliente ser adimplente, isto é, que dado um novo cliente x, verificar se admite $y = 1$). Em que, $\theta_1, \ldots, \theta_n$ representam os coeficientes das características $x_1, \ldots, x_n$, com $x = (1, x_1, \ldots, x_n)$.

Para a função acima, prevemos que $y = 1$ se $\theta^T x \geq 0$ e $y = 0$ se $\theta^T x < 0$, logo, o hiperplano nesse caso será $\theta^T x = 0$. Observe que,

$z = \infty$ temos $e^{-z} \to 0$ portanto $P(y) = 1$

$z = -\infty$ temos $e^{-z} \to \infty$ portanto $P(y) = 0$

Estaremos considerando para o modelo de regressão logística, isto é, para podermos encontrar o melhor parâmetro θ, a minimização da seguinte função

$$f(\theta) = -\sum_{i=1}^{n} [y_i log(m_\theta(x^i)) + (1 - y_i) log(1 - m_\theta(x^i))]$$

4 Implementação Computacional

4.1 Proposta

A presente seção tem como objetivo principal apresentar e analisar os dados obtidos na pesquisa. Analisa-se a caracterização da amostra, das variáveis, o modelo de Regressão Logística e do SVM, o modelo estimado e discussão dos resultados para decisão final do fornecimento de crédito.

4.2 Banco de dados

Usamos os dados da base German Credit Dataset, disponível em [1]. Realizamos testes comparando as técnicas: Regressão Logística e SVM. Os resultados foram comparados com base na acurácia (medida pelo método ACC).

A base de dados é composta de mil dados, dos quais 30% são classificados como maus pagadores e o restante como bons pagadores. A base traz um conjunto de 20 características e uma variável predita (1==Bom e 2==Mau). Deste conjunto um total de sete características são do tipo contínuas (C), cinco características ordinais (O), seis características nominais (N) e três características binárias (B) (incluindo a variável predita).

Variável	**Descrição**	**Tipo**	**#**
Checking	Status da conta do solicitante (salário declarado)	O	4
Duration	Duração do empréstimo requerido	C	-
history	Histórico de créditos anteriores	O	5
purpose	Finalidade do crédito requerido	N	11
amount	Montante do crédito requerido	C	-
savings	Economias do cliente	O	5
emplory	Tempo no emprego atual	O	5
rates	Juros praticados (%) no empréstimo requerido	C	-
status	Sexo e estado civil	N	5
debtors	Outras dívidas	N	3
residence	Tempo na residência atual	C	-
property	Propriedades	N	4
Age	Idade	C	-
other inst	Outros planos de parcelamento	N	3
housing	Tipo de moradia	N	3
exist cr	Nº de créditos concedidos no banco	C	-
Job	Profissão	O	4
provider	Nº de dependentes	C	-
phone	Telefone próprio	B	2
foreign	Estrangeiro	B	2
goodbad	Bom ou mau pagador	B	2

Tabela 1: Variáveis presentes na base de dados

Para o banco de dados *german*, temos as seguintes descrições,

1. Título: Banco de dados German;
2. Fonte de informação: Prof. Dr. Hans Hofmann Institut f"ur Statistik und "Okonometrie Universit"at Hamburg FB Wirtschaftswissenschaften Von-Melle-Park 5 2000 Hamburg 13
3. Número de dados: 1000

   Dois conjuntos de dados são fornecidos, o conjunto de dados originais, na forma prevista pelo Prof Hofmann, contém atributos categóricos/simbólicos e está no arquivo *german.data*.

A11	6	A34	A43	1169	A65	A75	4	A93	...	1	A192	A201	1
A12	48	A32	A43	5951	A61	A73	2	A92	...	1	A191	A201	2
A14	12	A34	A46	2096	A61	A74	2	A93	...	2	A191	A201	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Tabela 2: german.data

   No nosso caso, precisamos de atributos numéricos, a Universidade de Strathclyde produziu o arquivo *german.data-numérico*. Este arquivo foi editado e diversas variáveis adicionadas para torná-lo adequado para algoritmos que não conseguem lidar com as variáveis categóricas.

4. Número de atributos german: 20 (7 numéricos e 13 categóricos).

   Número de atributos german.numer: 24 numéricos;

   Entretanto, iremos considerar um arquivo *dados.m*, que editamos em Matlab, composto por apenas 20 variáveis e a variável preditiva.

1	6	4	3	1169	5	5	4	3	...	2	1	1
4	12	4	6	2096	1	4	2	3	...	1	1	1
1	42	2	2	7882	1	4	2	3	...	1	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Tabela 3: dados.m

5. Descrição dos atributos:
   (a) Atributo 1: (qualitativo) Status do saldo da conta do solicitante:
      - A11: menor do que 0;
      - A12: entre 0 e 200;
      - A13: maior ou igual a 200;
      - A14: não possui conta;

   (b) Atributo 2: (numérico) Duração da conta em mês;

(c) Atributo 3: (qualitativo) Histórico de crédito

- A30: não tomou crédito/todos os crédito tomados pagos;
- A31: todos os créditos disponíveis neste banco pagos devidamente;
- A32: créditos existentes pago devidamente até agora;
- A33: atraso no pagamento no passado;
- A34: relato crítico / outros créditos existentes (não neste banco)

(d) Atributo 4: (qualitativo) Finalidade do crédito requerido

- A40: carro (novo);
- A41: carro (usado);
- A42: móveis/equipamentos;
- A43: rádio/televisão;
- A44: eletrodomésticos;
- A45: reparos;
- A46: educação;
- A47: férias;
- A48: reciclagem;
- A49: negócios;
- A410: outros;

(e) Atributo 5: (numérico) Montante do crédito requerido

(f) Atributo 6: (qualitativo) Economias do cliente

- A61: menor do que 100;
- A62: entre 100 e 500;
- A63: entre 500 e 1000;
- A64: mais do que 1000;
- A65: Não consta;

(g) Atributo 7: (qualitativo) Tempo no emprego atual

- A71: desempregado;
- A72: menos do que 1 ano;
- A73: entre 1 e 4 anos;
- A74: entre 4 e 7 anos;
- A75: mais do que 7 anos;

(h) Atributo 8: (numérico) Taxa de juros praticados no empréstimo requerido;

(i) Atributo 9: (qualitativo) Sexo e estado civil

- A91: masculino e divorciado
- A92: feminino e divorciada ou casada
- A93: masculino e solteiro

- A94: masculino e casado
- A95: feminino e solteira

(j) Atributo 10: (qualitativo) Outras dívidas

- A101: nenhuma
- A102: co-aplicação;
- A103: fiador;

(k) Atributo 11: (numérico) Tempo na residência atual

(l) Atributo 12: (qualitativo) Propriedades

- A121: imobiliária
- A122: se não A121 - seguro de vida
- A123: se não A121 e A122 - carros ou outros, excluindo os do atributo 6.
- Sem propriedades

(m) Atributo 13: (numérico) Idade em anos

(n) Atributo 14: (qualitativo) Outro planos de parcelamento

- Banco
- Lojas
- nenhum

(o) Atributo 15: (qualitativo) Tipo de moradia

- alugado
- próprio
- parentes

(p) Atributo 16: (numérico) Número de créditos concedidos no banco

(q) Atributo 17: (qualitativo) Profissão

- desempregado
- autônomos
- empregado
- empresário

(r) Atributo 18: (numérico) Número de dependentes

(s) Atributo 19: (qualitativo) Telefone próprio

- nenhum
- sim, registrado no nome do cliente

(t) Atributo 20: (qualitativo) Estrangeiro

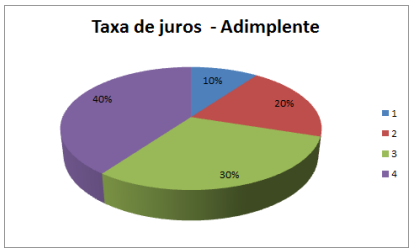
- sim
- não

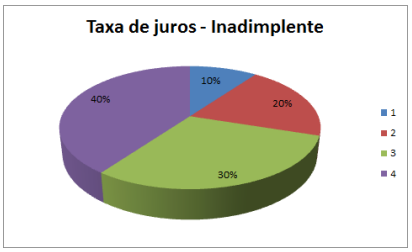
Os dados como citado acima, são compostos por 300 empréstimos “ruins” (por exemplo, ausência de pagamento ou atraso) e 700 empréstimos “bons” (por exemplo, pagamentos sem atraso). O objetivo foi obter uma função ou regra de decisão para futuros empréstimos com base nos dados fornecidos.

Na análise de crédito geralmente classifica-se os clientes “ruins” como aqueles que em algum período de tempo apresentaram inadimplência, por exemplo 60 dias ou mais.

A partir das variáveis descritas, analisamos sua influência na tomada de decisão. Respondendo a questões, como:

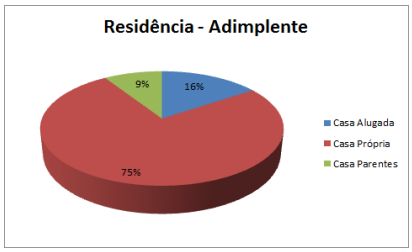
- Para a inadimplência, influência a taxa de juros empregada no empréstimo?

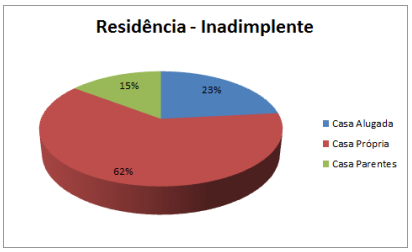




Como podemos acompanhar nos gráficos Taxa de juros, não existe diferença entre inadimplentes e adimplentes quanto a taxa de juros empregada. Dito de outra forma, no banco de dados German, a mesma porcentagem de pessoas inadimplentes ou adimplentes, optaram pela mesma taxa de juros.

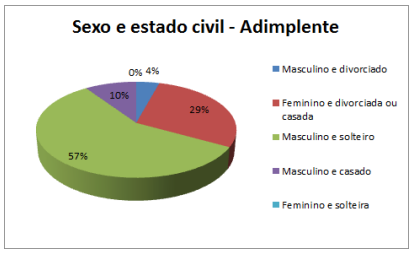
- Para a inadimplência, influência morar em casa alugada?

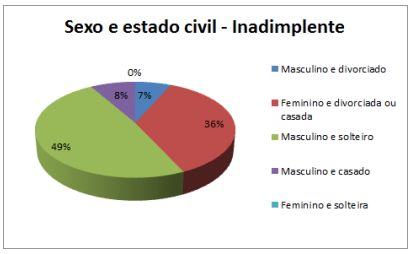




Percebemos pelos gráficos Residência, que pessoas que moram em casa alugada ou de parentes acabam tendo uma porcentagem maior de não cumprindo com o pagamento do crédito fornecido.

- Para a inadimplência, influência ser homem ou mulher? E o estado civil?



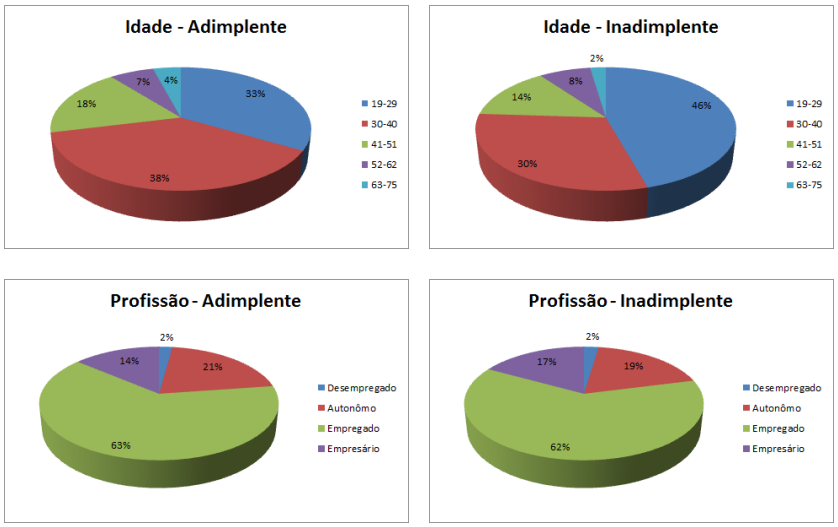


No nosso banco de dados, consta que temos 5 possíveis respostas para sexo e estado civil, entretanto não existe nenhum cliente solteiro e feminino, algo que precisariam revisar em tal banco de dados. Percebemos que na maioria dos casos homens solteiros fazem mais empréstimos, depois mulheres divorciadas ou casadas. Mas em relação a bons ou maus pagadores não percebemos muita diferença entre os sexos e estados civis.

- Para a inadimplência, influência a idade? Será que mais velhos são mais assíduos? Os jovens acabam precisando mais de empréstimo?

  Como podemos verificar nos gráficos Idade acima, quanto maior for a idade do cliente temos uma probabilidade maior de ser um bom pagador. Vemos também que acima dos 40 anos temos um decréscimo da procura por crédito.

- Para a inadimplência, influência a profissão?



Percebemos que não existe uma grande diferença em clientes bom ou maus pagadores, quanto a sua profissão, mas notamos que a maioria dos devedores (quem obtém o crédito) são empregados e depois autônomos.

Apresentamos nas figuras anteriores a análise da taxa de juros, do estatuto residencial, do sexo, estado civil, idade e profissão dos dados presentes em nosso banco de dados German.

4.2.1 Distribuição dos dados

O próximo passo foi separar os dados em dois grupos:

1. Dados para obter uma regra de decisão. (Treinamento);
2. Dados para teste. (Validação)

Separamos a base de dados da seguinte forma: 80% dos dados compondo a base de treinamento e 20% a base de validação. Mais adiante, fizemos uma mudança para verificar se acontece um aumento nos acertos dos modelos.

Construímos os modelos para a análise de crédito com o treinamento e em seguida avaliamos o ajuste na fase de teste. Analisamos se todas as características dos dados são necessárias, usando a análise de componentes principais para esse fim, onde o objetivo foi obter maior representatividade dos dados.

A Análise de Componentes Principais (PCA) é uma técnica estatística, bem utilizada para reduzir a dimensão dos vetores. Ela busca uma representação em dimensão menor de variáveis não correlacionadas.

O objetivo da PCA é obter maior representatividade com respeito à matriz de variância-covariância Σ e os autovalores obtidos da mesma. Se o autovalor

for grande, significa que esse fica há uma grande variância dos dados, isto é, tem grande relevância na construção dos modelos.

Onde a matriz de variância-covariância é uma matriz quadrada que contém as variâncias e covariâncias associadas a diversas variáveis. Os elementos diagonais da matriz contêm as variâncias das variáveis e os elementos fora da diagonal contêm as covariâncias entre todos os pares possíveis de variáveis.

Por exemplo, você cria uma matriz de variância-covariância para três variáveis, X, Y e Z. Na tabela a seguir, as variâncias são exibidas ao longo da diagonal a variância de X, Y e Z é 2,0, 3,4 e 0,82 respectivamente. A covariância entre X e Y é -0,86.

	X	Y	Z
X	2,0	-0,86	-0,15
Y	-0,86	3,4	0,48
Z	-0,15	0,48	0,82

Tabela 4: Matriz de variância-covariância

A matriz de variância-covariância é simétrica porque a covariância entre X e Y é a mesma covariância entre Y e X. Onde $var(Y) = E((y - E(Y))^2)$ e $cov(X, Y) = E(XY) - E(X)E(Y)$.

Para obtermos o valor da variância, usamos o comando do Matlab $var(x, y)$, para o cálculo da covariância usamos o comando $cov(x, y)$. Mas iremos calcular diretamente os coeficientes de correlação, usando o comando $R = corrcoef(TR)$, onde TR representa a matriz com os dados de treinamento, em que cada linha contém as informações dos clientes e as colunas representam as 20 características descritas na tabela acima.

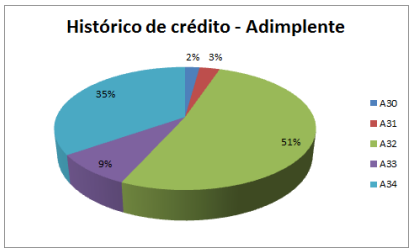
Então, usando o comando $d = eig(R)$, obtemos os seguintes autovalores,

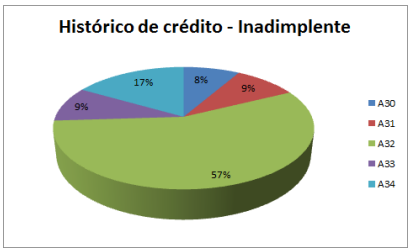
$$d = [2.5492; 1.9615; 0.2520; 1.4243; 1.3184; 0.4796; 0.4624; 1.2304; 1.1412; 1.1253;$$

$$1.0809; 0.5539; 0.6343; 0.6735; 0.9424; 0.7843; 0.8038; 0.8209; 0.8914; 0.8705];$$

Mostrando que a variável Histórico de crédito e Tempo no emprego atual, são de certa forma irrelevantes para os resultados, já que os autovalores correspondentes são pequenos (0.2520 e 0.4624) em relação aos demais.

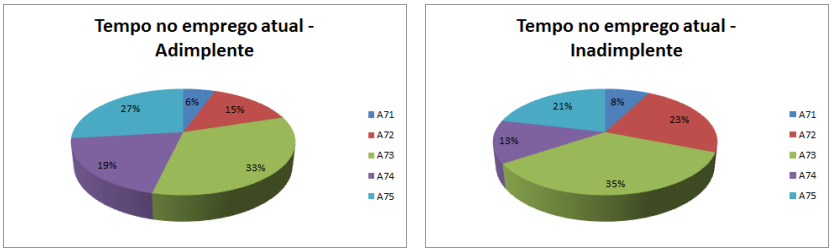
Podemos analisar esses resultados através dos seguintes gráficos





Vemos que para as variáveis A32 e A33, que podemos consultar sua descrição na Subseção 4.2, não existe diferença significativa entre bom ou mau pagador. Para A30 e A31, que induz que o cliente é um bom pagador, não influência, já que clientes com A30 e A31 foram mais inadimplentes do que adimplentes. Por fim, clientes com A34, induz que os clientes são maus pagadores, mas isso não influência no resultado, já que para os clientes adimplentes temos 35% da variável A34 e dos inadimplentes temos 17%.

Para o tempo no emprego atual temos os seguintes gráficos



Vemos que a maioria dos clientes maus pagadores é próximo em porcentagem da maioria dos clientes bons pagadores. Analisamos também que os clientes, com variáveis A71 e A72 aumentam em porcentagem dos bons para os maus pagadores, o que já esperava-se que ocorre-se por serem desempregados (mas não há uma diferença significativa) ou de serem trabalhadores de menos de 1 ano, aumentando quase 10% em diferença dos bons com os maus pagadores.

Assim, só poderemos desconsiderar a variável Histórico de crédito. Entretanto, não iremos fazer uso da PCA, pela dimensão dos dados estarem em $\mathbb{R}^{20}$, isto é, ter uma dimensão relativamente pequena para resolução do nosso problema e por apenas uma variável poder ser desconsiderada.

4.3 Resultados

A implementação desse trabalho foi feita utilizando o *software* MATLAB em sua versão R2011a. Os testes foram realizados em um computador portátil com processador Intel Core^{TM} i3-4010U com 3 MB de memória cache, com velocidade do clock de 1.7 GHz e com 4 GB de memória RAM, com sistema Operacional Windows 8.1 Single com arquitetura 64 bits.

4.3.1 Validação do Modelo

Para medirmos a acurácia dos resultados, pelo critério ACC, calculamos

$$ACC = \frac{T_p + T_n}{T_p + F_p + T_n + F_n}$$

que fornece a porcentagem de amostras positivas e negativas classificadas corretamente sobre a soma de amostras positivas e negativas. Em que T_p é o número

de classificações verdadeiras positivas, T_n é o número de classificações verdadeiras negativas, F_p é o número de classificações falsas positivas e F_n é o número de classificações falsas negativas.

Esses dados foram obtidos pela matriz de confusão, que mostra o número de classificações corretas (obtidas pelo modelo) versus as preditas (obtidas no conjunto de dados) para cada classe, sobre o conjunto de treinamento. O número de acertos, para cada classe, se localiza na diagonal principal e os demais elementos da matriz representam erros na classificação. Para um classificador ideal, todos os elementos fora da diagonal são zero, já que não comete erros.

Classe	Predita X_1	Predita X_2
Verdadeira X_1	T_p	F_n
Verdadeira X_2	F_p	T_n

Tabela 5: Matriz de confusão para 2 Classes

4.3.2 Support Vector Machine

Para o método SVM, foi utilizado o programa *quadprog* do MATLAB. Tal programa resolve o seguinte problema,

$$\begin{aligned} \min_{z} \quad & \frac{1}{2} z^T H z + f^T z \\ s.a \quad & Az \leq c \\ & Aeqz = ceq \\ & LB \leq z \leq UB \end{aligned}$$

usando o comando $[z, fval, exitflag] = quadprog(H, f, A, c, Aeq, ceq, LB, UB)$. As saídas $fval$ e $exitflag$ retornam o valor da função no ponto mínimo e as condições do problema, obtidas pelo programa respectivamente.

Implementação margem flexível

Para o problema primal, consideramos que alguns dados estejam classificados incorretamente. Assim, como vimos na Seção 3, temos o seguinte problema de otimização

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \xi_i \\ s.a \quad & y_i(w^T x^i + b) \geq 1 - \xi_i, \ i = 1, \ldots, m \\ & \xi_i \geq 0, \ i = 1, \ldots, m, \end{aligned}$$

com variáveis $w \in \mathbb{R}^n$, $b \in \mathbb{R}$ e $\xi \in \mathbb{R}^m$ e onde $C \in \mathbb{R}$ e $x^i, y_i, i = 1, \ldots, m$ são dados. Se definirmos $k = m + 1 + n$ e tomando a variável de otimização como

$$z = \begin{bmatrix} w \\ b \\ \xi \end{bmatrix} \in \mathbb{R}^k \text{ e } LB = \begin{bmatrix} -inf \\ \vdots \\ -inf \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^k, UB = \begin{bmatrix} inf \\ \vdots \\ inf \end{bmatrix} \in \mathbb{R}^k$$

e definindo as matrizes

$$H = \begin{bmatrix} I_{n\times n} & 0_{n\times(m+1)} \\ 0_{(m+1)\times n} & 0_{(m+1)\times(m+1)} \end{bmatrix} \in \mathbb{R}^{k\times k},\ f = \begin{bmatrix} 0_{n\times n} \\ 0 \\ C.e \end{bmatrix} \in \mathbb{R}^k$$

$$A = \begin{bmatrix} -diag(y)X & -y & -I_{m\times m} \end{bmatrix} \in \mathbb{R}^{m\times k},\ c = \begin{bmatrix} -e \end{bmatrix} \in \mathbb{R}^m$$

onde e é um vetor com todas as componentes iguais a 1, e X e y são definidos como

$$X = \begin{bmatrix} (x^1)^T \\ (x^2)^T \\ \vdots \\ (x^m)^T \end{bmatrix} \in \mathbb{R}^{m\times n},\ y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \in \mathbb{R}^m$$

Portanto, os comandos para resolver o problema recorrendo ao *quadprog* serão os do programa *SKCR.m* contido na Seção 6.

Para o problema dual, temos o seguinte problema de otimização

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha^T H\alpha - e^T\alpha \\ s.a \quad & y^T\alpha = 0 \\ & 0 \le \alpha \le C \end{aligned}$$

com $H(i,j) = y_i y_j (x^i)^T x^j$, $-f_j = e_j = 1, j = 1, \ldots, m$, $Aeq = y^T$, $ceq = 0$, $LB = 0$ e $UB = C$. Assim, usaremos o programa *quadprog* do MATLAB para resolver tal problema, fazendo uso do programa *SKCRD.m* contido na Seção 6.

SVM - Kernel com Regularização

Para o problema primal, precisamos conhecer a função Φ, que é um mapeamento do espaço de entrada no espaço $feature$, isto é, a cada $x^i \in \mathbb{R}^n$ associamos $\Phi(x^i) = z^i \in \mathbb{R}^N$, com $N > n$. Assim, considerando o primal com regularização, temos o seguinte problema de otimização

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\Phi(\xi_i) \\ s.a \quad & y_i(w^T\Phi(x^i) + b) \ge 1 - \xi_i,\ i = 1, \ldots, m \\ & \xi_i \ge 0,\ i = 1, \ldots, m \end{aligned}$$

Caso tenhamos conhecimento da função Φ, podemos usar novamente o *quadprog* do MATLAB, mas trocando x^i pela função $\Phi(x^i)$. Entretanto, não iremos realizar cálculos com tal problema, pois precisaremos fazer uso da função Φ, verificando o Teorema de Mercer e este não é o foco do trabalho.

Entretanto, para o problema dual não é necessário conhecer a função Φ, precisando apenas do produto interno com função Kernel $\Phi(x^i)^T\Phi(x^j) = K(x^j, x^i)$. Então, precisamos resolver o seguinte problema

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha^T H\alpha - e^T\alpha \\ s.a \quad & y^T\alpha = 0 \\ & 0 \leq \alpha \leq C \end{aligned}$$

onde $H(i,j) = y_i y_j K(x^i, x^j)$ e $e_j = 1, j = 1, \ldots, m$. Por esta razão resolvemos o dual e não o primal apresentado acima.

Portanto, usamos o programa *quadprog* do MATLAB para resolver tal problema, fizemos uso dos programas *CRCRD.m*, que resolve o problema considerando a função Kernel Gaussiana, do programa *CKCRDNH.m*, que resolve o problema considerando a função Kernel Polinomial não-homogênea e o programa *CKCRDH.m*, que resolve o problema considerando a função Kernel Polinomial homogênea, encontrados na Seção 6.

Usaremos as seguintes funções Kernel:

1. Kernel Gaussiana $K(x,y) = e^{\frac{-\|x-y\|^2}{2\sigma^2}}$;
2. Kernel Polinomial não-homogênea $K(x,y) = (x^Ty + k)^d$, com d e k parâmetros dados pelo usuário;
3. Kernel Polinomial homogênea, onde $k = 0$;

Para podermos classificar um novo ponto $z \in \mathbb{R}^N$, usamos a função de decisão

$$f(z) = w^T\Phi(x) + b$$

na construção do problema dual, temos que $w = \sum_{i=1}^{m} \alpha_i y_i \Phi(x^i)$, então a função de decisão passa a ser

$$f(z) = \sum_{i=1}^{m} \alpha_i y_i \Phi(x^i)^T\Phi(x) + b = \sum_{i=1}^{m} \alpha_i y_i K(x^i, x) + y_j - \sum_{i=1}^{m} \alpha_i y_i \Phi(x^i)^T\Phi(x^j).$$

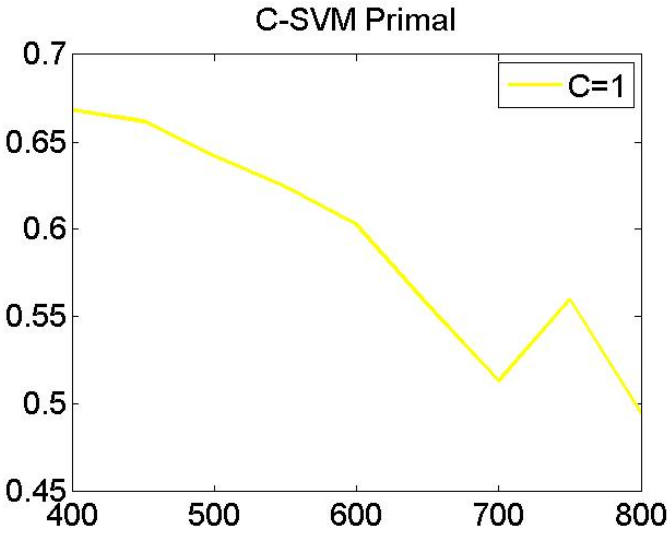
para algum $j \in V = \{i, \alpha(i) > 0\}$. Assim, se $f(z) > 0$ diremos que x é adimplente e inadimplente caso contrário. Usamos estes resultados para calcular a taxa de acerto e a Acurácia nos problemas abaixo.

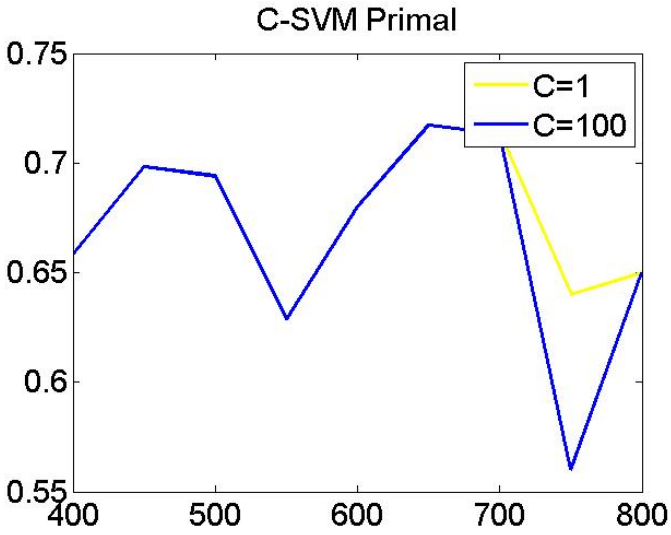
Para os seguintes valores de M (quantidade de dados no conjunto de treinamento) e variando o valor do parâmetro de regularização C, verificamos

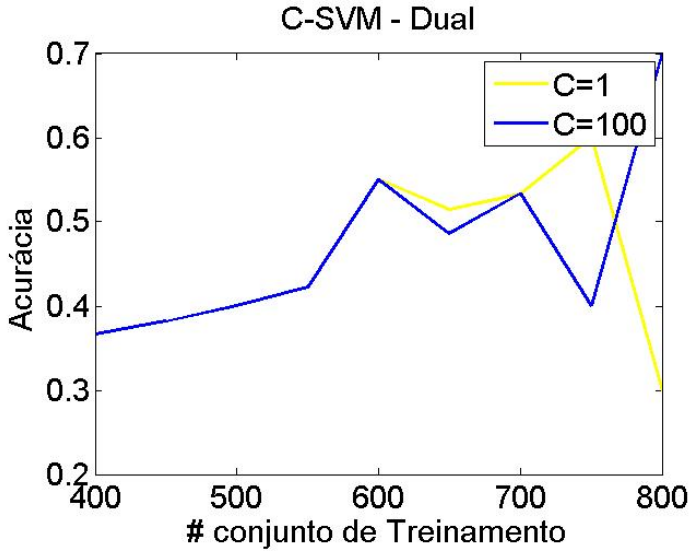
a taxa de acerto, calculada pelo programa $taxa3.m$ e a acurácia, calcula pelo programa $ACCSVM.m$, contidos na Seção 6.

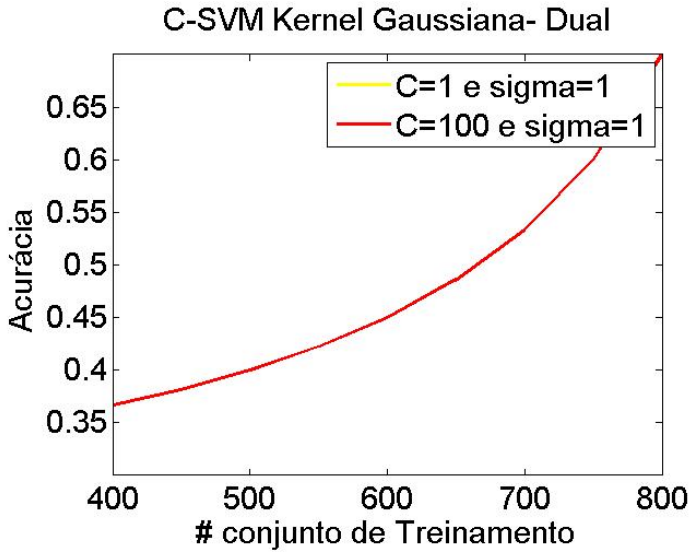
$M = [400\ 450\ 500\ 550\ 600\ 650\ 700\ 750\ 800];$ 75% bom e 25% mau

Assim, obtemos os seguintes gráficos,




C-SVM Primal
0.75
0.7
0.65
0.6
0.55
400
500
600
700
800
C=1
C=100


C-SVM - Dual
0.7
0.6
0.5
0.4
0.3
0.2
Acurácia
400
500
600
700
800
# conjunto de Treinamento
C=1
C=100



Tomamos como conjunto de treinamento 60% de clientes adimplentes e 20% de clientes inadimplentes. Para C-SVM Primal - Sem Kernel, tomando $C = 1$, obtemos $TA = 0.6400$ e $ACC = 0.6450$. Para C-SVM Dual - Sem Kernel, tomando $C = 1$, obtemos $TA = 0.500$ e $ACC = 0.500$.

Testamos para $C = 100$ e $C = 0.1$ e obtemos os mesmos resultados. Por este motivo, resolvemos testar outros parâmetros, mas agora recorrendo ao Truque do Kernel, para assim, obtermos uma Acurácia maior.

Tomando como conjunto de treinamento aleatoriamente 800 dados, obtemos a Acurácia e a quantidade de vetores de suporte (vs), para cada valor de C e σ da função Kernel Gaussiana, conforme podemos acompanhar na Tabela (4.3.2) . Realizamos o mesmo processo, mas agora variando os parâmetros k e d da função Kernel Polinomial, conforme podemos acompanhar na Tabela (4.3.2).

Dentro todos os testes realizados, os resultados em que obtemos melhor Acurácia (77,5 %)foi o método C-SVM com Kernel Gaussiano com parâmetro $C = 0.2500$ e $\sigma = 0.0625$.

Obtemos assim, o seguinte modelo, usando a técnica Support Vector Machine, para prever se dado um novo cliente ele será adimplente ou inadimplente,

$$f(x) = \sum_{i \in V} \alpha_i y_i K(x^i, x) + y_j - \sum_{i \in V} \alpha_i y_i K(x^i, x^j).$$

para algum $j \in V = \{i, \alpha(i) > 0\}$.

Assim, se $f(x) > 0$ diremos que x é adimplente e inadimplente caso contrário.

Acurácia	C	d
0.7400	0.0000	2
0.7000	0.0000	3
0.2850	0.0001	2
0.7050	0.0010	2
0.7200	0.0010	3
0.6800	0.0100	2
0.3100	0.0100	3
0.7300	0.1000	2
0.3050	1.0000	2
0.2950	1.0000	3
0.2900	10.0000	2
0.7100	100.0000	2

Tabela 6: Acurácia - Kernel Polinomial, k=0

4.3.3 Regressão Logística

Para o método Regressão Logística, utilizou-se o programa *fminunc* do MATLAB, pela facilidade no manuseio e pouco tempo de processamento.

Para o modelo a seguir, tomamos como ponto inicial o vetor zeros(21,1). Tentamos com o vetor rand(21,1), mas pelo algoritmo usado não conseguimos resolver o problema.

Para os seguintes valores de N, quantidade de dados no conjunto de treinamento, verificamos a taxa de acerto, calculada pelo programa *taxa.m* e a acurácia, calcula pelo programa *ACCLOG.m*, contidos na Seção 6.

$N = [400\ 450\ 500\ 550\ 600\ 650\ 700\ 750\ 800]$; 75% bom e 25% mau

Assim, obtemos os seguintes gráficos,

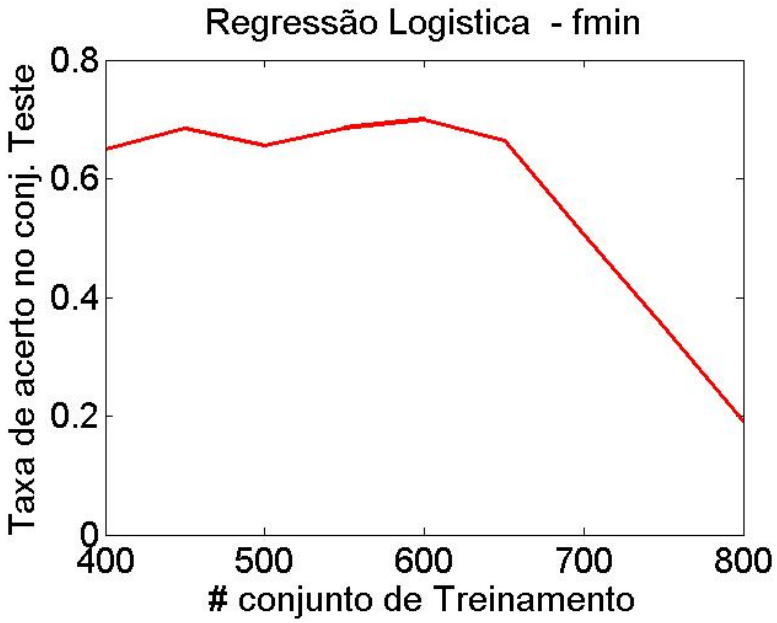


Figura 1: Gráfico N vs Taxa de acerto

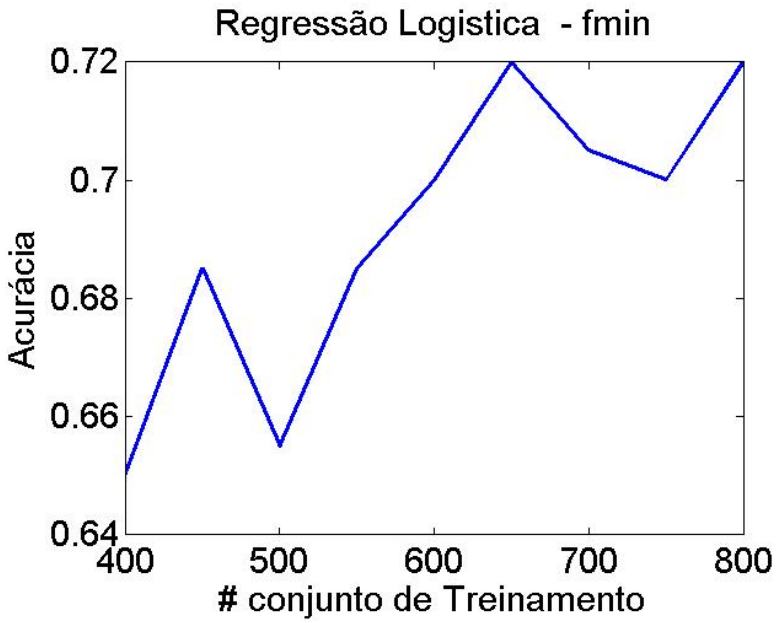


Figura 2: Gráfico N vs Acurácia

O maior valor que obtemos para Acurácia (72%), foi quando tomamos 640 dados de clientes adimplentes e 160 de inadimplentes para o Conjunto de Treinamento. Obtendo o seguinte modelo de previsão:

$$P_a = \frac{1}{e^{-\theta^T x}}$$

com $\theta = ...$ Logo, dado um novo cliente x, calculamos a probabilidade de ele ser adimplente. Nosso valor de corte, como discutido na Subseção 3.2 será 0.5, ou seja, se $P_a \geq 0.5$ diremos que o cliente será adimplente e inadimplente caso contrário.

Para o modelo a seguir, tomamos como ponto inicial o vetor rand(21,1), o qual gera um vetor em $\mathbb{R}^{21}$, aleatório com distribuição uniforme, pertencente ao intervalo [0,1]. Usamos também o programa *reglog.m*, programado como podemos consultar na Seção 6, pois para tal programa obtemos uma taxa de acerto maior do que quando usamos o $fminunc$.

Abaixo apresentamos um gráfico iteração vs função, para o método do gradiente acelerado de Nesterov, para obter o θ para o modelo de Regressão Logística.

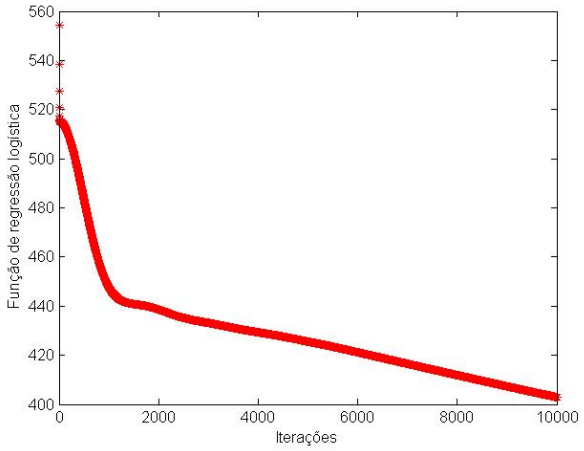


Figura 3: iteração vs função - Gradiente Acelerado de Nesterov

$$\theta = (0.0071, 0.1256, -0.0324, 0.0702, 0.0108, 0, 0.0738, 0.0565, -0.0109, 0.0405,$$

$$0.0127, 0.0133, -0.0132, 0.0188, 0.0389, 0.01, 0.0107, 0.0155, 0.0054, 0.0135, 0.0109)$$

Entretanto seu tempo computacional é elevado, por isso decidimos em usar somente o $fminunc$ para obter os resultados.

5 Conclusão e Trabalhos Futuros

O objetivo deste estudo foi construir um modelo de previsão de risco de crédito comparando os modelos SVM e Regressão Logística, analisando seu desempenho na predição de clientes adimplentes e inadimplentes.

Os dois modelos apresentados e testados no trabalho, mostraram um bom desempenho, entretanto o método SVM obteve melhores resultados.

Esses testes contribuíram para uma melhor compreensão da teoria do método SVM, confirmando seu uso ao problema de análise de crédito.

Analisamos os resultados das técnicas, vendo que o SVM apresentou maior número médio de acertos, mesmo variando as funções Kernel e seus parâmetros. Percebemos também que a mudança no banco de dados e treinamento, assim como a quantia para ambos, influência nos resultados citados acima. Entretanto, não é correto afirmar que quanto maior o conjunto de treinamento temos uma maior acurácia, mas sim que a escolha dos dados em tal conjunto aumenta a acurácia.

No mesmo site que obtemos o banco de dados, existem mais alguns bancos de dados relacionados a análise de crédito, que seria o da Austrália e um do Japão (Japanese Credit Screening). Em trabalhos futuros, podemos trabalhar um pouco nestes bancos de dados.

Podemos também, implementar os métodos para um número maior de casos, usar outras funções Kernel ou até mesmo variar mais os parâmetros das usadas aqui e fazer outros testes de validação para os métodos aqui abordados.

6 Códigos em Matlab

Esta seção é destinada a apresentar os algoritmos utilizados na pesquisa.

Cálculo da Acurácia para o modelo de Regressão Logística

```
%=====ACCLOG.m==================================
% Evelin Heringer Manoel Krulikovski - 2016
% Matriz de Confusão - Acurácia ACC - Regressão Logística
Tp=0; % contador de Verdadeiro positivo
Fp=0; % contador de Falso positivo
Fn=0; % contador de Faso negativo
Tn=0; % contador de Verdadeiro negativo
for  i=1:t
      if  ytel(i)==1
            if  mtheta(TEl(i,:)',theta)>= 0.5
            Tp=Tp+1;
            else  Fp=Fp+1;
            end
      else
```

```
            if  mtheta(TEl(i,:)',theta)>= 0.5
            Fn=Fn+1;
            else  Tn=Tn+1;
            end
        end
end
disp('A acurácia obtida foi: ')
ACC=(Tp+Tn)/(Tp+Fp+Tn+Fn)
```

Cálculo da Acurácia para o modelo SVM

```
%=====ACCSVM.m=====================================
% Evelin Heringer Manoel Krulikovski - 2016
% Matriz de Confusão - Acurácia ACC - SVM
Tp=0; % contador de Verdadeiro positivo
Fp=0; % contador de Falso positivo
Fn=0; % contador de Faso negativo
Tn=0; % contador de Verdadeiro negativo
for  i=1:t
        if  ytel(i)==1
            if  TE(i,:)*z(1:N)+z(N+1)>= 0.5
            Tp=Tp+1;
            else  Fp=Fp+1;
            end
        else
            if  TE(i,:)*z(1:N)+z(N+1)>= 0.5
            Fn=Fn+1;
            else  Tn=Tn+1;
            end
        end
end
disp('A acurácia obtida foi: ')
ACC=(Tp+Tn)/(Tp+Fp+Tn+Fn)
```

Taxa de acerto do modelo no conjunto de teste - Regressão Logística

```
%=====Taxa.m=======================================
% Evelin Heringer Manoel Krulikovski - 2016
% Taxa de acerto do modelo no conjunto de teste
c = 0; % contador para amostras classificadas corretamente
for i = 1:t
        m = mtheta(TEl(i,:)',theta);
        if m >= 0.5
                x(i) = 1;
        else x(i) = 0;
        end
```

```
end
for i = 1:t
        if (x(i) == 1)
                if ytel(i)==1
                c = c+1;
                end
        else
                if ytel(i)==0
                c=c+1;
                end
        end
end
disp('Taxa de acerto Modelo = ')
c/t
```

Obtenção de theta para o modelo Regressão Logística

```
%=====fmin.m=================================
% Evelin Heringer Manoel Krulikovski - 2016
% Obtenção de theta para o modelo Regressão Logística
dados % Leitura dos dados
theta=input('Forneca um ponto inicial em ℝ^21: ');
[theta,fval,exitflag,output,grad]=fminunc(@ftheta2,theta);
Taxa % Taxa de acerto
ACCLOG % Critério ACC
```

Cálculo da Função Regressão Logística

```
function[f]=ftheta2(theta)
dados % Leitura dos dados
        f=0;
        for i=1:size(TRl,1)
                mth=mtheta(TRl(i,:)',theta);
                f=f+ytrl(i)*log(mth)+(1-ytrl(i))*lo g(1-mth);
        end
        f=-f;
end
```

Cálculo da função mtheta

```
function [mtheta]= mtheta(x,theta)
mtheta= 1/(1+exp(-theta'*x));
end
```

Cálculo para o modelo C-SVM - Primal - Sem Kernel

```
%=====SKCR.m=================================
% Evelin Heringer Manoel Krulikovski - 2016
clc
clear
```

```
dados % Leitura dos dados
C=input(' Forneça o parâmetro C: ');
H=[eye(N) zeros(N,M+1);
zeros(M+1,M+N+1)];
f=[zeros(N+1,1); C*ones(M,1)];
c=-ones(M,1);
for i=1:M
        for j=1:N
        A(i,j)=-ytr(i)*TR(i,j);
        end
A(i,N+1)=-ytr(i);
end
A=[A -eye(M,M)];
for i=1:N+1
        LB(i)=-inf;%LB para theta
end
for i=(N+2):N+M+1
        LB(i)=0; %LB para xi
end
UB=inf*ones(N+M+1,1);
[z,fval,exitflag]=quadprog(H,f,A,c,[ ],[ ],LB,UB)
Taxa3 % Taxa de acerto
ACCSVM % Acurácia
```

Cálculo para o modelo C-SVM - Dual - Sem Kernel

```
%=====SKCRD.m=================================
% Evelin Heringer Manoel Krulikovski - 2016
clc
clear
dados % Leitura dos dados
C=input(' Forneça o parâmetro C: ');
f=-ones(M,1);
for i=1:M
        for j=1:M
        H(i,j)=ytr(i)*ytr(j)*TR(i,:)*TR(j,:)';
        end
end
LB=zeros(M,1);
UB=C*ones(M,1);
Aeq=ytr';
ceq=0;
[alpha,fval,exitflag]=quadprog(H,f,[],[],Aeq,ceq,LB,UB)
%Obtenção de w e b
sum=0;
cont=0; % contador de vetor de suporte
```

```
for i=1:M
        if alpha(i)> 10^-4
        k=i;
        sum=sum+ytr(i)*alpha(i)*TR(i,:)';
        cont=cont+1;
        end
end
sum=0;
Z(1:N)=sum; % Obtenção de w
sum1=0;
for i=1:M
        sum1=sum1+alpha(i)*ytr(i)*TR(i,:)*TR(k,:)';
end
Z(N+1)=ytr(k)+sum1;% Obtenção de b
z=Z';
Taxa3 % Taxa de acerto
ACCSVM % Acurácia
```

Cálculo para o modelo C-SVM - Dual - Com Kernel

```
%======CKCRD.m=================================
% Evelin Heringer Manoel Krulikovski - 2016
clc
clear
dados % Leitura dos dados
C=input(' Forneça o parâmetro C: ');
sigma=input(' Forneça o parâmetro sigma: ');
f=-ones(M,1);
for i=1:M
        for j=1:M
        H(i,j)=ytr(i)*ytr(j)*gaus(TR(i,:)',TR(j,:)',sigma);
        end
end
LB=zeros(M,1);
UB=C*ones(M,1);
Aeq=ytr';
ceq=0;
[alpha,fval,exitflag]=quadprog(H,f,[ ],[ ],Aeq,ceq,LB,UB) %Obtenção de w e b
sum=0;
cont=0; % contador de vetor de suporte
for i=1:M
        if alpha(i)> 10^-4
        k=i;
        sum=sum+ytr(i)*alpha(i)*TR(i,:)';
        cont=cont+1;
```

```
        end
end
sum=0;
Z(1:N)=sum; % Obtenção de w
sum1=0;
for i=1:M
        sum1=sum1+alpha(i)*ytr(i)*TR(i,:)*TR(k,:)';
end
Z(N+1)=ytr(k)+sum1;% Obtenção de b
z=Z';
Taxa3 % Taxa de acerto
ACCSVM % Acurácia
```

Cálculo da função Kernel Gaussiana

```
function [g]=gaus(x1,x2,sigma)
```

$g = exp(-(norm(x1 - x2)^2)/(2 * sigma^2))$

```
end
```

Taxa de acerto do modelo no conjunto de teste - SVM

```
%======Taxa3.m=====================================
% Evelin Heringer Manoel Krulikovski - 2016
% Taxa de acerto do modelo no conjunto de teste
c = 0; % contador para amostras classificadas corretamente
for i = 1:t
        m = TE(i,:)*z(1:N)+z(N+1);
        if m >= 0.5
                x(i) = 1;
        else x(i) = -1;
        end
end
for i = 1:t
        if (x(i) == 1)
                if yte(i)==1
                c = c+1;
                end
        else
                if yte(i)==-1
                c=c+1;
                end
        end
end
disp('Taxa de acerto Modelo = ')
c/t
```

Cálculo para o modelo C-SVM - Dual - Com Kernel

```
%======CKCRDH.m=====================================
```

```
% Evelin Heringer Manoel Krulikovski - 2016
clc
clear
dados % Leitura dos dados
C=input(' Forneça o parâmetro C: ');
d=input('Forneça o parâmetro d: ');
f=-ones(M,1);
for i=1:M
        forfor j=1:M
        H(i,j)=ytr(i)*ytr(j)*ph(TR(i,:)',TR(j,:)',d);
        end
end
LB=zeros(M,1);
UB=C*ones(M,1);
Aeq=ytr';
ceq=0;
[alpha,fval,exitflag]=quadprog(H,f,[],[],Aeq,ceq,LB,UB) %Obtenção de w e b
sum=0;
cont=0; % contador de vetor de suporte
for i=1:M
        if alpha(i)> 10^-4
        k=i;
        sum=sum+ytr(i)*alpha(i)*TR(i,:)';
        cont=cont+1;
        end
end
sum=0;
Z(1:N)=sum; % Obtenção de w
sum1=0;
for i=1:M
        sum1=sum1+alpha(i)*ytr(i)*TR(i,:)*TR(k,:)';
end
Z(N+1)=ytr(k)+sum1;% Obtenção de b
z=Z';
Taxa3 % Taxa de acerto
ACCSVM % Acurácia
```

Cálculo para o modelo C-SVM - Dual - Com Kernel

```
%=====CKCRDNH.m==============================
% Evelin Heringer Manoel Krulikovski - 2016
clc
clear
dados % Leitura dos dados
C=input(' Forneça o parâmetro C: ');
d=input('Forneça o parâmetro d: ');
```

```
k=input('Forneça o parâmetro k: ');
f=-ones(M,1);
for i=1:M
        forfor j=1:M
        H(i,j)=ytr(i)*ytr(j)*pnh(TR(i,:)',TR(j,:)',k,d);
        end
end
LB=zeros(M,1);
UB=C*ones(M,1);
Aeq=ytr';
ceq=0;
[alpha,fval,exitflag]=quadprog(H,f,[],[],Aeq,ceq,LB,UB) %Obtenção de w e b
sum=0;
cont=0; % contador de vetor de suporte
for i=1:M
        if alpha(i)> 10^-4 % Pode ser valores mais baixos
        k=i;
        sum=sum+ytr(i)*alpha(i)*TR(i,:)';
        cont=cont+1;
        end
end
sum=0;
Z(1:N)=sum; % Obtenção de w
sum1=0;
for i=1:M
        sum1=sum1+alpha(i)*ytr(i)*TR(i,:)*TR(k,:)';
end
Z(N+1)=ytr(k)+sum1;% Obtenção de b
z=Z';
Taxa3 % Taxa de acerto
ACCSVM % Acurácia
```

Cálculo da função Kernel Polinomial Homogênea

```
function [ph]=ph(x1,x2,d)
ph=(x1' * x2)^d;
end
```

Cálculo da função Kernel Polinomial não -homogênea

```
function [pnh]=pnh(x1,x2,k,d)
pnh=(x1' * x2 + k)^d;
end
```

Obtenção de theta com Algoritmo de Nesterov

```
%=====reglog.m==============================
% Evelin Heringer Manoel Krulikovski - 2016
dados % Leitura dos dados
[theta, k, f, n] = gan(ytrl, TRl); % Método do gradiente acelerado de Nesterov
```

```
Taxa % Taxa de acerto
ACCLOG % Critério ACC
```

Gradiente Acelerado de Nesterov

```
%=====gan.m==================================
% Evelin Heringer Manoel Krulikovski - 2016
function [theta,k,f,n] = gan(y1,X)
theta=input('Forneca um ponto inicial: ');
L=norm(X)^2;
y=theta;
lambda=0;
k=0;
[mn] = size(X);
while
    norm(gradf(X,theta,m,y1)) > 10^-2
    yold=y;
    y=theta-(1/L)*gradf(X,theta,m,y1);
    lambdaold=lambda;
    lambda=(1 + sqrt(1 + 4 * lambdaold^2))/2;
    gama=(1-lambdaold)/lambda;
    theta=(1-gama)*y+gama*yold;
    k=k+1
    %Caso queira plotar o gráfico função vs iteração:
    plot(k,ftheta(theta,X,y1),'*r');
    xlabel('Iterações') % eixo horizontal
    ylabel('Função de regressão logística') % eixo vertical
    hold on
end
f=ftheta(theta,X,y1);
n=norm(gradf(X,theta,m,y1));
end
```

Função Regressão Logística

```
function[f]=ftheta(theta,TRl,ytrl)
    f=0;
    for i=1:size(TRl,1)
    mth=mtheta(TRl(i,:)',theta);
    f=f+ytrl(i)*log(mth)+(1-ytrl(i))*log(1-mth);
    end
    f=-f;
end
```

Gradiente Regressão Logística

```
function[grad]=gradf(X,theta,m,y)
    for i=1:m
```

```
        alpha(i)=mtheta(X(i,:)', theta);
    end
    grad=X'*(alpha'-y);
end
```

7 Referências

[1] Banco de dados German Credit Data, https://archive.ics.uci.edu/ml/ datasets/statlog+(german+credit+data), acessado em 05/10/2016.

[2] V. N. Vapnik. *The nature of Statistical learning theory*. Springer-Verlag, New York, 1995.

[3] V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, 1998.

[4] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[5] Credit Score usando o R, http://pedrounb.blogspot.com.br/2015/09/credit-score-usando-o-r.html, acessado em 05/10/2016.

[6] Projects Machine Learning, http://cs229.stanford.edu/projects2016spr.html, acessado em 05/10/2016.

[7] A.A.Ribeiro and E. W. Karas. *Otimização Contínua: Aspectos teóricos e computacionais*. Cengage Learning, Brazil, 2013.

[8] A. S. Manuel. "O incumprimento dos empréstimos no mercado de microcrédito do sistema bancário angolano."*Universidade de Coimbra - Faculdade de Economia* (2010).

[9] S. Sinhorigno. "Previsão de inadimplência de transações com cartão de crédito."*Universidade de São Paulo - USP* (2007).

C	sigma	Acurácia	vs	C	sigma	Acurácia	vs
0.0313	0.0313	0.7500	800	1.0000	0.0313	0.7300	800
0.0313	0.0625	0.6850	800	1.0000	0.0625	0.6850	800
0.0313	0.1250	0.6900	800	1.0000	0.1250	0.7050	800
0.0313	0.2500	0.3300	800	1.0000	0.2500	0.3200	800
0.0313	0.5000	0.3250	800	1.0000	0.5000	0.6800	800
0.0313	1.0000	0.7150	800	1.0000	1.0000	0.3350	800
0.0313	2.0000	0.7250	800	1.0000	2.0000	0.2700	800
0.0313	4.0000	0.7550	800	1.0000	4.0000	0.3050	800
0.0313	8.0000	0.7400	787	1.0000	8.0000	0.3200	787
0.0313	16.0000	0.7000	658	1.0000	16.0000	0.7300	664
0.0313	32.0000	0.7150	0	1.0000	32.0000	0.6550	0
0.0625	0.0313	0.2800	800	2.0000	0.0313	0.7550	800
0.0625	0.0625	0.7050	800	2.0000	0.0625	0.7000	800
0.0625	0.1250	0.7200	800	2.0000	0.1250	0.7100	800
0.0625	0.2500	0.6800	800	2.0000	0.2500	0.6950	800
0.0625	0.5000	0.3100	800	2.0000	0.5000	0.6650	800
0.0625	1.0000	0.7300	800	2.0000	1.0000	0.7050	800
0.0625	2.0000	0.6950	800	2.0000	2.0000	0.6800	800
0.0625	4.0000	0.6950	800	2.0000	4.0000	0.2800	800
0.0625	8.0000	0.2950	781	2.0000	8.0000	0.3450	794
0.0625	16.0000	0.2900	652	2.0000	16.0000	0.2850	650
0.0625	32.0000	0.7350	0	2.0000	32.0000	0.2750	0
0.1250	0.0313	0.7100	800	4.0000	0.0313	0.7050	800
0.1250	0.0625	0.7150	800	4.0000	0.0625	0.6950	800
0.1250	0.1250	0.7150	800	4.0000	0.1250	0.6850	800
0.1250	0.2500	0.2950	800	4.0000	0.2500	0.7350	800
0.1250	0.5000	0.6650	800	4.0000	0.5000	0.6400	800
0.1250	1.0000	0.7000	800	4.0000	1.0000	0.2800	800
0.1250	2.0000	0.3450	800	4.0000	2.0000	0.6550	800
0.1250	4.0000	0.6350	800	4.0000	4.0000	0.7300	800
0.1250	8.0000	0.2600	783	4.0000	8.0000	0.3150	787
0.1250	16.0000	0.7050	656	4.0000	16.0000	0.3200	617
0.1250	32.0000	0.6650	0	4.0000	32.0000	0.6900	0
0.2500	0.0313	0.7200	800	8.0000	0.0313	0.7350	800
0.2500	0.0625	0.7750	800	8.0000	0.0625	0.3000	800
0.2500	0.1250	0.6600	800	8.0000	0.1250	0.2950	800
0.2500	0.2500	0.3100	800	8.0000	0.2500	0.2700	800
0.2500	0.5000	0.7050	800	8.0000	0.5000	0.3050	800
0.2500	1.0000	0.3450	800	8.0000	1.0000	0.7000	800
0.2500	2.0000	0.7000	800	8.0000	2.0000	0.6600	800
0.2500	4.0000	0.2850	800	8.0000	4.0000	0.3100	800
0.2500	8.0000	0.3100	779	8.0000	8.0000	0.3000	785
0.2500	16.0000	0.3200	622	8.0000	16.0000	0.3250	641
0.2500	32.0000	0.7350	0	8.0000	32.0000	0.6350	0
0.5000	0.0313	0.7400	800	16.0000	0.0313	0.3400	800

Tabela 7: Acurácia e vetores de suporte para parâmetro C e σ