UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

Valdecy Vieira Junior

CLASSIFICAÇÃO DE DADOS USANDO TÉCNICAS DE DATAMINIG E APRENDIZADO DE MÁQUINA

São Luís
2013

VALDECY VIEIRA JUNIOR

CLASSIFICAÇÃO DE DADOS USANDO TÉCNICAS DE DATAMINIG E APRENDIZADO DE MÁQUINA

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientadora: Prof. Dr. Alexandre César Muniz de Oliveira

São Luís

2013

Vieira Junior, Valdecy

Classificação de dados usando técnicas de datamaning e aprendizado de máquina/ Valdecy Vieira Junior. – São Luís, 2013.

62f.

Impresso por computador (Fotocópia).

Orientadora: Alexandre César Muniz de Oliveira.

Monografia (Graduação) – Universidade Federal do Maranhão, Curso de Ciência da Computação, 2013.

1.Base de dados – Classificação 2.KDD 3.Método SVM. I. Título.

CDU 004.633.2

Valdecy Vieira Junior

CLASSIFICAÇÃO DE DADOS USANDO TÉCNICAS DE DATAMINIG E APRENDIZADO DE MÁQUINA

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Aprovada em: 28 / 02 / 2013

BANCA EXAMINADORA

Prof. Dr. Alexandre César Muniz de Oliveira (Orientador)
Universidade Federal do Maranhão

Valeska Martins de Souza

Profª. Dr. Valeska Martins de Souza
Universidade Federal do Maranhão

Portela

Prof. Msc.Carlos Eduardo Portela Serra de Castro
Universidade Federal do Maranhão

AGRADECIMENTOS

O fim de uma jornada é sem dúvida o momento ideal para agradecer as pessoas que contribuíram para que meu objetivo fosse alcançado.

Primeiramente a Deus, por ter me dado força durante toda essa trajetória, pois, é Ele que nos guia, protege, orienta e conduz nossas vidas.

Aos meus pais Valdecy Vieira e Bernadete Fernandes Vieira e aos meus irmãos Darcio Vieira, Nirvana Vieira, Ellem Vieira e Rafael Vieira que contribuíram com grande incentivo em toda minha vida e jornada acadêmico.

À minha querida noiva Thaliane Maia Silva pelo incentivo, apoio e compreensão em todos os momentos.

Ao meu primo-pai Gladston Fernandes de Araújo que me deu total apoio para que eu pudesse alcançar meu objetivo.

A todos meus familiares em especial a meu avô Saraiva Fernandes pelo incentivo e companheirismo.

Aos meus primos, em especial a Danilo Vieira, Thiago Aróso, Jéssica Aróso, Gabriel Costa e Jorge Ricardo pelo incentivo em todos os momentos de bate-papo.

A meu orientador Prof. Dr. Alexandre César Muniz de Oliveira por aceitar a coordenação deste trabalho de conclusão de curso, pois com sua orientação, dedicação e auxílio, pude desenvolver e concluir este trabalho.

Aos amigos Jeferson Costa, Elizaldo Pinheiro, Marcelo Oliveira e Tales Maia.

À professora Tatiana pelo ensinamento e orientação dada no momento importante de minha vida.

A todos que direta ou indiretamente contribuíram para que este trabalho fosse realizado com sucesso.

Obrigado!

“Deus me defende dos amigos, que dos inimigos me defendo eu”.

(Valtaire)

RESUMO

O *Knowledge* Discovery in *Database* é uma técnica de descoberta de conhecimento em Base de Dados que identifica padrões úteis em dados não processados. O KDD é uma técnica que possui várias etapas, sendo o processo de Mineração de Dados a mais importante na extração de conhecimento. Entre os diversos algoritmos usados nessa etapa, explicamos um das técnicas de classificação de dados mais antigas, o método SVM. Esse trabalho, além de esclarecer o algoritmo de SVM (*Support Vector Machine*), buscou aplicá-lo a um estudo de caso, onde um pesquisador médico obteve um conjunto de dados que contém as características de um certo número de amostras de células humanas extraídas de pacientes que se acreditava estar em risco de desenvolver câncer. Essa base de dados esta disponível na UCI Repositório (Assunção e Newman, 2007), além disso consta com exemplo no repositório do software *Clementine,* onde este gerou uma classificação dessas. Esta classsificação foi feita pelos *Kernels* do SVM, ou seja, o *Kernel Sibmoidal*, Linear, RBF (Gausiano) e o *Polynomial*, além da comparação feita entre *Kernels*, foi feita também uma analise entre diversos algoritmos, tais como: QUEST, CHAID e o *Neural Network*

Palavras-chave: KDD. Classificação. Método SVM.

ABSTRACT

The Knowledge Discovery in Database is a technique for knowledge discovery in database identifying useful patterns in raw data. KDD is a technique that has several stages, the process of Data Mining in the most important knowledge extraction. Among the different algorithms used in this step, explained one of the techniques of data classification oldest, the SVM method. This work, as well as clarifying algorithm SVM (Support Vector Machine), we attempted to apply it to a case where a medical researcher obtained a data set that contains the characteristics of a number of samples extracted from human cells patients believed to be at risk of developing cancer. This database is available at the UCI Repository (Asuncion and Newman, 2007) also appears in the example with Clementine software repository, where it generated a classification of these. This was done by CLASSIFICATION SVM kernels, namely the Kernel Sibmoidal, Linear, RBF (Gausiano) and polynomial, and the comparison made between kernels, an analysis was also made between different algorithms, such as QUEST, CHAID and Neural Network.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

LISTA DE SIGLAS

SVM – *Support Vector Machine*

DM - *Data Mining*

DW - *Data Warehounsing*

KDD - *Knowledge Discovery in Database*

MD - Mineração de Dados

AM – Aprendizado de Máquina

VC – *Vapnik-Chervonenkis*

RBF - *Radial-Basis Function*

SUMÁRIO

1. INTRODUÇÃO

As últimas décadas acompanharam um aumento dramático na quantidade de informações armazenadas em formato eletrônico. Esta acumulação aconteceu a uma taxa explosiva. Pesquisas mostram que a quantia de informação dobra a cada 20 meses e o tamanho e número de bancos de dados estão aumentando ainda mais rapidamente.

Estes dados armazenados estão tipicamente ligados à capacidade de extrair informações de mais alto nível que se encontra subjacentes a estes dados, ou seja, informação útil que sirva para dar suporte a decisões, e para exploração e melhor entendimento do fenômeno gerador dos dados. Podem existir padrões ou tendências úteis interessantes que, se descobertos, podem ser utilizados.

A maioria das operações e atividades das instituições públicas e privadas é registrada computacionalmente e acumula-se em grandes bases de dados; existe a necessidade de extrair conhecimento dessas fontes de dados, a fim de descobrir relações ocultas, padrões e regras para predizer e correlacionar dados, que podem ajudar as instituições nas tomadas de decisões (PACHECO et al., 1999).

O conjunto de técnicas para descobrir padrões ou extrair conhecimento em dados não processados é chamada de Descoberta de Conhecimento em Bases de Dados, ou em inglês *Knowledge Discovery in Database (*KDD). O KDD é apoiado em técnicas de Mineração de Dados, ou *Data Mining* (DM), que transforma dados em informação (MANNILA, 1997 apud KLEINSCHMIDT, 2007).

A informação e o conhecimento são prerrogativas estratégicas e indispensáveis na procura de maior autonomia nas ações das empresas, domínio social e na tomada de decisão com períodos cada vez menores. Por isso, várias empresas nacionais e internacionais de produção, consumo, mercado financeiro e instituições de ensino já adotaram nas suas rotinas o DM para monitorar arrecadações, consumo de clientes, prevenir fraudes além da previsão de riscos do mercado, dentre outras (DIAS, 2002).

Os algoritmos e as técnicas usados no processo de KDD provêm de inúmeras áreas, como por exemplo, a Estatística. Este processo envolve o uso de algumas tarefas de KDD, tais como: Classificação, Associação e Agrupamento. Tais tarefas utilizam técnicas de DM baseadas em Redes Neurais Artificiais, Árvores de Decisão, Algoritmos Genéticos, Métodos Bayesianos, entre outras (CARVALHO, 2001).

Entre as diversas técnicas e algoritmos usados no processo de KDD, o método SVM foi escolhido, pois é um método utilizado quando a segmentação é definida em termos de características genéticas aceitando variáveis categóricas nominais ou ordinais como variáveis dependentes. Normalmente, este tipo de variável é utilizado em pesquisas tradicionais.

As técnicas de Aprendizado de Máquina (AM) empregam um princípio de inferência denominado indução, no qual se obtém conclusões genéricas a partir de um conjunto particular de exemplos. O aprendizado indutivo pode ser dividido em dois principais: supervisionado e não supervisionado.

As Máquinas de Vetores de Suporte (SVMs do Inglês *Support Vector Machine*) constituem uma técnica de aprendizado que vem recebendo crescente atenção da comunidade de Aprendizado de Máquina. Os resultados da aplicação dessa técnica são comparáveis e muitas vezes superiores aos obtidos por outros algoritmos de aprendizado, como redes neurais artificiais. Exemplos de aplicações de sucesso podem ser encontrados em diversos domínios, como categorização de texto, imagens e biomedicina.

As SVMs são embasadas pela teoria do aprendizado estatístico, desenvolvida por *Vapnik*. Essa teoria estabelece uma série de princípios que devem ser seguidos na obtenção de classificadores com boa generalização, definidos como a sua capacidade de prever corretamente a classe de novos dados do mesmo domínio em que o aprendizado ocorreu.

Sendo assim, esta monografia tem como objetivo geral, apresentar um estudo sobre o algoritmo SVM e aplicá-lo a um estudo de caso. Como objetivos específicos:

- Estudar a Mineração de Dados, pois é a etapa mais importante no processo de KDD.
- Estudar a tarefa de Classificação do KDD, a fim de compreender essa tarefa a qual o método SVM pertence.
- Estudar o processo de *Knowledge Discovery in Database* (KDD), a fim de entender todas as fases necessárias para a extração de conhecimento em base de dados.
- Estudar os *Kernels*, pois sua compreensão é essencial para interpretar e comparar os resultados gerados pelo SVM.
- Aplicar o algoritmo a um estudo de caso para interpretar seus resultados.

- Estudar o algoritmo SVM, pois esse é o objeto de estudo desta monografia.

Nesta monografia, utilizamos no estudo de caso, as informações da base de dados *cell_samples.data* do ano de 2007, que possui diversas informações de um certo número de amostras de células extraídas de pacientes que se acreditava estarem em risco de desenvolver câncer. O SVM usa valores destas características de células para indicar se as amostras são benignas ou malignas.

O presente trabalho é composto de mais três capítulos, conforme descrição sumária a seguir:

Capítulo 2 – apresenta a fundamentação teórica com os conceitos de KDD, suas fases e principais tarefas, além da técnica de redes neurais. É nesse capítulo que apresentamos também de forma detalhada o método SVM.

Capítulo 3 – nele mostramos o nosso estudo de caso que usa amostras de células para verificar se tais pacientes está em risco de desenvolver câncer, o procedimento seguido para utilizar o método SVM e os resultados encontrados pelo algoritmo.

Capítulo 4 – apresenta as conclusões levantadas com o conhecimento minerado e as sugestões para trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são utilizados temas relevantes para compreensão deste trabalho, onde partiremos da obtenção de conhecimento sobre base de dados, Classificadores, Aprendizado de Máquina e, enfim, o método *de classificação Support Vector Machines* (SVM) que é o objeto de estudo desta monografia.

2.1 Knowledge Discovery in Database (KDD)

A técnica de obter padrões úteis em dados não processados é representada por diversos nomes, dentre eles *Knowledger Discovery in Database* (KDD) ou Descoberta de Conhecimento em Base de Dados, *Data Mining* (DM) extração de conhecimento, descoberta de informação e processamento de padrões de dados. O KDD foi criado em 1989 para referenciar o processo de descoberta de conhecimento em dados.

O KDD é o processo não trivial de reconhecer em dados padrões que sejam válidos, novos (que ainda não foram identificados), potencialmente úteis e compreensíveis, visando melhorar o entendimento de um problema ou um procedimento útil na tomada de decisão (FAYYAD et al. 1996 apud KLEINSCHMIDT, 2007).

O *Knowledger Discovery in Database* (KDD) também é definido como um método que possibilita averiguar um grande contingente de dados, empregando técnicas aproximadas. O procedimento do KDD constitui-se em duas etapas fundamentais, o armazenamento de informações e mineração de dados. Primeiro é importante e essencial criar uma base de dados organizada e com informações suficientes sobre o conteúdo a analisar, depois se utiliza métodos aproximados que aceitem minerar os dados, para encontrar as relações contidas em tais dados (COLLAZOS; BARRETO, 2003 apud KLEINSCHMIDT, 2007).

Esse tipo de técnica possui várias ferramentas poderosas para a exploração eficaz de informações em grandes bases de dados, na intenção de auxiliar na tomada de decisão, sendo o DM uma das fases do processo de maior importância. Toda a metodologia pode ser dividida em cinco etapas principais: *Warehounsing* (DW), o Pré-processamento, o Enriquecimento, a Mineração de Dados e por último,

o Pós-processamento. A Figura 2.1 representa o processo de KDD de forma hierárquica destacando fases e tarefas (AURÉLIO et al., 1999).

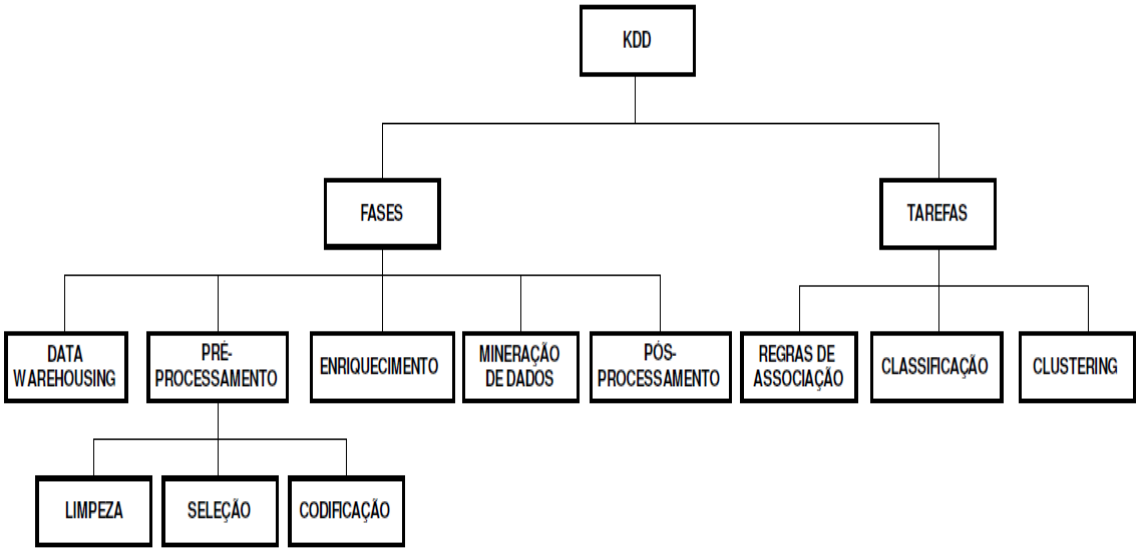


Figura 2.1- Visão hierárquica do processo de KDD (AURÉLIO et al., 1999)

2.1.1 *Data Warehounsing* (DW)

A Data *Warehounsing* (DW) facilita a análise de grandes volumes de dados coletados dos sistemas transacionais. Sendo estas as chamadas séries históricas que possibilita uma essencial análise de eventos passados, oferecendo suporte às tomadas de decisões presentes e a previsão de eventos futuros

Atualmente, por sua capacidade de sumarizar e analisar grandes volumes de dados, a Data *Warehounsing* é o núcleo dos sistemas de informações gerenciais e apoio à decisão das principais soluções. Segundo *Inmon*, Data *Warehouse* é uma coleção de dados orientados por assuntos integrados, variáveis com o tempo e não voláteis, para dar suporte ao processo de tomada de decisão.

Além disso, é uma arquitetura que organiza, totaliza e armazena dados permitindo traçar estratégias de maneira confiável dando suporte ao nível gerencial nas tomadas de decisão.

2.1.2 Pré-Processamento

Nesta fase se tem a intenção de consolidar as informações relevantes para o algoritmo minerador, com o objetivo de reduzir o grau de complexidade do problema em análise (PACHECO et al., 1999). O processo divide-se em três fases: Limpeza de dados: esta etapa é responsável por realizar a correção de eventuais erros existentes e eliminação de valores nulos e redundantes. Ela melhora a base de dados retirando consultas indesejadas que seriam executadas pelo algoritmo de mineração e que consequentemente afetariam o seu processamento.

- Seleção de dados: nesta fase o objetivo é escolher os atributos mais importantes no conjunto de atributos existentes na base de dados.
- Codificação dos dados: fase que decompõe valores contínuos dos atributos em uma lista de intervalos, convertendo valores quantitativos em valores categóricos, cujo objetivo facilita a qualidade de resultados.

2.1.3 Enriquecimento dos dados

Esta técnica tem como objetivo obter mais informações aos dados existentes, melhorando os dados, podendo ser realizadas pesquisas para adicionar conhecimento, como consultas a banco de dados externos, entre outras técnicas (DIAS, 2002).

O melhoramento dos dados consiste em melhorar a informação contida nos atributos dos bancos de dados através da elaboração de novos registros a partir dos já existentes, adicionando novos dados. A geração de totalizadores em variáveis numéricas, a criação de faixa ou classes de valores para registros contínuos e a generalização de valores de atributos são exemplos de enriquecimento dos dados (CARVALHO, 2001).

2.1.4 Mineração de Dados (MD)

É uma área multidisciplinar que adiciona técnicas utilizadas em diversas áreas como Inteligência Artificial (especialmente a aprendizagem de máquina), Banco de

Dados (recursos para manipular grandes volumes de dados) e Estatística (na avaliação e validação dos dados).

O principal objetivo é encontrar os relacionamentos entre as informações e fornecer uma fonte para que possa ser realizada uma previsão de tendências futuras baseadas em dados históricos (DIAS, 2002).

Poderá ser determinada como a utilização de técnicas automáticas de exploração de grandes quantidades de dados de forma a reconhecer novos padrões e relações que, devido ao volume de informações, não seriam facilmente descobertos (CARVALHO, 2001).

A Mineração de Dados é de grande importância em aplicações, como por exemplo, análises financeiras e de investimentos, detecção e predição de erros em grandes empresas, na área de segurança para detectar fraudes em cartões de créditos, análise de informações, *marketing,* limpeza em bases de dados e melhoria no processo industrial entre outros.

Também tem como objetivos a previsão e descrição de modelos. A previsão pode ser adquirida através da utilização de variáveis contidas na base de dados para prever valores não conhecidos ou futuros. Essa descrição é baseada na descoberta de padrões interpretáveis pelos humanos. Dentro do processo de KDD, descrever modelos possui maior importância que prever os mesmos. A previsão e a descrição dos modelos são conseguidas selecionando as tarefas, algoritmos e técnicas de extração de dados (FAYYAD et al., 1996 apud KLEINSCHMIDT 2007).

As técnicas e os algoritmos usados para desenvolver modelos a partir de informações provêm de diversas áreas como Reconhecimento de Padrões e Estatística. Estas técnicas, muitas vezes, podem ser combinadas para adquirir melhores resultados (SILVA, 2003).

As principais tarefas de KDD são: Classificação, Associação e Agrupamento. Estas tarefas podem utilizar técnicas de Mineração Dados baseadas em Redes Neurais Artificiais, Árvores de Decisão, Estatística, classificação entre outras.

Abaixo é mostrado um Quadro 2.1 com as principais tarefas de KDD e consequentemente algumas das técnicas mais utilizadas para a Mineração de Dados.

Quadro 2.1 - Tarefas e Técnicas de KDD

Tarefas de KDD	Técnicas
Associação	Estatística e Teoria dos Conjuntos
Classificação	Árvores de Decisão, Redes Neurais e Algoritmos Genéricos
Agrupamento ou *Clustering*	Redes Neurais e Estatística
Previsão de Séries Temporais	Lógica Nebulosa e Redes Neurais

Fonte: AURÉLIO et al., 1999.

Como podemos observar no Quadro acima, uma tarefa está proporcionalmente relacionada ao domínio da aplicação e interesse do usuário, onde cada uma possui um conjunto de técnicas. Cada tarefa de KDD extrai um tipo diferente de conhecimento da base de dados, logo, será necessário de um algoritmo diferente para realizá-la.

2.1.4.1 Associação

A tarefa de associação permite relacionar a ocorrência de um específico conjunto de itens com as ocorrências de outro conjunto de itens. As regras de associação buscam determinar que fatos ocorram simultaneamente com probabilidade razoável ou que itens estão presentes juntos com certa chance (CARVALHO, 2001).

Em outras palavras, as regras de associação reconhecem afinidades entre dados de um subconjunto de informações. Sendo essas afinidades/associações expressas na forma de regras (BAPTISTA; CARVALHO, 2003).

Essas normas caracterizam o quanto a presença de um conjunto de itens nos registros de um banco de dados implica na presença de algum outro conjunto distinto de itens nos mesmos registros (AGRAVAL; SRIKANT, 1994 apud DOMINGUES, 2004).

Esta regra é representada por combinações de itens que ocorrem com determinada frequência em uma determinada base de dados. Uma de suas típicas aplicações é a análise de transações de compra (*market basket analysis*).

2.1.4.2 Classificação

É o meio mais estudado em *Knowledge Discovery in Databasse* (KDD) e tem como meta buscar um conhecimento que venha a ser usado para prever a classe de um registro (AURÉLIO et al., 1999).

É uma técnica que tem por objetivo descobrir um relacionamento entre um atributo meta, pré-definido, e um conjunto de atributos, buscando classificar uma população de registros através da aplicação em um conjunto menor de dados, a fim de desenvolver um modelo de classificação (BAPTISTA; CARVALHO, 2003).

Algumas regras são de fundamental importância no modelo de classificação, como é o caso das regras do tipo SE...ENTÃO..., onde representam uma forma simbólica de classificação e possuem o seguinte formato:

- SE <antecedente> ENTÃO <consequente>

O antecedente é composto por expressões de condição que envolvem atributos do domínio da aplicação existentes na base de dados. Já o consequente tem como composição uma expressão que evidencia algum valor para um atributo meta, descoberto em função dos valores contidos nos atributos que completam o antecedente (ROMÃO, 2002). Logo, as regras de classificação podem ser interpretadas como: SE os atributos preditivos de uma tupla satisfazem as condições no antecedente da regra, ENTÃO a tupla tem a classe indicada no consequente da regra.

Os SVMs são considerados pela comunidade científica, como uma importante técnica usada na tarefa de classificação, devido a sua representação simples, intuitiva e de fácil compreensão.

Para buscar um melhor entendimento sobre os inúmeros algoritmos utilizados em classificação, podemos visualizar a Figura 2.1 que ilustra as relações de funcionamento e utilização de vários métodos e modelos.

Uma base de dados que armazena características de cliente, baseando em históricos de transações anteriores, podem-se classificar estes em categorias para liberação de crédito. Um novo cliente poderá ser classificado em uma das categorias definidas de acordo com suas características.

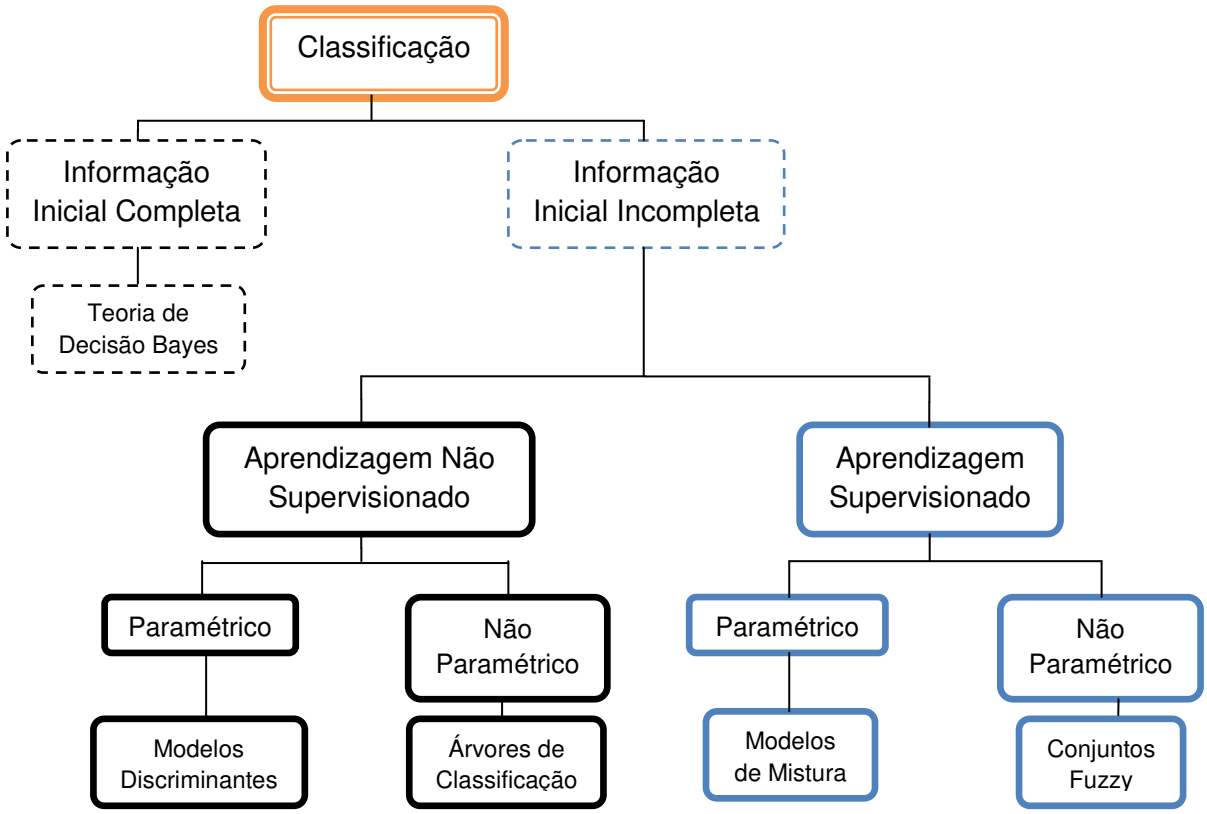


Figura 2.2: - Hierarquia de Classificação (RODRIGUES, 2005)

A classificação de um processo de discriminação de unidades concretas ou abstratas em classes ou categorias, ou de forma abreviada, efeito ou processo de distribuir por classes. Se estas classes estiverem definidas e existir informação sobre a probabilidade de um determinado objeto pertencer a uma classe, entende-se como informação inicial completa, caso contrário, esta informação inicial estará incompleta. Com base no tipo de informação disponível o passo seguinte passa pela utilização e escolha de um método. Os métodos podem ser classificados em paramétricos e não paramétricos (RODRIGUES, 2005).

- Métodos paramétricos: nos métodos paramétricos a distribuição da população tem uma dada forma e as inferências, condicionadas por esse pressuposto, dizem respeito a um ou a vários parâmetros, por exemplo, a Regressão Linear e os Modelos Discriminantes.
- Métodos não paramétricos: nesse método, a forma da distribuição da população não é conhecida e as inferências processam-se em quadro muito menos restrito e muitas vezes não envolvem parâmetros.

A principal distinção em relação ao paradigma de aprendizagem, válido para todo tipo de sistemas com capacidade de adaptação, é a aprendizagem supervisionada e aprendizagem não supervisionada.

2.1.4.3 Agrupamento

É uma técnica que tem como objetivo segmentar os dados formando grupos homogêneos. O agrupamento é aplicado quando ainda não é conhecida nenhuma classe e sua função é produzir uma segmentação do conjunto de registros de entrada de acordo com algum critério estabelecido (SILVA, 2003).

Esta tarefa tem como principal função descobrir classes utilizando a similaridade dos valores de seus atributos como fator de decisão. O agrupamento é um método que procura baseado em medidas de semelhança, definir quantas e quais classes existe em um conjunto de entidades (CARVALHO, 2001).

Outro fato principal tem meta originar classes através de partições da base de dados em conjunto com tuplas. Essa partição é feita agrupando tuplas com valores de atributos parecidos em uma mesma classe. Quando criado as classes, é possível aplicar algoritmos de classificação para produzir regras para as mesmas (PACHECO et al., 1999).

Através dessas funcionalidades de agrupamento é possível subdividir os dados em subconjuntos homogêneos fáceis de descrever e visualizar. Tais dados podem ser exibidos para o usuário em vez de tentar mostrar todos os dados, o que resultaria na perda de padrões embutidos (FAYYAD, 1997 apud ROMÃO, 2002).

O agrupamento pode ser usado, por exemplo, em um banco de dados escolar, relacionando disciplinas e alunos. Onde uma regra do tipo, que a maioria dos alunos inscritos em disciplina de Banco de Dados também estão inscritos em Sistemas Operacionais, isso poderá ser usado pela direção ou secretaria no planejamento do currículo anual, ou adicionar recursos como sala de aula e professores (SCHENATZ, 2005).

2.1.4.4 Previsão de Séries Temporais

A previsão de séries temporais é uma manifestação relativa a sucessos desconhecidos em um futuro determinado. A previsão não constitui um fim em si, mas um meio de fornecer informações e subsídios para uma consequente tomada de decisão, visando atingir determinados objetivos (MORETTIN, 1981 apud MUELLER 1996).

É definida como a classe de fenômenos cujo processo observacional e consequente quantificação numérica gera uma sequência de dados distribuídos no tempo (SOUZA, 1989 apud MUELLER 1996). A natureza de uma série temporal e a estrutura de seu mecanismo gerador está relacionada com o intervalo de ocorrência das observações no tempo (ANDERSON, 1971 apud MUELLER 1996).

Tem como objetivo a realização de inferências sobre as propriedades ou características básicas do mecanismo gerador do processo estocástico das observações da série. Assim, através da abstração de regularidades contidas nos fenômenos observáveis de uma série temporal existe a possibilidade de se construir um modelo matemático como uma representação simplificada da realidade (BARBANCHO1970 apud MUELLER 1996).

Após a formulação do modelo matemático, obtido pela seleção entre as alternativas de classes de modelos identificadas como apropriadas para essa representação e subsequente estimação de seus parâmetros, é possível utilizá-lo para testar alguma hipótese ou teoria a respeito do mecanismo gerador do processo estocástico e realizar a previsão de valores futuros da série temporal (GRANGER1977 apud MUELLER 1996).

2.1.4.5 Técnicas de Mineração de Dados

Existem inúmeras técnicas de mineração e algoritmos que facilita a busca por padrões não conhecidos nos dados. Além disso, ter certo conhecimento sobre essas técnicas facilita muito no momento da escolha de uma delas de acordo com os problemas apresentados (SILVEIRA, 2003).

O Quadro 2.2 é mostrado técnicas, descrição, tarefas associadas a cada uma e alguns dos algoritmos principais.

Quadro 2.2 - Técnicas de KDD e algoritmos

Técnicas	Descrição	Tarefas	Exemplos
Regras de Associação	Estabelece uma correlação estatística entre atributos de dados e conjuntos de dados.	Associação	Apriori, AprioriTid, AprioriHybrid, AIS, SEM e DHP
Árvores de Decisão	Hierarquização dos dados, baseada em estágios de decisão e na separação de classes e subconjuntos.	Classificação e Regressão	CART, CHAID, C5.0, ID-3, QUEST, SLIQ e SPRINT
Raciocínio Baseado em Casos ou MBR	Baseado no método do vizinho mais próxima, combina e compara atributos para estabelecer hierarquia de semelhança.	Classificação e Segmentação	BIRCH, CLARANS e CLIQUE
Algoritmos Genéticos	Métodos gerais de busca e otimização, inspirados na Teoria da Evolução.	Classificação e Segmentação	Algoritmo Genético Simples, Genitor, CHC, GA-Nuggets e GA-PVMINER
Redes Neurais Artificiais	Modelos baseados na fisiologia do cérebro, onde o conhecimento é fruto do mapa das conexões neurais.	Classificação e Segmentação	Perceptron, Rede MLP, Redes de Kohonen, Rede Hopfield, Rede BAM, Rede IAC e Rede LVQ

Fonte: DIAS, 2002.

2.1.5 Pós-Processamento

Nesta fase, é feita a avaliação do processo de descoberta, que tenta melhorar a compreensão e selecionar o conhecimento adquirido que seja mais relevante para o objetivo pretendido.

O objetivo principal desta fase é aprimorar a compreensão do conhecimento encontrado pelo algoritmo minerador, por meio da análise dos dados por um especialista. Muitas vezes, a mineração não gera bons resultados, sendo importante uma nova etapa de enriquecimento, a fim de busca adicionar mais informações de forma que contribuam no processo de descoberta de conhecimento (PACHECO et al., 1999).

Nesta fase, inicia-se com a avaliação dos padrões realmente interessantes, que representem conhecimento útil, seguido da apresentação do conhecimento extraído para o usuário final, através de técnicas de visualização e representação do conhecimento (SILVA, 2003).

2.2 Aprendizado de Máquina

O aprendizado de máquina é uma área da computação que realiza pesquisa no ramo de inteligência artificial, podendo esta ser definida como o campo que utiliza métodos que permitam ao computador obter comportamento inteligente, ou seja, que adquire novos conhecimentos com a experiência, logo o objetivo do aprendizado de máquina é o incremento de técnicas computacionais sobre o aprendizado (MONARD *& BARANAUSKAS 2003a*).

Possuem vários algoritmos de aprendizado de máquina, que apresentam diferentes objetivos, estratégias de aprendizagem e representação do conhecimento. Entretanto, todos eles realizam o aprendizado através de um processo de busca, para encontrar uma generalização aceitável (LUGER & STUBBLEFIELD, 1998). O processo de aprendizagem de um conceito genérico através de um conjunto de padrões fornecidos ao sistema de aprendizado por um processo externo. De acordo com (MONARD *& BARANAUSKAS 2003a*), esse conjunto pode ser dividido em dois subconjuntos:

- Conjunto de treinamento: contém os padrões utilizados para o aprendizado do conceito.
- Conjunto de teste: contém os padrões utilizados para medir o grau de efetividade do conceito aprendido.

O aprendizado de máquina pode ser basicamente dividido em duas categorias: supervisionado e não supervisionado. Tal aprendizado também é conhecido, na literatura, como reconhecimento de padrões (MICHIEL et al., 1994), (KUNCHEVA, 2004).

No aprendizado Supervisionado o algoritmo de aprendizado (indutor) recebe um conjunto de exemplos de treinamento para os quais os rótulos da classe associada são conhecidos. Cada exemplo (instância ou padrão) é descrito por um vetor de valores (atributos) e pelo rótulo de classe associada. O objetivo do indutor é

construir um classificador que possa determinar corretamente a classe de novos exemplos ainda não rotulados.

No aprendizado não supervisionado indutor analisa os exemplos fornecidos e tenta determinar se alguns deles podem ser agrupados de alguma maneira, formando agrupamentos ou clusters. Após a determinação dos agrupamentos, em geral, é necessário uma análise para determinar o que cada agrupamento significa no contexto do problema que está sendo analisado.

2.2.1 Paradigma do Aprendizado

Os classificadores são induzidos a um processo de aprendizagem que através da utilização dos algoritmos de aprendizado de máquina, que foram desenvolvidos ao longo dos anos, cada um com suas vantagens e desvantagens. De acordo com as particularidades de cada um, esses algoritmos podem ser divididos nos seguintes paradigmas: Estatístico, Simbólico, Baseado em Exemplos e Conexionista (MONARD *& BARANAUSKAS 2003a*). A seguir, será apresentada uma breve descrição de cada um deles.

- Paradigma Estatístico: A ideia geral desse paradigma é a utilização de modelos estatísticos para encontrar uma boa aproximação do conceito induzido (MONARD *& BARANAUSKAS 2003a*). A teoria de decisão Bayesiana é uma abordagem estatística fundamental
- Paradigma Simbólico: Os sistemas de aprendizado simbólico buscam aprender desenvolver representações simbólicas de um conceito através da análise de exemplos e contra-exemplos desse conceito. Essas representações estão geralmente na forma de alguma expressão lógica, árvore de decisão, regras de classificação ou rede semântica (MONARD *& BARANAUSKAS 2003a*). Os indutores gerados contribuem para a compreensão dos dados, ao contrário de outros indutores que visam apenas a uma grande precisão. Geralmente, é utilizado em problemas de classificação que utilizam dados nominais, ou seja, descrições que são discretas e sem nenhuma noção natural de similaridade ou ordenação (DUDA et al., 2001).

- Paradigma Baseado em Exemplos: Os algoritmos desse paradigma classificam novos padrões por meio de outros que sejam similares, e cujas classes sejam conhecidas. Em outras palavras, a classe de um novo padrão é definida como sendo a mesma que rotula os padrões mais similares a ele (MONARD *& BARANAUSKAS 2003b*).
- Paradigma Conexionista: A representação dos algoritmos desse paradigma envolve unidades altamente interconectadas, realizando uma metáfora biológica com as conexões neurais do sistema nervoso. As grandes representantes desse paradigma são as Redes Neurais Artificiais (MONARD *& BARANAUSKAS 2003a*).

2.3 Support Verctor Machine (SVM)

A Máquina de Vetores e Suporte (SVMs, do Inglês *Support Vector Machines*) usam uma técnica de aprendizado supervisionado que vem recebendo paulatinamente atenção da comunidade de Aprendizado de Máquina (AM). Muitos resultados da aplicação dessa técnica são comparáveis e em grande parte superiores aos obtidos por outros algoritmos de aprendizado, como as Redes Neurais Artificiais (RNAs). Essa técnica vem sendo usada em diversas aplicações que podem ser encontradas em diversos domínios, como na categorização de textos, na análise de imagens e em Bioinformática, entre outras [Vapnik et al, 1992].

As SVMs são embasadas pela teoria de aprendizado estatístico, desenvolvida por *Vapnik*. Essa teoria estabelece uma série de princípios que devem ser seguidos na obtenção de classificadores com boa generalização, definidos como a sua capacidade de prever corretamente a classe de novos dados do mesmo domínio em que o aprendizado ocorreu. Esta técnica originalmente surgiu para classificação binária, busca a construção de um hiperplano como superfície de decisão, de tal forma que a separação entre exemplos seja máxima, isso considerando padrões linearmente separáveis.

para padrões não linearmente separáveis, busca-se uma função de mapeamento apropriado para tornar o conjunto mapeado linearmente separável. Devido a sua eficiência em trabalhar com dados de alta dimensionalidade é reportada na literatura como uma técnica altamente robusta, muitas vezes

comparada a Redes Neurais. Abaixo são descritas algumas características fundamentais no processo de classificação das SVMs.

- A classificação gerada pelas SVMs muitas das vezes alcançam resultados satisfatório de generalização. Onde sua capacidade de generalização de um classificador é definida por sua eficiência na classificação de dados que não pertençam ao conjunto utilizado em seu treinamento.
- As *Support Vector Machine* são robustas em relação a objetos de grandes dimensionalidade.
- As SVMs possuem uma base teórica bem definida dentro da Matemática e Estatística.
- A aplicação das SVMs resultará na otimização de uma função quadrática, que possui apenas um mínimo global.

2.3.1 Teoria do Aprendizado Estatístico

Seja f um classificador e F o conjunto de todos os classificadores que um determinado algoritmo de AM pode gerar. Esse algoritmo, durante o processo de aprendizado, utiliza um conjunto de treinamento T, composto de n pares (Xi, Yi), para gerar um classificador particular $\hat{f}$ ϵ F. Considere, por exemplo, o conjunto de treinamento da Figura 2.3. O objetivo do processo de aprendizado é encontrar um classificador que separe os dados das classes “círculo” e “triângulo”. As funções ou hipóteses consideradas são ilustradas na Figura 2.3 por meio das bordas, também denominadas fronteiras de decisão, traçadas entre as classes.

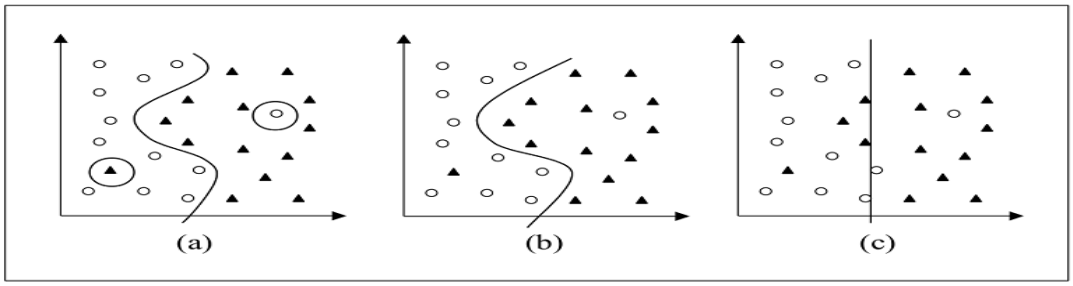


Figura 2.3: conjunto de treinamento binário e três diferentes hipóteses

2.3.2 Hiperplano de Separação

Um *Support Vector Machine* (SVM) estabelece um classificador de acordo com um conjunto de amostras por ele identificado nos exemplos de treinamento, onde esta classificação é aceita. Considerando o exemplo da Figura 2.4, nela existe um conjunto de classificadores lineares que separam duas classes, mas apenas um (em destaque) que maximiza a margem de separação (distância da instância mais próxima ao hiperplano de separação das duas classes em questão).

O hiperplano com margem máxima é chamado de hiperplano ótimo, que será o objeto de busca do treinamento do classificador (GUNN 1998). Na Figura 2.5 (a) mostra um dos possíveis hiperplanos de separação com margem pequena e (b) mostra o hiperplano de separação ótimo com a margem maximizada.

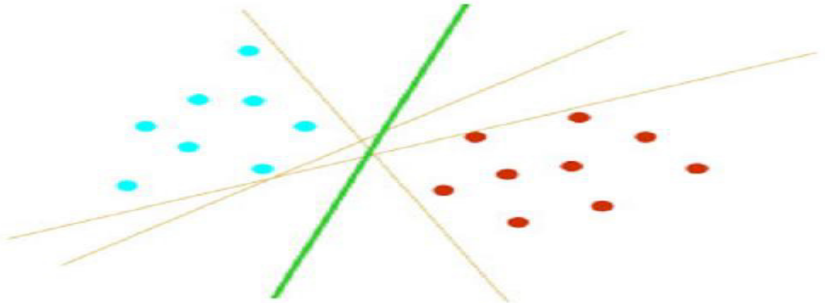


Figura 2.4: Possíveis hiperplanos de separação e hiperplano ótimo

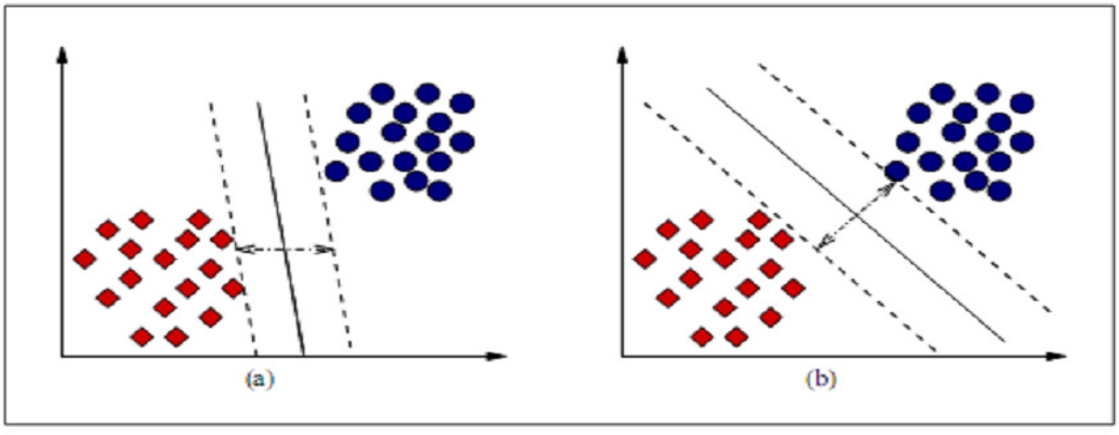


Figura 2.5: (a) Hiperplano com margem pequena. (b) Hiperplano com margem máxima

Seja (x1, y1),... , (xi, yi) , tal que x ∈ Rn e y ∈ {− 1, 1 }, i = 1, ..., N, onde N é o número de instâncias do treinamento, x é o vetor de entrada e y é a classificação

desejada. O objetivo é estimar uma função F: Rn -> {-1 ou 1}, usando os exemplos de treinamento, e aplicá-la nos exemplos de teste, não utilizados anteriormente, com o objetivo de que sejam classificados corretamente.

Quando um classificador perde a capacidade de generalizar ocorre um fenômeno denominado de sobre-ajuste (*overfitting*), onde a complexidade da função obtida é superior a necessidade do problema, além desse, existe outro problema que ocorre de forma contrária, chamado de sub-ajuste (*underfitting*), onde a complexidade da função obtida é inferior a necessidade do problema. A Figura 2.6 lustra a ocorrência desses fenômenos.

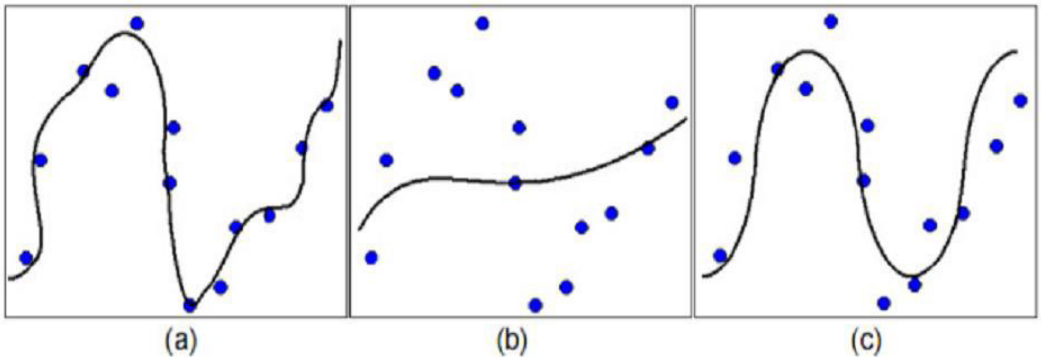


Figura 2.6: (a) sobre ajuste, (b) sub ajuste, (c) função de aproximação.

Logo se nenhuma restrição for atribuída à função pretendida F, mesmo uma função que tenha uma boa taxa de acerto no treinamento, não é garantido que ela adquira bons resultados nos exemplos de teste. Por isso, minimizar o erro no treinamento não implica em um pequeno erro no teste. Uma restrição indispensável, determinada pela teoria da aprendizagem estatística, relaciona o risco esperado da função ao seu risco empírico e a um termo de capacidade. Esse limite, apresentado na equação três, é garantido com probabilidade 1 - θ, em que θ ∈ [0; 1] (LORENA et al. 2007).

$$R(f) \leq R_{emp}(f) + \sqrt{\frac{h(\ln(\frac{2n}{h}) + 1) - \ln(\frac{\theta}{4})}{l}} \qquad (2.1)$$

Na equação a cima, h denota a dimensão VC (*Vapnik-Chervonenkis*) da classe de funções a qual f pertence e n representa a quantidade de instâncias de treinamento. A dimensão VC, mede a capacidade das funções contidas em F. Quanto maior for o valor de VC, mais complexas são as funções de classificação.

Para ficar mais claro o conceito de capacidade, considere três dados apresentados na Figura 2.7. Note que em qualquer disposição arbitrária dos dados, é possível definir uma reta que separe as classes dos dados.

R -> Risco esperado – é o erro do kernel em questão.

R_{emp} -> Risco (erro) do kernel na amostra.

h -> Dimensão VC – parâmetro que representa a capacidade do kernel.

n -> Número de itens de amostra.

l -> Probabilidade do risco esperado ser ultrapassado.

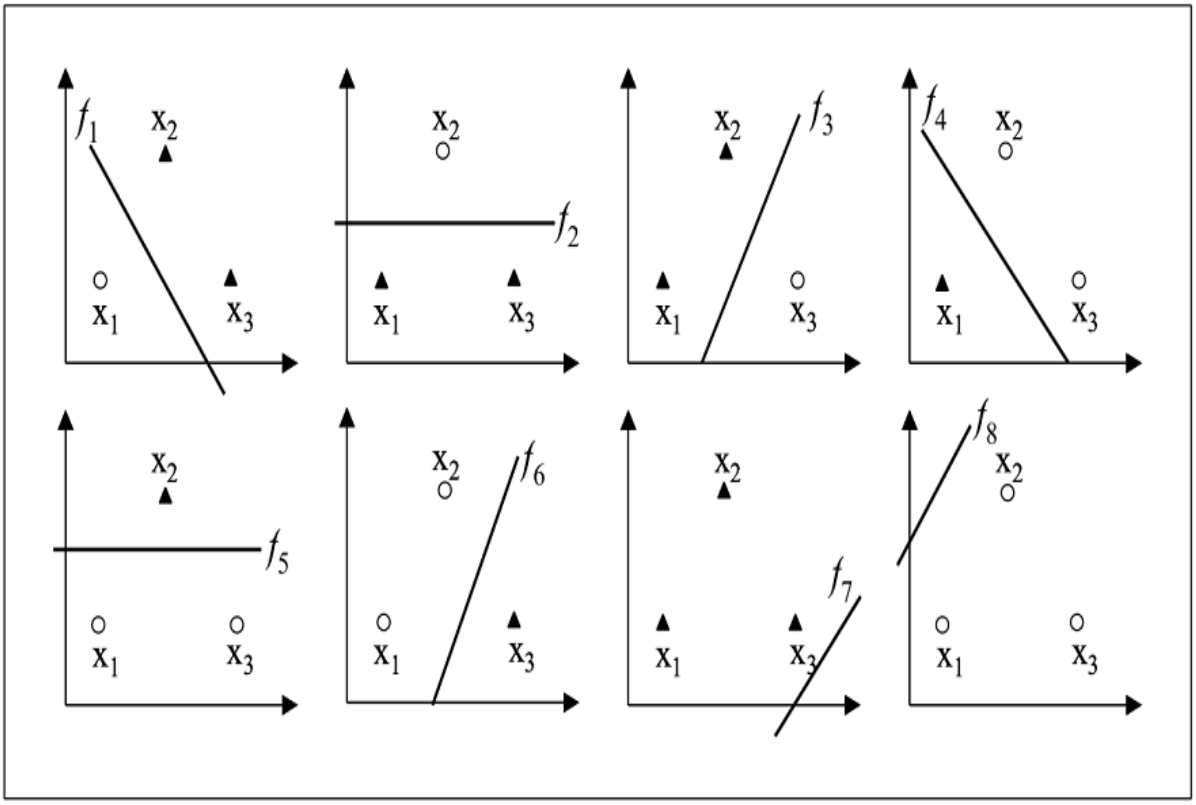


Figura 2.7: Separação dos dados usando uma reta no R²

2.3.3 Função do Kernel

As funções de *kernel* ou Núcleos têm a finalidade de determinar os vetores de características de entrada em um espaço de características de alta dimensão para uma classificação de problemas que estão nos espaços não linearmente separáveis.

Isso é determinado, pois à medida que se aumenta a área da dimensão do problema, aumenta também a probabilidade desse problema se tornar linearmente

separável em relação a um espaço de baixa dimensão. Entretanto, para obter uma boa distribuição para esse tipo de problema é necessário um conjunto de treinamento com um elevado número de instâncias (GONÇALVES 2010).

Em seguida é mostrada a transformação de um domínio não linearmente separável e um linearmente separável através do aumenta da dimensão, onde é feito por uma função *Kernel* F(x)

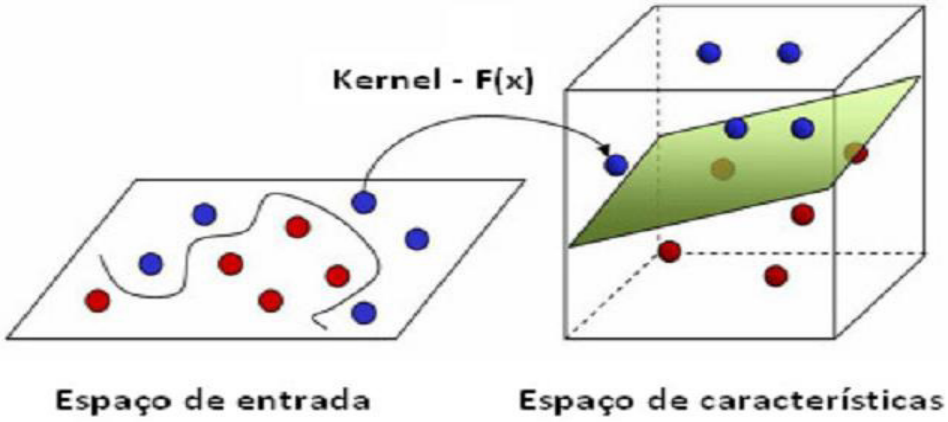


Figura 2.8: Transformação: Problema não linearmente separável em um problema linearmente separável (RABELO 2008)

Os *Kernels* são conhecidos como *Kernels* de Mercer. Há vários tipos de *Kernels* que podem ser usados, porém a utilidade dos *Kernels* está relacionada, na simplicidade de cálculo e na capacidade de representar espaços muito abstratos. Alguns dos *Kernels* mais utilizados são os polinomiais, os Gaussianos ou RBF (*Radial-Basis Function*), o Linear e os *Sigmoidais*, como é mostrado na Tabela 2.3.

Quadro 2.3: *Kernels* mais usados.

Tipo de Kernel	Função K(xi,xj) correspondente	Comentários
Polinomial	$\left(x_i{}^T . x_j + 1\right)^p$	A potência *p* deve ser especificada pelo usuário.
Função básica radial (RBF)	$e^{-\gamma\|\|x-y\|\|^2}$	O parâmetro γ é definido pelo usuário.
Gaussiano	$\exp\left(-\frac{1}{2\sigma^2}\|\|x_i - x_j\|\|^2\right)$	A amplitude σ^2 é especificada pelo usuário.
Sigmoidal	$\tanh\left(\beta_0 x_i . x_j + \beta_1\right)$	Utilizado somente para alguns valores de β_0 e β_1.
Linear	$x^T y$	

2.3.4 Casos não Separáveis

Em alguns casos o *Kernel* não tem a capacidade para transformar o espaço dos dados num espaço de dimensionalidade suficiente para que a separação seja linear, levando a que os dados possam sofrer de alguns defeitos. Assim, é permitida uma margem para alguns dados mal classificados (designada de soft margin). O controle desta margem é feito pelo utilizador com recurso a um parâmetro C. Este parâmetro tem como efeito, a limitação dos valores que os multiplicadores lagrangeanos podem tomar, restringindo-os a $0 \leq \alpha_i^0 \leq C$.

2.3.5 SVMs Lineares

As SVMs lineares surgiram pela colocação direta dos resultados fornecidos pela TAE. Nesta seção é apresentado o uso de SVMs na obtenção de fronteiras lineares para a separação de dados pertencentes a duas classes. A primeira formulação, mais simples, lida com problemas linearmente separáveis.

2.3.5.1 SVMs com Margens Rígidas

As SVMs lineares com margens rígidas definem fronteiras lineares a partir de dados linearmente separáveis. Onde os classificadores que separam os dados por meio de um hiperplano são denominados lineares. A equação 2.2 de um hiperplano é apresentada abaixo, em que w e x, é o produto escalar entre os vetores w e x, w ϵ X é o vetor normal ao hiperplano descrito e $\frac{b}{\|w\|}$ corresponde à distância do hiperplano em relação à origem, com b ϵ R.

$$f(x) = w.x + b = 0 \tag{2.2}$$

$$g(x) = \text{sgn}(f(x)) = \begin{Bmatrix} +1 se w.x + b > 0 \\ -1 se w.x + b < 0 \end{Bmatrix} \tag{2.3}$$

Restrição:

$$y_i[w^T.\phi(x_i) + b] \geq 1 - \xi_i \tag{2.4}$$

Na equação 2.4 *w* representa o vetor de pontos perpendicular no hiperplano de separação, b > 0 é um parâmetro escolhido pelo usuário que corresponde a penalidade do erro e os ξ_i*'s* são variáveis de folga que penalizam os erros de treinamento.

Para um dado vetor w e um bias b, a separação entre o hiperplano definido na Equação acima e o ponto de dado mais próximo é denominada a margem de separação, representada por ρ (HAYKIN, 2001). Ou seja, sempre que for possível obter um ρ > 0, existirão infinitos hiperplanos entre as classes, como na Figura 2.5.

2.3.5.2 SVMs com Margens Suaves

Na classificação de uma Máquina de Vetores e Suporte não linearmente separável, tem-se um conjunto de tuplas que não é possível separar as classes por meio de um hiperplano sem encontrar erros de classificação.

Logo, deseja-se buscar um hiperplano ótimo que minimize a probabilidade de erro de classificação (HAYKIN, 2001). Por isso, admite-se que algumas informações possam violar a restrição estabelecida na Equação de restrição. Isso é feito com a introdução de variáveis de folga ɛi para todo i = 1, 2,...,n. Estas variáveis relaxam as restrições impostas ao problema de otimização.

Esse procedimento busca suavizar as margens do classificador linear, deixando que alguns dados estejam entre os vetores de suporte e também a ocorrência de alguns erros de classificação (LORENA; CARVALHO, 2007).

Na Figura 2.9(a) o ponto x_i está dentro da área de separação e no lado correto. Na Figura 2.9(b) o ponto x_i encontra-se dentro da região de separação, porém no lado incorreto da superfície de decisão. Já na Figura 2.9(c) o ponto x_i encontra-se fora da região de separação e no lado incorreto da superfície de decisão (SEMOLINI, 2002).

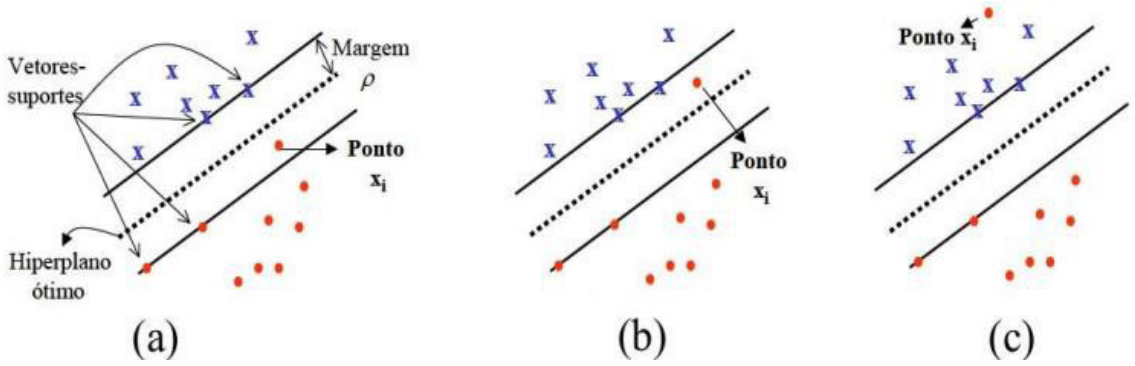


Figura 2.9: Hiperplanos ótimos para padrões não linearmente separáveis (SEMOLINE 2002)

2.3.6 SVMs Não Lineares

Várias aplicações utilizam tuplas que não são linearmente separáveis. A Máquina de Vetores e suporte (SVM) tem a capacidade de aprender em espaços não lineares pelo mapeamento dos dados para um espaço de características onde eles podem ser efetivamente separados, como mostra a Figura 2.10.

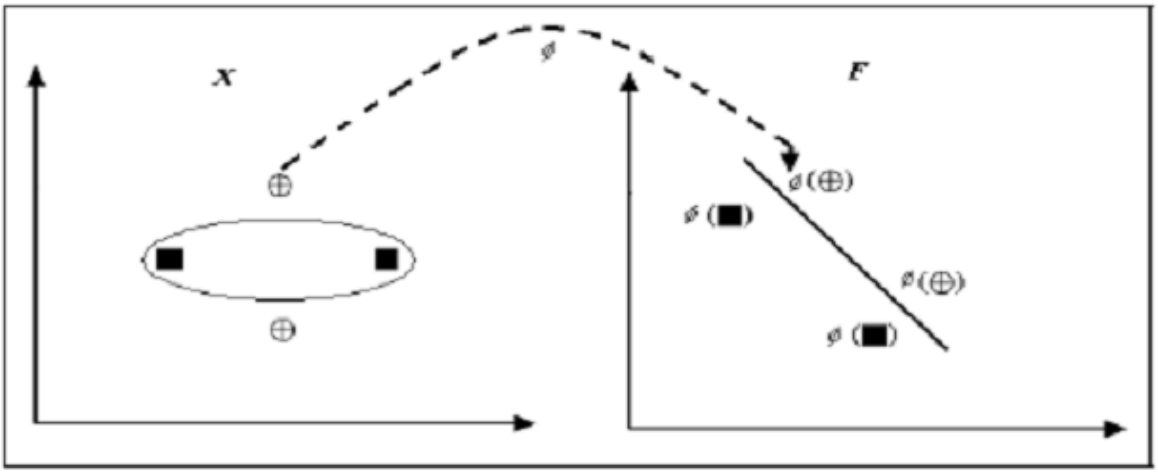


Figura 2.10: Representação de mudança de um espaço de entrada bidimensional, para um espaço de características (SEMOLINI 2002).

A realização da transposição dos espaços é feita através do uso de uma função chamada Kernel ou também de Núcleo. Os dados originais ficam no espaço de entrada. Quando se faz uma função ø: $R^n \rightarrow R^p$, com $p > n$, os dados são mapeados para um espaço de característica. Esta mudança é representada na seguinte equação

$$k(x, \dots x_n) \rightarrow \Phi(x) = (\Phi(x_i), \dots \Phi(x_n)). \quad (2.5)$$

2.3.7 Estratégia de Separação Multiclasses

PLATT et al. (2000) dizem que um problema de classificação Multiclasse, especialmente para a Máquina de Vetores e Suporte (SVM) não apresenta uma solução fácil e a construção de SVMs *Multiclasses* é ainda um problema de pesquisa não resolvido. Para problemas que possuem mais de duas classes, o conjunto de dados de treinamento deve ser combinado para formar problemas de duas classes (BISOGNIN, 2007). A seguir são descritos os dois principais métodos:

2.4.7.1 Um Contra Um

Este método é simples e eficiente para a solução de vários problemas Multiclasses. Vamos supor que um problema com n classes, para cada par dessas n classes é criado um classificador binário. Onde cada classificador é construído

utilizando elementos das duas classes envolvidas, obtendo um total de n (n - 1) / 2 classificadores, como ilustra a Figura 2.11.

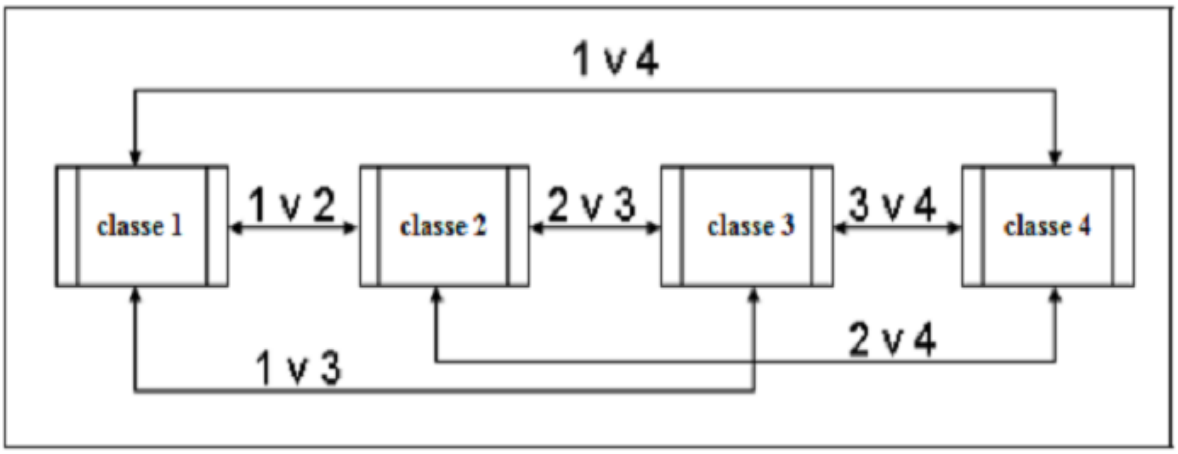


Figura 2.11: Representação do método Um Contra Um (BISOGNIN, 2007)

2.4.7.2 Um Contra Todos

Supondo que um problema tem n classes, logo este método busca particionar estas n classes em dois grupos, onde um grupo é formado por uma classe e o outro é formado pelas classes restantes. Um classificador binário é treinado para esses dois grupos e este procedimento é repetido para cada uma das n classes.

Nesse método a uma grande vantagem que é redução de classificadores, comparado ao método Um Contra Um, o que torna esta classificação mais rápida em casos de poucas classes. Uma desvantagem é que cada classificador utiliza todas as classes, sendo assim o desempenho depende do número de classes.

3. ESTUDO DE CASO

Faz-se neste capítulo, um estudo de caso, cujo objetivo é explicar o uso do algoritmo SVM (*Support Vector Machine*). Para tanto, serão utilizadas as informações da base de dados que está contida na base do Software *Clementine* chamada de cell_samples.data, e disponível na UCI Repositório (Assunção e Newman, 2007) que é um conjunto de dados que contém as características de um certo número de amostras de células extraídas de pacientes que se acreditava estar em risco de desenvolver câncer. Desta forma, as seções seguintes descrevem em detalhes a utilização do mesmo.

3.1 Descrição do Contexto

Um médico pesquisador WIlliam H. Wolberg obteve um conjunto de dados que contém as características de certo número de amostras de células humanas extraídas de pacientes que se acreditava está em risco de desenvolver câncer.

Nesta análise os dados originais mostram que muitas das características deferiram significativamente entre amostras benignas e malignas. O objetivo é desenvolver um modelo SVM que use os valores destas características de células de pacientes para gerar um modelo de classificação que determine se suas amostras são benignas ou malignas.

Este estudo de caso usa o fluxo de dados, disponível como exemplo de uso do SVM na ferramenta Clementine. O Clementine é uma ferramenta de Mineração de Dados que permite rapidamente desenvolver modelos preditivos que utilizam conhecimento de negócio e implantá-los em operações de negócios para melhorar a tomada de decisão. Ele foi concebido em torno do padrão da indústria, Clementine apoia o processo de MD inteiro, a partir de dados de melhores resultados de negócios. A ferramenta possui três versões, são elas:

- Cliente Clementine. É uma versão funcional completa do produto que é instalado e executado no computador desktop do usuário. Ele pode ser executado em modo local como um produto independente ou em modo distribuído junto com Clementine Server para melhorar o desempenho em grandes conjuntos de dados.

- Servidor *Clementine*. É executado continuamente no modo de análise distribuída juntamente com uma ou mais instalações do cliente Clementine, proporcionando desempenho superior em grandes conjuntos de dados, porque executa no servidor sem download de dados para o computador cliente. Clementine Server também fornece suporte para SQL, otimização de lote, modelo de processo e recursos em bancos de dados, oferecendo mais benefícios no desempenho e automação.
- Lote *Clementine*. É uma versão especial do cliente que é executado em modo de lote único, fornecendo suporte para as capacidades completas de análise de dados sem acesso à interface de usuário regular. Isso permite que as tarefas de longa duração ou repetitivas podem ser realizadas sem a intervenção do usuário e sem a presença da interface do usuário na tela. Ao contrário do Cliente Clementine, que pode ser executado como um produto autônomo, Lote Clementine deve ser licenciado e usado somente em combinação com o servidor Clementine.

O exemplo é baseado em um conjunto de dados que está disponível ao público de pesquisadores da área de Aprendizado de Máquina no UCI *Machine Learning Repository* (Assunção e Newman, 2007). O conjunto de dados é constituído por várias centenas de células humanas.

A UCI *Machine Learning Repository* é uma coleção de bancos de dados, teorias de domínio, e geradores de dados que são utilizados pela comunidade de aprendizado de máquina para a análise empírica dos algoritmos de aprendizagem de máquina. O repositório foi criado em 1987 por *David Apha* e colegas estudantes de pós-graduação na Universidade da Califórnia em Irvine (LORENA; CARVALHO, 2007).

Desde então, tem sido amplamente utilizado por estudantes, educadores e pesquisadores de todo o mundo como uma fonte primária de conjuntos de dados de aprendizado de máquina. Como uma indicação da importância do repositório, ele foi citado mais de 1000 vezes, tornando-se um dos 100 mais citadas *papers* da comunidade científica. A versão atual do site foi concebido em 2007 por *Arthur Assunção e Newman David*, e este projeto é em colaboração com a Universidade de *Massachusetts Amherst.*

3.2 Passos seguidos na Mineração dos Dados

Após a escolha da base de dados a ser utilizada em nosso estudo de caso, fizemos os seguintes passos:

- Determinar as variáveis de interesse: a fim de escolher aquelas que são mais importantes para classificar os registros da base de dados.
- Aplicar o algoritmo SVM: para que ele gere a classificação das variáveis mais importantes na determinação de casos de câncer maligno ou benigno.
- Analisar os resultados: para verificar o que foi encontrado, levantar discussões e avaliar as implicações.

3.3 Determinação das Variáveis de Interesse

A base de dados apresenta 11 variáveis, como uniformidade do tamanho da célula, mitoses, tamanho da célula epitelial única entre outras. Primeiro, escolhemos a variável dependente que é aquela que é o alvo do nosso estudo, onde usaremos as outras variáveis chamadas de independentes para classificar o grau de importância destas variáveis sobre a variável dependente. Escolhemos a variável *class* que é aquela que possui a classificação da célula em benigna ou maligna. Sendo assim a variável *class* possui as classes mostradas no Quadro 3.1:

Quadro 3.1 - Variável dependente e suas classes

Variável dependente	Classes
Class	Benigna
	Maligna

Após a escolha da variável dependente, o passo seguinte é determinarmos as variáveis independents. Como a nossa intenção é classificar os registros de pacientes que têm uma grande probabilidade de adquirir algum tipo de câncer, para isso serão mostradas no Quadro 3.2 as variáveis independentes disponíveis na base de dados.

Quadro 3.2: Variáveis Independentes

Variável Independentes	Classes
Clump	Espessura da Célula
UnifSize	Tamanho da Célula
UnifShape	Formato da Célula
MargAdh	Margem da Célula
SingEpiSize	Tamanho da Célula Epitelial
BareNuc	Ausência de Núcleo
BlandChrom	Tipo de Cromatina
NormCucl	Nucléolos Normais
Mit	Mitose

3.4 Análise e Interpretação dos Dados

Nessa etapa, utilizamos o *Clementine*, como explicado na Seção 3.1, na versão 12.0, para a execução do SVM. Nessa ferramenta, todas as etapas são feitas através de nó, como por exemplo, carregar a base de dados para análise. No primeiro nó carregamos a base de dados, o nó utilizado nessa etapa é de acordo com o formato do arquivo da base de dados.

O segundo nó é o *Type,* que lista todas as variáveis presentes na base de dados, nele determinamos a variável dependente e as variáveis independentes. O terceiro é o nó de modelagem, no nosso caso o nó SVM (*Support Vector Machine*). Esses três nós são conectados em sequência, onde o primeiro se conecta com o segundo e este ao terceiro nó, isso é feito, pois a saída do primeiro nó é entrada para o segundo (nó *Type*) e a saída deste último é entrada para o terceiro (nó SVM). Ao executar o nó que representa o algoritmo SVM, ele gera outro nó com a Classificação encontrada. Além do método SVM, o *Clementine* possui outros algoritmos de classificação de dados, como o *CHAID, Neural Network, CART, QUEST* e etc(LORENA; CARVALHO, 2007).

Para a construção dos resultados a fim de fazermos a analise deles, fizemos as seguintes etapas.

Primeiro acrescentamos o nó para importar a base de dados (Figura 3.1).

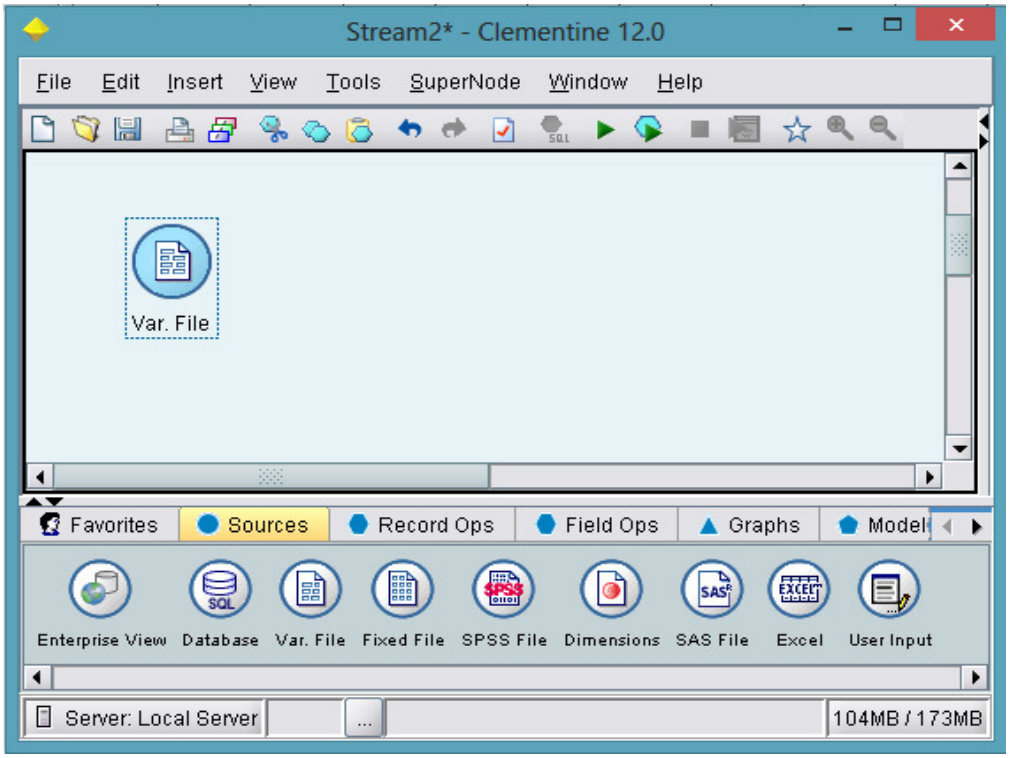


Figura 3.1: Adicionando nó *File*

Nele carregamos a base de dados com as informações das amostras dos pacientes no nó *File*, como podemos observar na Figura 3.2.

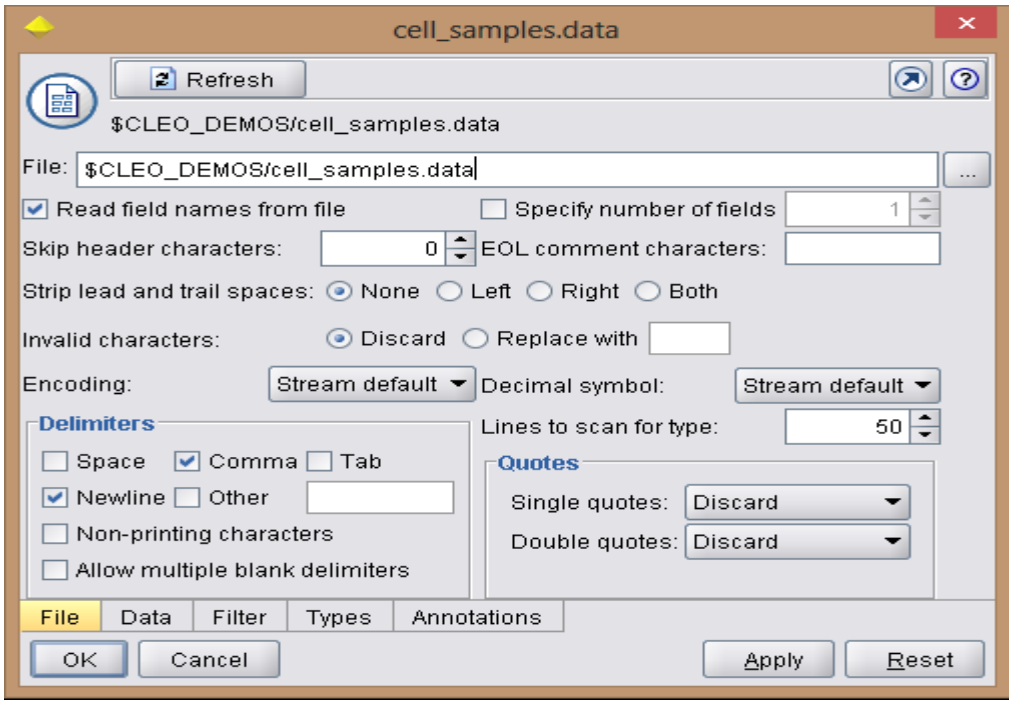


Figura 3.2: Importando a Base de Dados

Depois acrescentamos o nó *Type* (Figura 3.3), que lista todas as variáveis, usamos ele para separar a variável dependente das variáveis independentes.

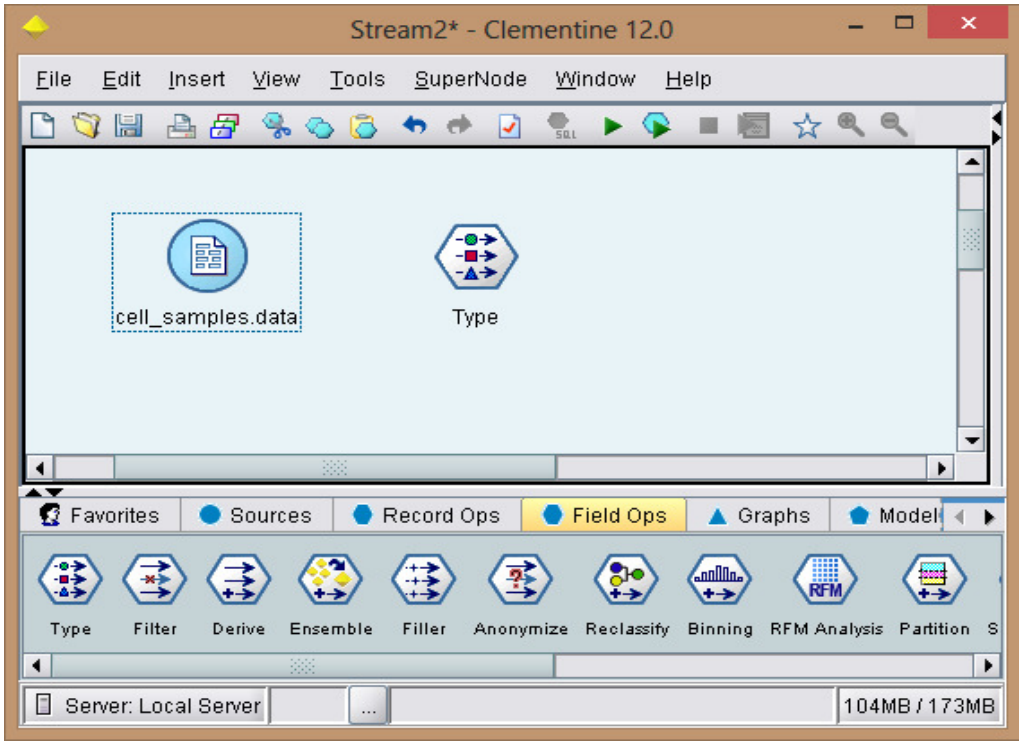


Figura 3.3: Acrescentando o nó *Type*

Nessa etapa será adicionado o nó SVM (Support Vector Machine) (Figura 3.4), que quando executado mostrará a classificação das amostras encontrada pelo algoritmo.

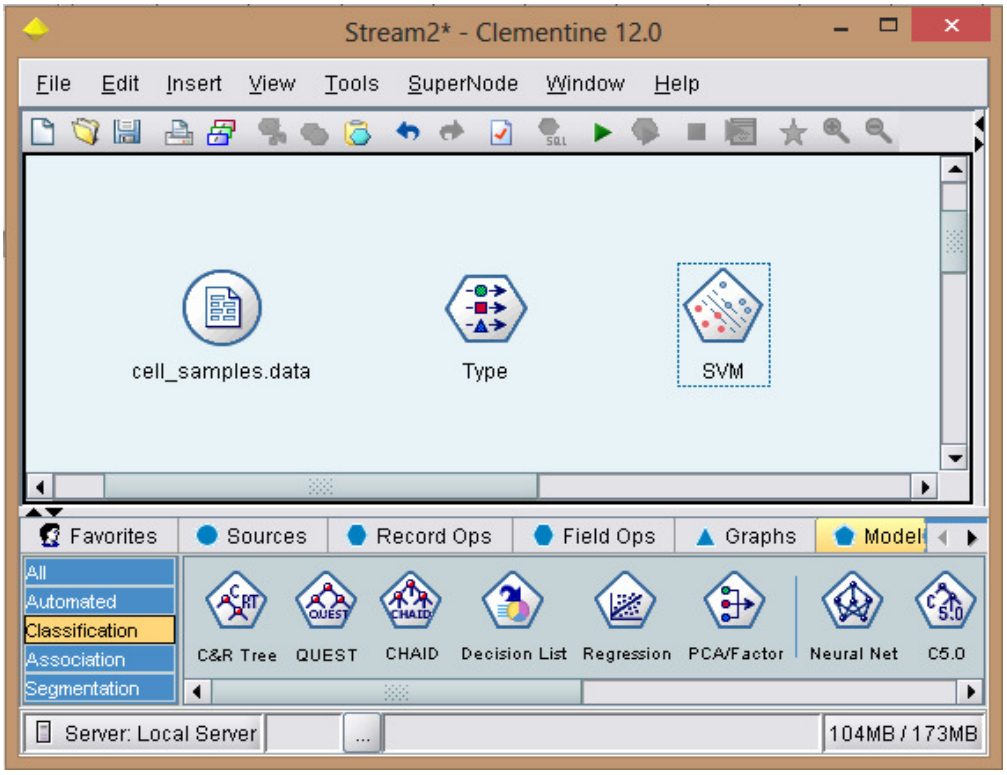


Figura 3.4: Acrescentando o nó SVM

Agora, conectamos o nó *File* ao *Type* e este ao SVM, como na Figura 3.5.

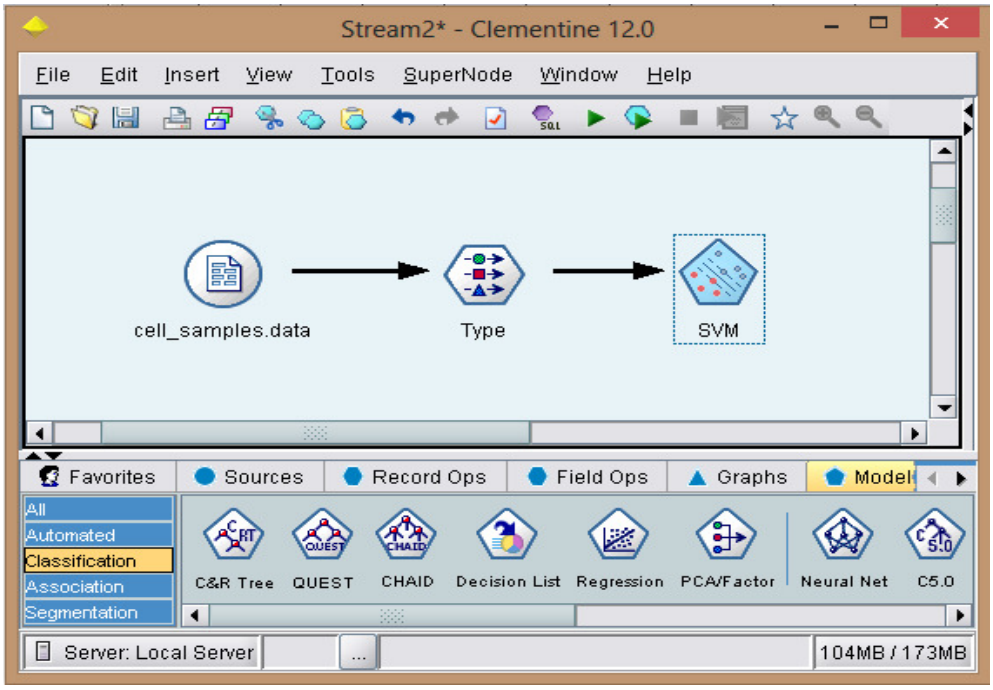


Figura 3.5: Conectando os nós para análise

O *Type é* o nó que possui as variáveis selecionadas, nele selecionamos qual será a variável dependente e as independentes. Para escolhermos a variável dependente mudamos sua direção para saída (Out), como indicado na Figura 3.6, pois ela vai ser entrada para o nó SVM, no caso a variável *class*, o tipo de gravidade da amostra dos pacientes se é benigna ou maligna, as demais variáveis são as independentes.

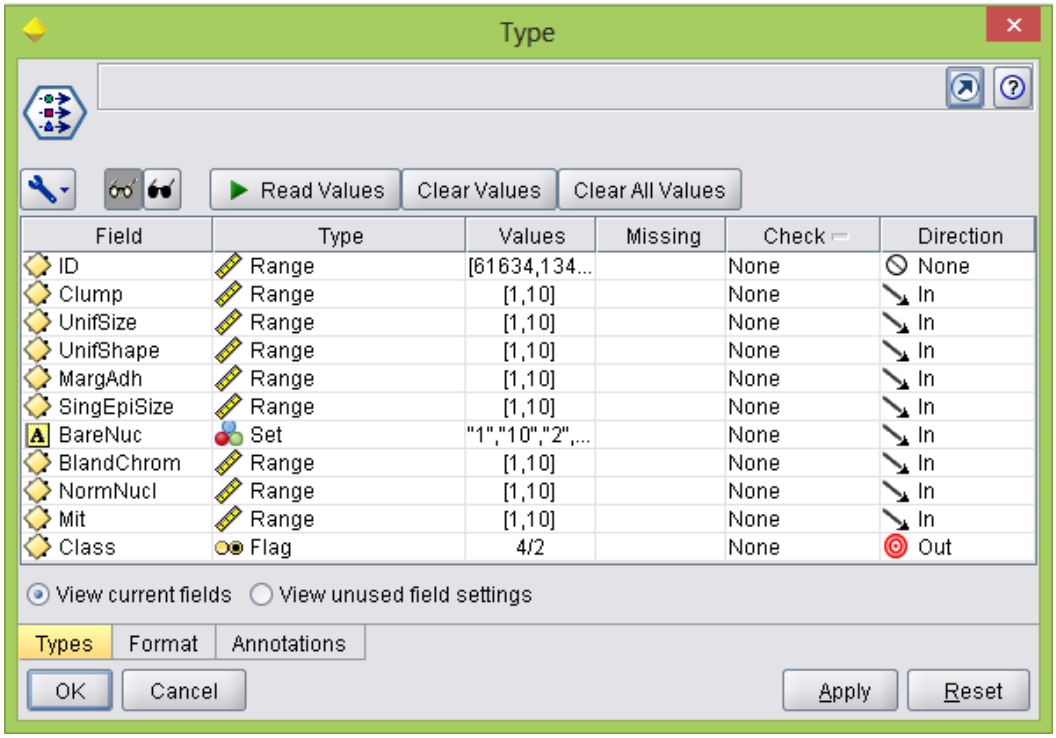


Figura 3.6: Conectando os nós para análise

Na Figura 3.7 será mostrado de forma geral o uso dos *Kernels* implementados no SVM do *Software Clementine* para analisarmos os resultados gerados pelos diferentes núcleos, sendo eles: RBF(*Radial-Basis Function*) ou também conhecido como *Gausiano, Polynomial, linear e o Sigmoid*.

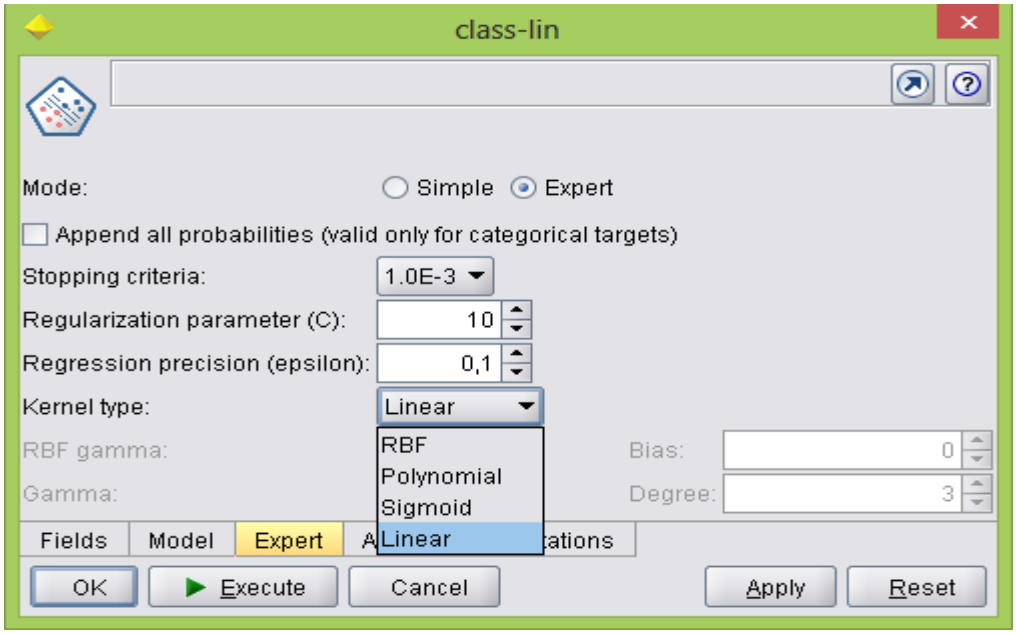


Figura 3.7: Seleção do *Kernel* do SVM

Na Figura 3.8 mostramos como ficou a disposição dos nós para cada um dos *Kernels* do SVM selecionado, onde foi acrescentado um nó de análise nos respectivos resultados dos núcleos.

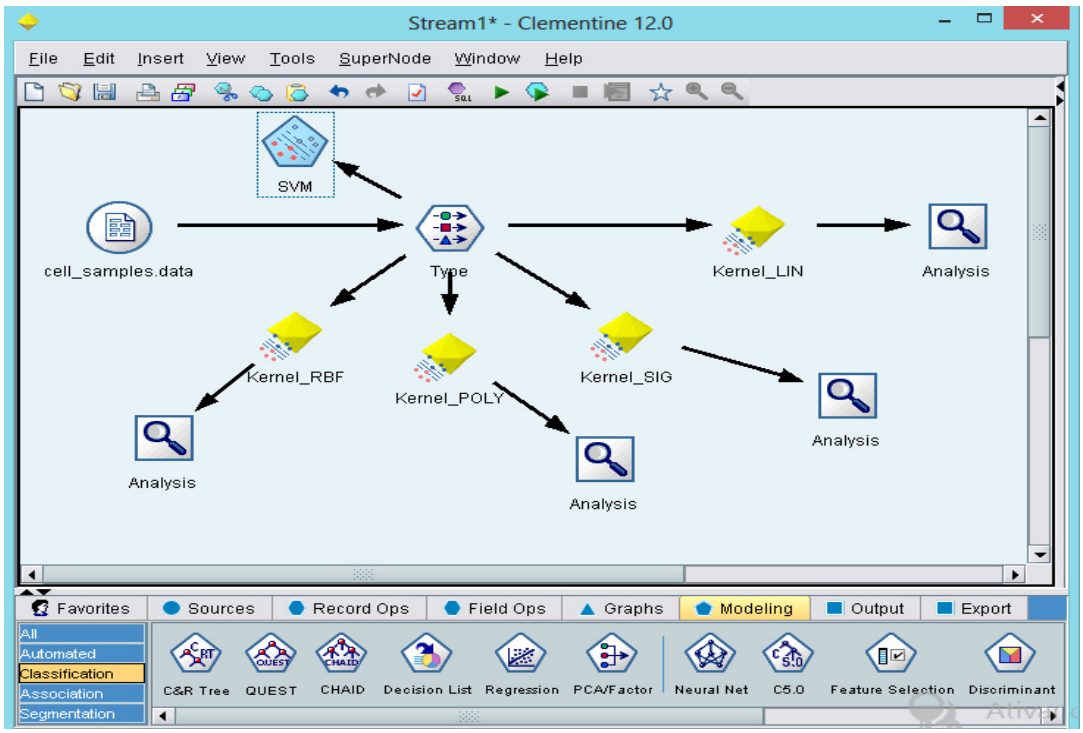


Figura 3.8: Kernels analizados pelo Clementine

Em seguida, executamos o algoritmo SVM para cada um dos núcleos selecionados para realizar a classificação da base de dados utilizada no nosso estudo de caso.

Para o *Kernel Sigmoid* obtivemos como resultado o gráfico mostrado na Figura 3.9, onde ele classificou como variável mais importante a *BareNuc*, ou seja, considerou apenas essa variável para analisar as amostras e determinar se o paciente poderá adquirir algum tipo de câncer.

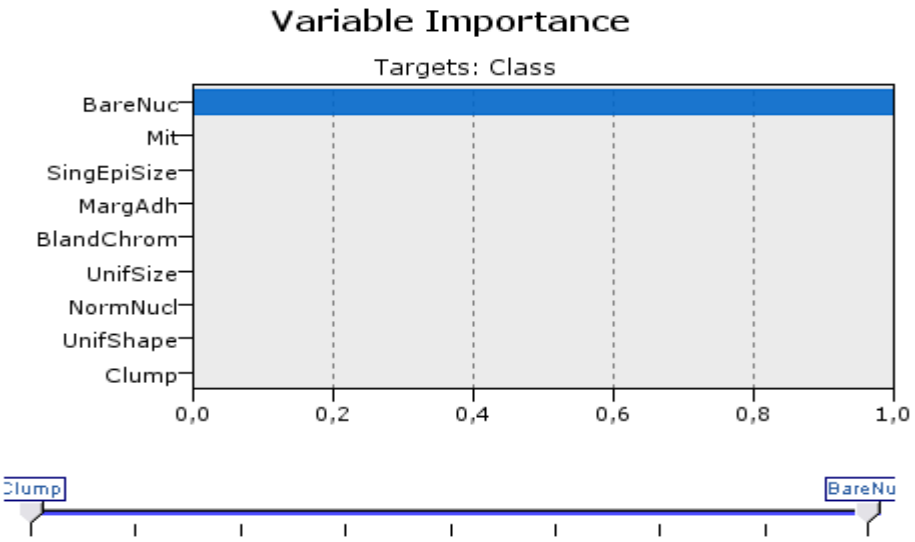


Figura 3.9: Resultado da análise feita pelo *kernel* Sigmoid

Através do nó de analise podemos avaliar o resultado encontrado pelo *Kernel sigmoid*, onde classificou as amostras de pacientes em benigna e maligna de forma correta em 83,55 %, ou seja, do total de 699 amostras ele classificou de forma correta 584 e de forma incorreta 115 que representa 16,45% do total de registros, como podemos observar na Figura 3.10.

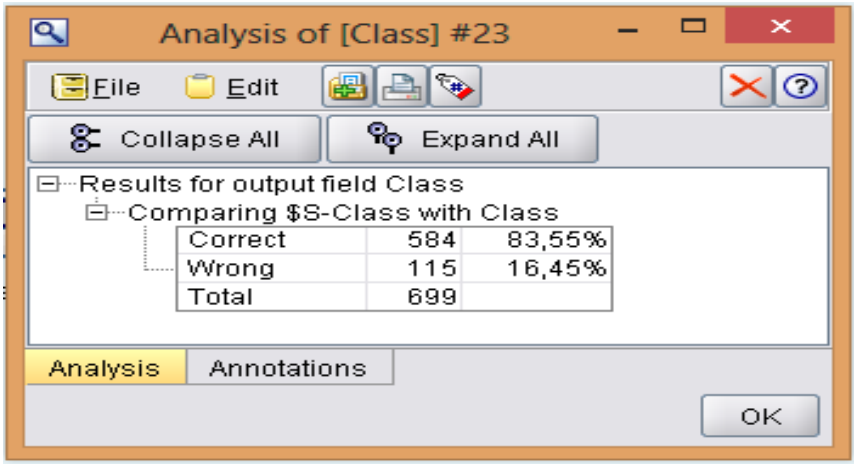


Figura 3.10: Classificação dos resultados em corretos e errados do *Kernel Sigmoid*

Na Figura 3.11 é mostrado o resultado obtido com o uso do *Kernel Linear,* onde ele classificou como variáveis mais importantes em ordem decrescente a *BareNuc* (Ausência de Núcleo) que correspondeu a 31,8 %, a *Clump* (Espessura da Célula) a 23%, a *UnifShap* (Formato da Célula) com 14,7% e a *BlandChrom* (Tipo de Cromatina) com 13,7%, desprezando algumas variáveis como a *UnifSize* (Tamanho da Célula).

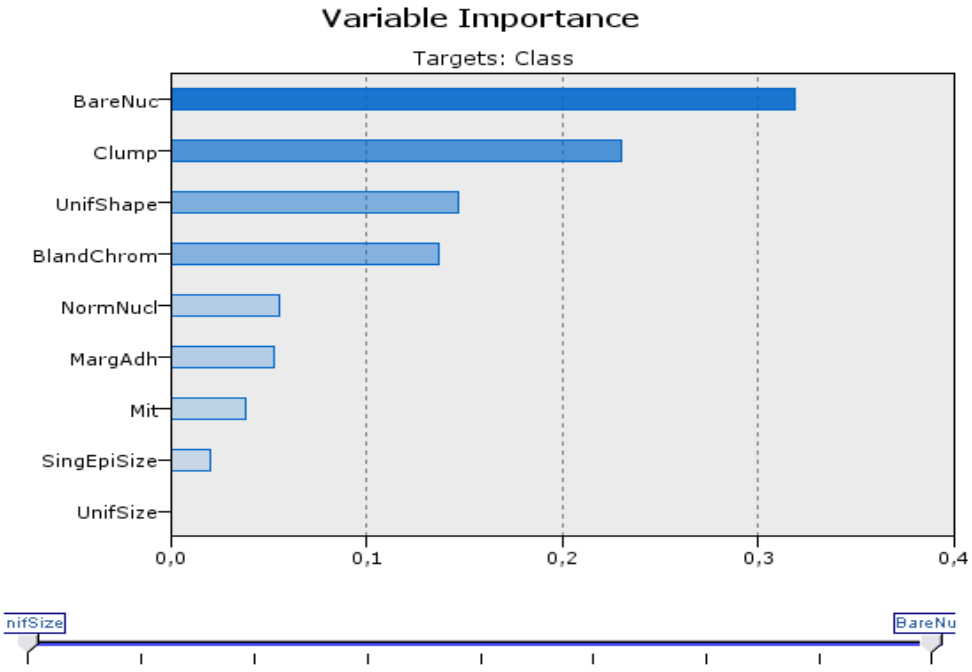


Figura 3.11: Resultado da análise feita pelo *kernel Linear*

Como foi feito no *Kernel Sigmoid*, também fizemos a adição de um nó de analise para podermos avaliar o resultado encontrado pelo Kernel *Linear*, onde

obteve como classificação correta em 97% das amostras de pacientes para determinar se tais são benignas ou malignas, ou seja, do total de 699 amostras ele classificou de forma correta 678 e de forma errada 21 correspondendo a 3% do total de registros, como podemos observar na Figura 3.12.

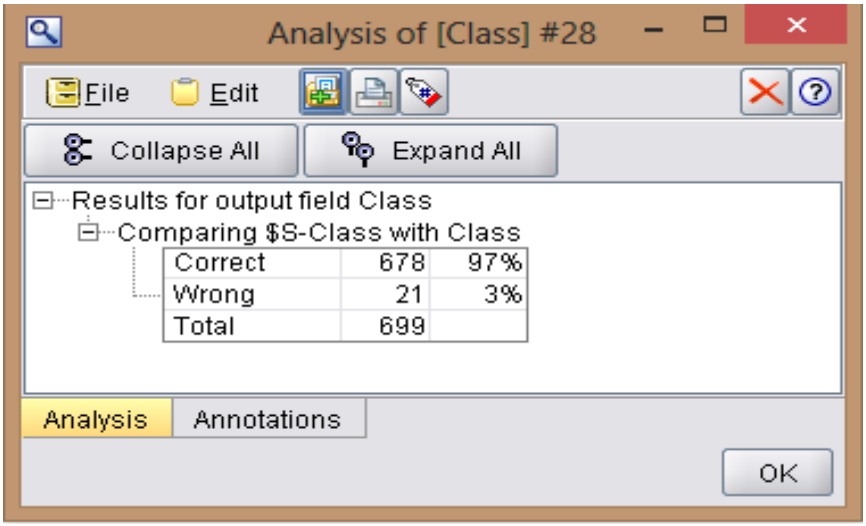


Figura 3.12: Classificação dos resultados em corretos e errados do *Kernel Linear*

Na Figura 3.13 é mostrado o resultado obtido com o uso do *Kernel RBF,* onde ele classificou como variáveis mais importantes em ordem decrescente a *BareNuc* (Ausência de Núcleo) que correspondeu a 39,7 %, a *UnifShap* (Formato da Célula) com 16,4% a *Clump* (Espessura da Célula) a 16,3%, e a *BlandChrom* (Tipo de Cromatina) com 9,7%, como pode-se observar na Figura seguinte, o *Kernel RBF* considerou todas as variáveis independentes, tendo a *SingEpiSize* (Tamanho da Célula Epitelial) a menos significativa correspondendo a 1,8%.

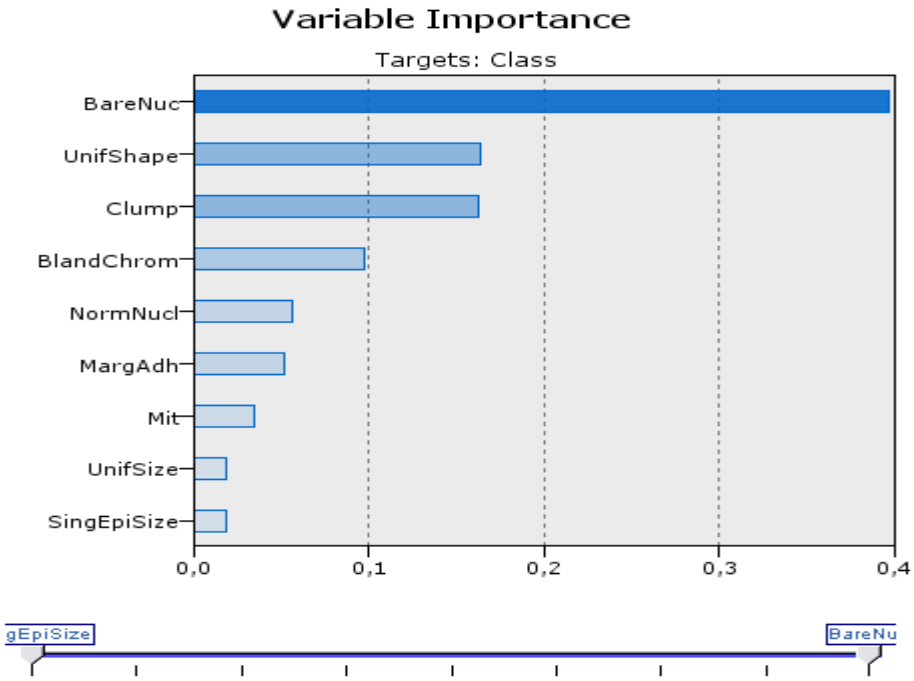


Figura 3.13: Resultado da análise feita pelo *kernel RBF*

Como foi feito nos *Kernel Sigmoid* e Linear, também fizemos a adição de um nó de analise para podermos avaliar o resultado encontrado pelo *Kernel RBF*, tendo como classificação correta em 97,85% das amostras de pacientes para determinar se suas células são benignas ou malignas, o que corresponde a 684 de avaliação correta e de 15 avaliações incorretas correspondendo a 2,15% do total de 699 registros, como podemos observar na Figura 3.14.

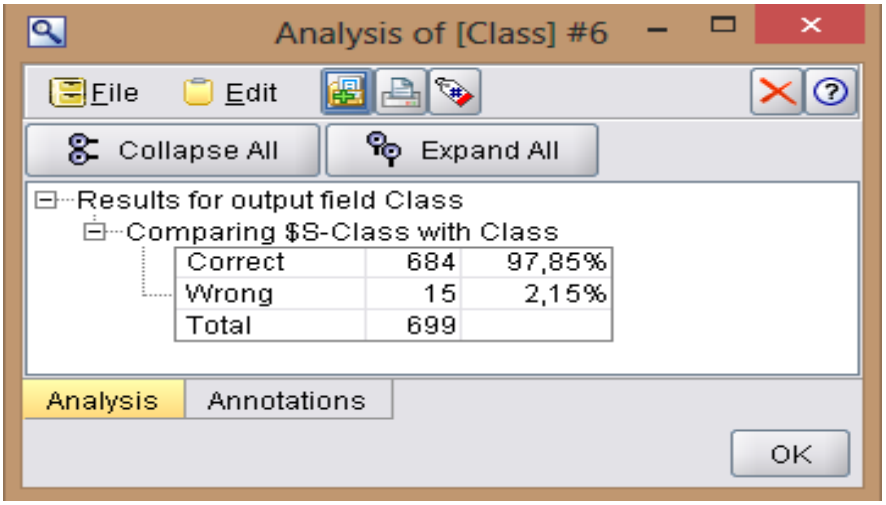


Figura 3.14: Classificação dos resultados em corretos e errados do *Kernel RBF*

Na avaliação das amostras usando o *Kernel Polynomia* como será mostrado na Figura 3.15, observer-se-á que a classificação das variáveis independentes correspondeu em uma maior homogeneidade de avaliação, onde ele classificou como variáveis mais importantes em ordem decrescente a *BareNuc* (Ausência de Núcleo) que correspondeu a 28,9 %, e a *BlandChrom* (Tipo de Cromatina) com 18,8%, a *MargAdh* (Margem da Célula) com 17,5%, a *Clump* (Espessura da Célula) a 12,1%, a *UnifShap* (Formato da Célula) com 9,7%, na análise feita pelo *Kernel Polynomial* observou-se a consideração de todas as variáveis independentes importantes para a classificação das amostras, como também foi mostrado na avaliação feita pelo *Kernel RBF, o Núcleo Polynomial considerou a NormNucl (*Nucléolos Normais*) como a variável menos significativa, correspondendo a 1,6%.*

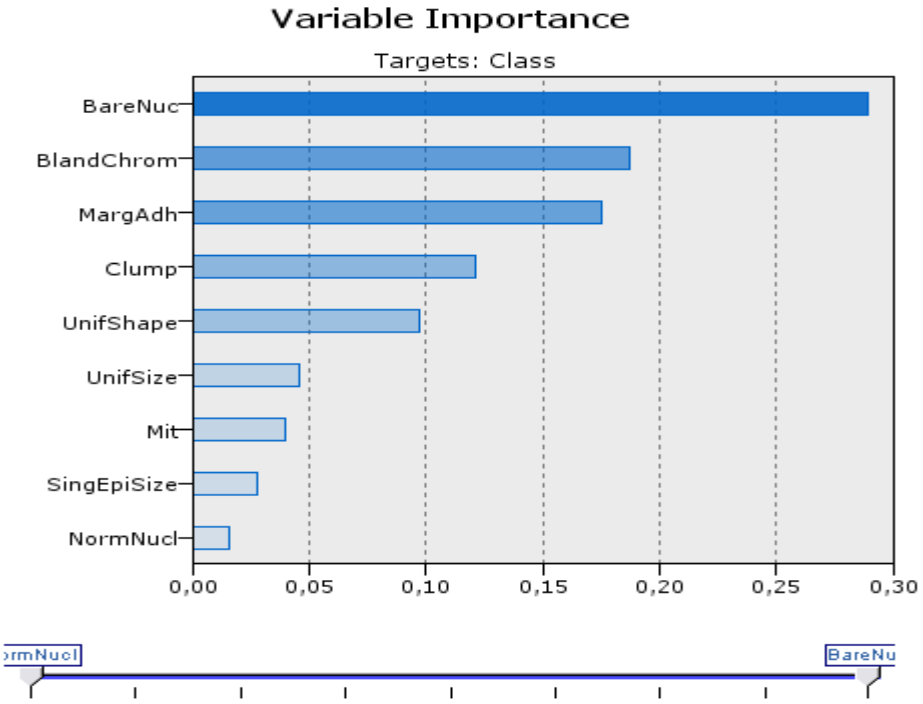


Figura 3.15: Resultado da análise feita pelo *kernel Polynomial*

Como foi feito nos *Kernels Sigmoid*, Linear e *RBF*, também fizemos a adição de um nó de analise para podermos avaliar o resultado encontrado pelo *Kernel Polynomial*, tendo como classificação correta em 100% das amostras de pacientes para determinar se suas células são benignas ou malignas, o que correspondeu a uma avaliação perfeita de todas as amostras analisadas pelo *Núcleo Polynomial,* ou seja, todos os 699 registros foram classificados corretamente, isso mostra que tal

Kernel obteve uma melhor classificação de suas amostras, como podemos observar na Figura 3.16

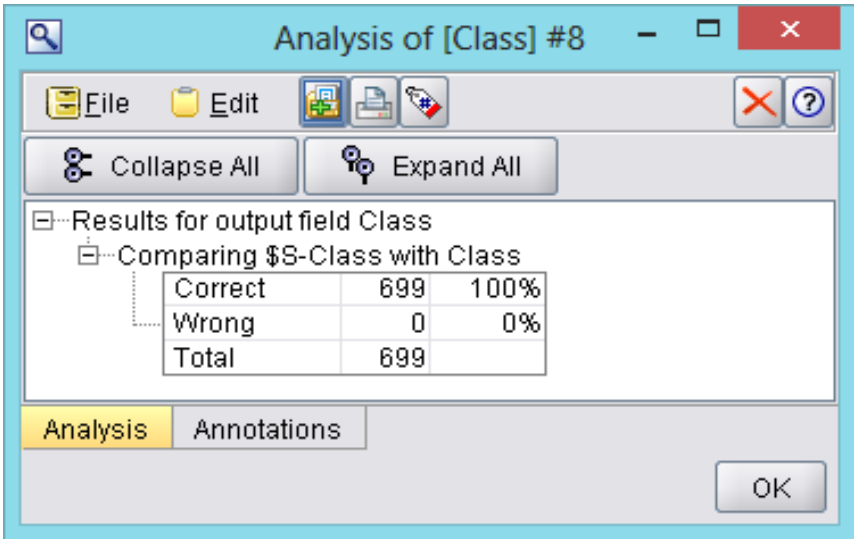
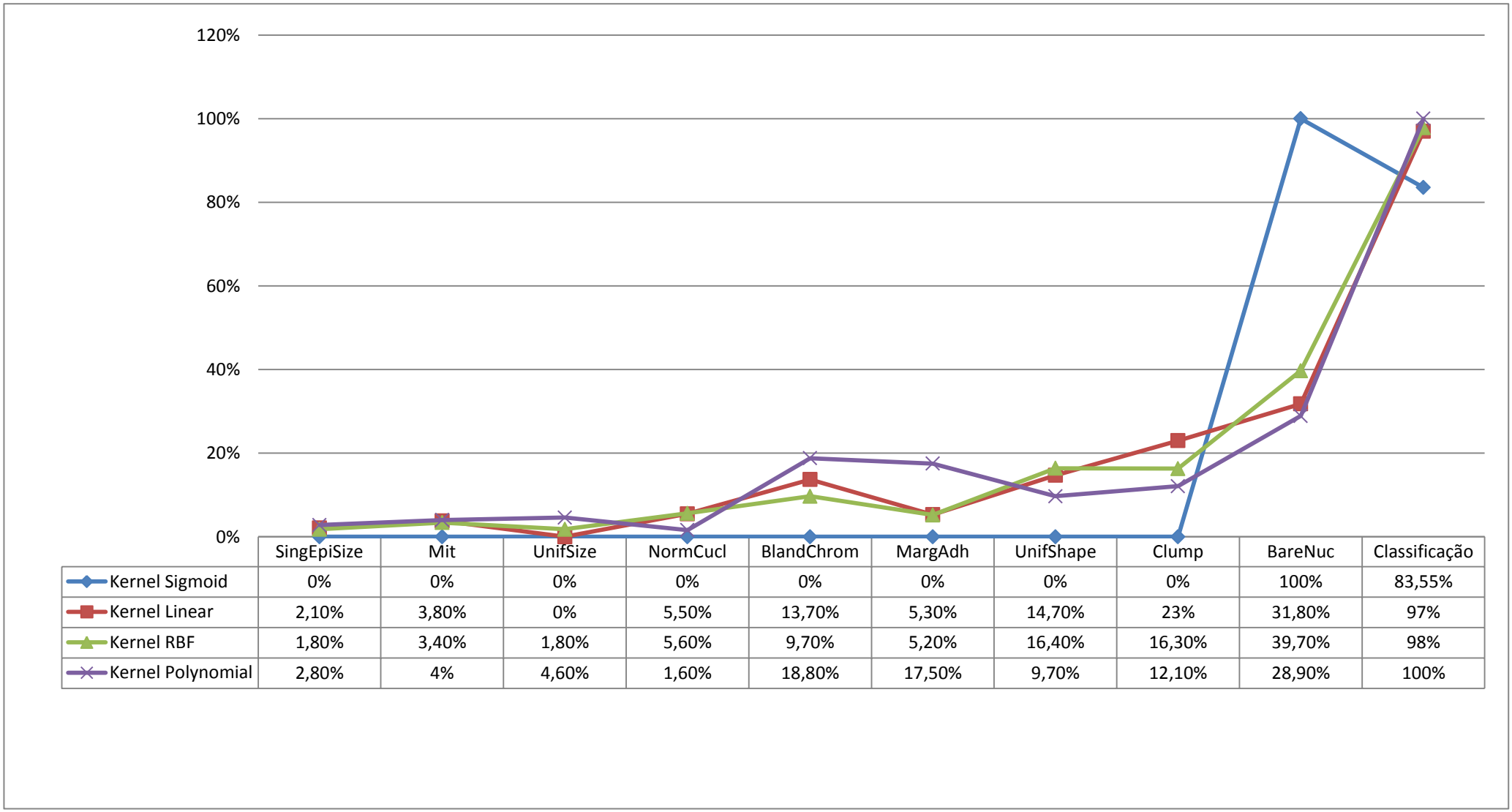


Figura 3.16: Classificação dos resultados em corretos e errados do *Kernel Polynomial*

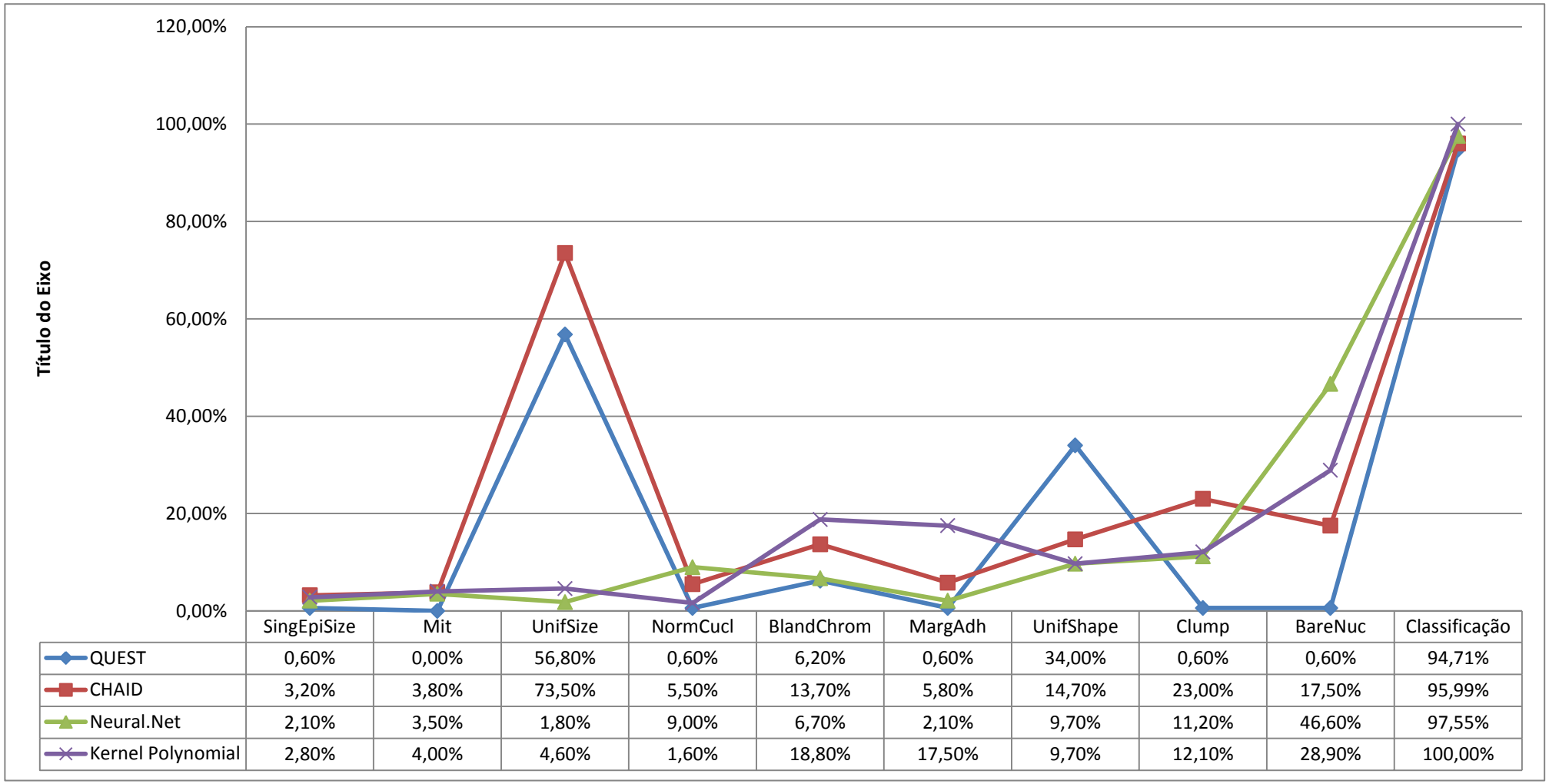
Na Figura 3.17 será mostrado o desempenho dos *Kernels* analisados, mostrando a variação das variáveis independentes em relação a variável dependente, a Figura mostra também que o *Kernel Polynomial* alcançou a melhor classificação das amostras analisadas.



	SingEpiSize	Mit	UnifSize	NormCucl	BlandChrom	MargAdh	UnifShape	Clump	BareNuc	Classificação
Kernel Sigmoid	0%	0%	0%	0%	0%	0%	0%	0%	100%	83,55%
Kernel Linear	2,10%	3,80%	0%	5,50%	13,70%	5,30%	14,70%	23%	31,80%	97%
Kernel RBF	1,80%	3,40%	1,80%	5,60%	9,70%	5,20%	16,40%	16,30%	39,70%	98%
Kernel Polynomial	2,80%	4%	4,60%	1,60%	18,80%	17,50%	9,70%	12,10%	28,90%	100%

Figura 3.17: Representação Gráfica da Classificação das Variáveis Independentes.

Como foi mostrado na Figura 3.17 o *Kernel Polynomial* teve melhor desempenho na classificação das amostras de pacientes que tinha alguma probabilidade de adquirir algum tipo de câncer. Na Figura 3.18 será mostrado o desempenho desse *Kernel* com outros algoritmos, tais como: Neural Network (Baseado em Redes Neurais), CHAID (Baseado em Árvore de Decisão) e o QUESTE (Baseado em Árvore de Decisão). Como será observado na Figura 3.18, o *Kernel Polynomial* obteve um resultado melhor que os algoritmos citados acima.



	SingEpiSize	Mit	UnifSize	NormCucl	BlandChrom	MargAdh	UnifShape	Clump	BareNuc	Classificação
QUEST	0,60%	0,00%	56,80%	0,60%	6,20%	0,60%	34,00%	0,60%	0,60%	94,71%
CHAID	3,20%	3,80%	73,50%	5,50%	13,70%	5,80%	14,70%	23,00%	17,50%	95,99%
Neural.Net	2,10%	3,50%	1,80%	9,00%	6,70%	2,10%	9,70%	11,20%	46,60%	97,55%
Kernel Polynomial	2,80%	4,00%	4,60%	1,60%	18,80%	17,50%	9,70%	12,10%	28,90%	100,00%

Figura 3.18: Representação Gráfica da Classificação das Variáveis Independentes com o uso dos algoritmo acima citados.

3.5 Discussão

A análise feita Com o método SVM foi possível levantar as seguintes conclusões com os dados da *UCI Machine Learning Repository*

O algoritmo SVM após verificar todos os registros, verificou que a variável mais importante ligada a classificação das amostras de pacientes que tinha alguma probabilidade de adquirir certo tipo de câncer, foi a *BareNuc*.

Segundo os resultados encontrados pelo SVM, a variável independente que mais influencia na definição da classificação das amostras feita pelos *Kernels* foi a *BareNuc*, sendo esta influenciado decisivamente em todos os *Kernels* utilizados.

Entre os resultados obtidos, verificou-se que Kernel *Polynomial* teve melhor desempenho na classificação dos registros, seguido pelo *kernel RBF*, Linear e por ultimo o *Kernel Sigmoidal*, esta análise mostrou também que o *Kernel Sigmoidal* utilizou somente a variável *BareNuc* como importante para a classificação das amostras.

Nesse capitulo vimos o método SVM aplicado a um estudo de caso de forma detalhada. Nele mostramos como usamos o algoritmo SVM para a base de dados do *UCI Machine Learning Repository*. Explicamos como foi feito a análise das variáveis de interesse, a análise e interpretação dos resultados, além de levantarmos discussão sobre o conhecimento minerado. Em seguida, vamos apresentar algumas conclusões, levantar as dificuldades encontradas, avaliar o que foi desenvolvido e explicar quais extensões do mesmo.

4. CONCLUSÃO

Nesta monografia apresentaram-se as etapas e técnicas para se chegar ao conhecimento através da Mineração de Dados em grandes repositórios de dados, onde foi mostrado a importância de sua utilização para construção do conhecimento. Pode-se concluir que a extração de conhecimento é feita através de um conjunto complexo de etapas, mas que no final ajuda a descobrir informações que estavam implícitas na base de dados, pois com a utilização de técnicas e ferramentas de análise de dados é possível obter uma visão mais detalhada sobre o que o repositório de dados possui de mais relevante, auxiliando na tomada de decisão e ajudando no aumento da produtividade das empresas públicas e privadas.

Apresentamos todas as etapas de KDD necessárias para a extração de conhecimento, além de algumas de suas técnicas. Mostramos como utilizar e interpretar o *Kernel Sigmoidal, RBF, Linear* e o *Polynomial* . Entre essas técnicas, mostramos de forma detalhada o uso do algoritmo do SVM e o aplicamos em um estudo de caso.

Nesse sentido, aplicamos o algoritmo *SVM (Support Vercto Machine*) na base de dados do *UCI Machine Learning Repository* que tinha como fundamentação a análise de amostras de pacientes que tinha uma probabilidade de adquirir câncer. Dessa forma podemos avaliar as variáveis mais significativas e menos significativas no processo de classificação dos registros verificados em questão.

Para trabalhos futuros, propomos utilizar uma nova base de dados para, ser

usados por outros algoritmos de classificação, tais como, Redes Neurais a fim de compararmos com o método *SVM* (*Support Vector Machine*), mostrando o desempenho e o conhecimento minerado pelos algoritmos.

Propomos também, usar o algoritmo *SVM* em outras bases de dados de pesquisas com informações semelhantes as da *UCI Machine Learning Repository* realizadas, por exemplo, em países emergentes que tenha um alto índice de doenças infecciosas, como é o caso de doenças com *HIV*, Malária, entre outras.

REFERÊNCIAS

SILVA L.M.M. Seleção de Tributos em comitês de Classifiadores Utilizando Algoritmo Genético. Universidade federal do Rio Grande do Norte – CEET, Disponível em: http://bdtd.bczm.ufrn.br/tedesimplificado//tde_arquivos/14/TDE-2011-07-8T010103Z-3550/Publico/LigiaMMS_DISSERT.pdf. Acesso em: fevereiro 2013

KLEINSCHIMIDT M. Mineração de Dados Para Avaliação do Perfil de Usuários do Sistema de Informação da Academia da UNIFALI, 2007. Disponível em: http://pt.scribd.com/doc/67478373/Monografia-Sistema-de-Informacao-para-Academia -de-Musculacao-noPW. Acesso em: fevereiro 2013

LORENA A.C.; CARVALHO A.C.P.F.L. Uma Introdução Às Support Vector Mahine Universidade de São Paulo, 2007. Dispnível em: onsumidorpositivo.com.br. Acesso em: janeiro 2013.

FARIAS. L. Support Vetor Machine. Inteligênia Artificial 5º ano Comp Disponivel em: http://www.slideshare.net/leandrofarias31/support-vector-machines-14492166. Acesso em: janeiro 2013.

GIRADELLO A.D. Um Estudo Sobre Máquina de Vetores de Suporte em Problema de Classifiação Universidade Estadual do oeste do Paraná, 2010. Disponível em: http://www.inf.unioeste.br/~tcc/2010/TCC-Adriano%20Douglas%20Girardello.pdf Aesso em: Fevereiro 2013.

JUNIOR G.B. Classificação de Regiões de Momografias em Massa e Não Massa Usando Estatísticas Espaciais e Máquina de Vetores e Suporte. Universidade Federal do Maranhão, 2008

MARTINS L.L. Detecção de Massas em Imagens Monográficas Através do Algoritmo Growing Neural Gas e da Função K de Ripley. Universidade Federal do Maranhão, 2007. Disponível em: http://www.tedebc.ufma.br/tde_arquivos/10/TDE-2008-02-11T191505Z-90/Publico/Leonardo%20Martins.pdf.

GONSALVES M.L.; NETTO M.L.A.; J J.Z.; COSTA J.A.f. Classifecação não Supervisionada de Imagens de Sensores Remotos Utilizadno redes Neurais Auto-

Organizáveis e Métodos de Agrupamento Hieráquicos. PUC – Minas, Unicamp – FEEC, UFRN, 2011. Disponível em: http://www.sigaa.ufrn.br/sigaa/verProducao?idProducao=277019&key=e6fed9886789840293e06aad0eb3bcd2.download-Downloads/M%C3%A1rcio%20-%20Unicamp%20-%202008%20-%20artigoRBC%20(1).pdf

AMO, S. Curso de Data Mining do Programa de Mestrado em Ciência da Computação da Universidade Federal de Uberlândia, 2003. Disponível em: <http://www.deamo.prof.ufu.br/CursoDM.html>. Acesso em: janeiro de 2013.

AURÉLIO, M.; VELLASCO, M.; LOPES C. H. Descoberta de conhecimento e mineração de dados apostila, 1999. Disponível em: < http://www.ica.ele.puc-rio.br/cursos/download/DM-apostila1.pdf>. Acesso em: janeiro de 2013.

BABTISTA J.; CARVALHO D. R. Data mining como apoio a decisão em projetos públicos. In: CONGRESSO BRASILEIRO DE COMPUTAÇÃO, 3., 2003, Itajaí. Anais...Itajaí:Univali-CTTMar, 2003. Disponível em: < http://www.ppgia.pucpr.br/~silla/publications/index.html>. Acesso em: janeiro de 2013.

BERRY, M. J. A.; LINOFF, G. S. Data mining techniques: for marketing, sales, and customer support. In: Michael J. A. Berry, Gordon Linoff, New York: JohnWiley, 2004. Disponível em: < http://www.wiley.com/WileyCDA/WileyTitle/productCd-0471470643.html>. Acesso em: desembro de 2012.

CARVALHO, L. A. V. Datamining: a Mineração de Dados no Marketing, Medicina, Economia, Engenharia e Administração. São Paulo: Érica, 2001.

COLLAZOS, K.; BARRETO, J. KDD ferramenta para análise de dados epidemiológicos. In: Anais do III Congresso Brasileiro de Computação – Workshop de Informática aplicada à Saúde - CBCOMP'2003, Itajaí, 2003. Acesso em: desembro de 2012.

DIAS, M. M. Parâmetros na escolha de técnicas e ferramentas de mineração de dados. Departamento de Informática, Maringá, Paraná, 2002. Disponível em:

<http://eduem.uem.br/ojs/index.php/ActaSciTechnol/article/ download/2549/1569>. Acesso em: fevereiro de 2013.

DOMINGUES, M. A. Generalização de regras de associação. Dissertação de Mestrado, ICMC-USP, São Carlos, São Paulo, 2004. Disponível em: <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-10082004-154242/pt-br.php > . Acesso em: dezembro de 2012.

HAIR, J. F.; ANDERSON, R. E.; TATHAM R. L.; BLACK, W. C. Multivariates Data Analysis. 4. ed. New Jersey: Prentice Hall, 1995. Disponível em: <http://books.google.com.br/books?id=S1gZAQAAIAAJ&hl=pt-BR&source=gbs_book_other_versions>. Acesso em: janeiro de 2013.

HAN, J.; KAMBER, M. Data Mining: Concepts and Techniques, In: Academic Press, USA, 2006. Disponível em: <http://www.cs.uiuc.edu/homes/hanj/bk2/toc.pdf>. Acesso em: fevereiro de 2013.

KLEINSCHMIDT, M. Mineração de dados para avaliação do perfil de usuários do Sistema de Informação da Academia da UNIVALI. Trabalho de Conclusão de Curso, Itajaí: UNIVALI, 2007. Disponível em: <http://siaibib01.univali.br/pdf/Marlon%20 Kleinschmidt.pdf>. Acesso em: janeiro 2013.

MUELLER, Alessandro. Uma Aplicação de Redes Neurais Artificiais na Previsão do Mercado Acionário. Dissertação de Mestrado, UFSC, Florianópolis. 1996. Disponível em: <http://www.eps.ufsc.br/disserta96/mueller/index/index.htm#sumario>. Acesso em: fevereiro de 2013.

MORGAN, J.N. e SONQUIST, J.A. Problems in the analysis of survey data and a proposal, Journal of the American Statistical Association, v. 58, 1963.

PACHECO, M. A.; VELLASCO, M.; LOPES, C. H. Descoberta de conhecimento e mineração de dados, Notas de Aula em Inteligência Artificial. Rio de Janeiro, ICA-Laboratório de Inteligência Computacional Aplicada, departamento de Engenharia

elétrica – PUC-RIO, 1999. Disponível em: <www.ica.ele.puc-rio.br/cursos/download/DMapostila1.pdf>. Acesso em: janeiro de 2013.

ROMÃO, W. Descoberta de Conhecimento Relevante em Banco de Dados sobre Ciência e Tecnologia. Trabalho de Pós-Graduação. Florianópolis-SC : UFSC, 2002. Disponível em: < http://www.din.uem.br/~intersul/intersul_arquivos/documentos/Tese%20Wesley.pdf>. Acesso em: dezembro de 2012.

SCHENATZ, B. N. Utilização de Data Mining em um sistema de informação gerencial para o diagnóstico da formação de professores da graduação. 2005. Dissertação de Mestrado, UFSC, Florianópolis. 2005. Disponível em: < http://aspro02.npd.ufsc.br/arquivos/220000/224900/ 18_224929.htm>. Acesso em: janeiro de 2013.

SILVA, S. G. Estudo de técnicas e utilização de mineração de dados em uma base de dados da saúde pública. Trabalho de Conclusão de Curso, Universidade Luterana do Brasil, Canoas, 2003. Disponível em: < http://pt.scribd.com/doc/56537190/Estudo-de-Tecnicas-e-Utilizacao-de-Mineracao-de-Dados-em-uma-base-de-dados-da-saude-publica-gercely-da-silva>. Acesso em: janeiro de 2013.

SILVEIRA, R. F. Mineração de dados aplicada a definição de índices em sistemas de raciocínio baseado em casos. Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul, Porto Alegre. 2003. Disponível em: < http://www.inf.ufrgs.br/bdi/administrator/components/com_jresearch/files/publications/Websis-RosemariSilveira.pdf >. Acesso em: janeiro de 2013.

TURE M.; TOKATLI F.; KURT I. Using Kaplan-Meier Analysis together with Decision Tree Methods (C&RT, CHAID, QUEST, C4.5 and ID3) in Determining Recurrence-free Survival of Breast Cancer Patients. Science Direct. Expert System with Applications 36. 2009. Disponível em: < http://dl.acm.org/citation.cfm?id=1465032>. Acesso em: fevereiro de 2013.