

UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"

FACULDADE DE CIÊNCIAS - CAMPUS BAURU

DEPARTAMENTO DE COMPUTAÇÃO

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

JOÃO PAULO DE VASCONCELOS

**APRENDIZADO DE MÁQUINA:
ESTUDO DE CASO EM BOLSA DE VALORES**

BAURU

2017

JOÃO PAULO DE VASCONCELOS

**APRENDIZADO DE MÁQUINA:
ESTUDO DE CASO EM BOLSA DE VALORES**

Trabalho de Conclusão de Curso do Curso
de Ciência da Computação da Universidade
Estadual Paulista “Júlio de Mesquita Filho”,
Faculdade de Ciências, Campus Bauru.
Orientador: Prof. Dr. João Paulo Papa

BAURU
2017

João Paulo de Vasconcelos

Aprendizado de Máquina:

Estudo de Caso em Bolsa de Valores/ João Paulo de Vasconcelos. – Bauru, 2017-
43 p. : il. (Colaboradores) ; 30 cm.

Orientador: Prof. Dr. João Paulo Papa

Trabalho de Conclusão de Curso – Universidade Estadual Paulista “Júlio de Mesquita Filho”
Faculdade de Ciências
Ciência da Computação, 2017.

1. Aprendizagem de Máquina 2. Ciência da Computação 3. Programação 4. Predição de Dados
5. Bolsa de Valores

João Paulo de Vasconcelos

Aprendizado de Máquina: Estudo de Caso em Bolsa de Valores

Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Faculdade de Ciências, Campus Bauru.

Banca Examinadora

Prof. Dr. João Paulo Papa

Filiação: Dco - FC - Unesp/Bauru

Profª Drª. Simone das Graças Domingues Prado

Filiação: Dco - FC - Unesp/Bauru

Prof. Dr. Kelton Augusto Pontara da Costa

Filiação: Dco - FC - Unesp/Bauru

Bauru, 29 de Novembro de 2017.

"Eu dedico este trabalho aos meus pais Jaime e Elisa, e meu irmão Luís. A sabedoria é a maior herança que alguém pode querer.

Agradecimentos

Agradeço à minha família, à cada um daqueles que participaram de perto dessa jornada. A minha mãe Elisa e meu pai Jaime, o meu profundo agradecimento pela imensa dedicação, carinho, educação, o amparo incondicional em cada passo da minha vida e por sempre transmitirem força, coragem e determinação. Eles sempre me proporcionaram a chance de ir além. Obrigado por me apoiarem sempre em minhas decisões que fizeram eu trilhar este caminho. Ao meu irmão Luís pelo companheirismo, calma, dedicação e paciência. Por sempre trazer alegria nos dias mais difíceis, você tem um grande coração e eu o admiro.

Agradeço ao meu orientador Prof. Dr. João Paulo Papa, por tudo o que ele cuidadosamente se empenhou para que eu aprendesse. Pela sua dedicação profissional e prazer pela pesquisa. Pela bondade e humildade do seu coração. Agradeço aos colegas da UNESP: Fasu, Manoella, Henrique, Thalita, Isabella, Empanado, Grecin, Ceja, Torta, Siri, Pala, Wellington, Mateus pelo bom convívio, pelas experiências divididas e pelo conhecimento compartilhado, pelas confraternizações e churrascos. Agradeço aos meus amigos pelo apoio e carinho durante esse tempo da Graduação, pelas vezes que me apoiaram e comemoraram comigo. Obrigado.

Agradeço ao pessoal da graduação da UNESP de Bauru, principalmente do Departamento da Computação, aos professores, secretários, coordenadores e todos aqueles que fazem parte dessa incrível instituição. Por fim, gostaria de agradecer à UNESP pela estrutura e apoio disponibilizados durante minha formação.

*"Só se pode alcançar um grande êxito quando
nos mantemos fiéis a nós mesmos"(F. Nietzsche)*

Resumo

Este projeto tem como objetivo apresentar os diversos métodos utilizados para Aprendizagem de Máquina para a predição de dados principalmente para a bolsa de valores e aplicar o método a um protótipo de sistema que tem por finalidade realizar a predição de ações para fins de estudo. Este trabalho expõe os métodos populares para a predição de dados com foco em ações da bolsa de valores que utilizam sistemas automatizados. Também destaca técnicas de tratamento de dados anterior a predição, o pré-processamento, e a aprendizagem de máquina supervisionada, como Máquina de Vetores de Suporte, *Support Vector Regression* (SVR), 1-vizinho Mais Próximo e K-vizinhos Mais Próximos. Este trabalho também tem como propósito transmitir os pontos de se criar aplicações multiplataforma, não apenas em foco de algoritmos, mas também em métodos para que facilite o entendimento dos estudos sobre Aprendizagem de Máquina e uma melhor experiencia para os usuários frisando a utilização do Python como linguagem plataforma. As ferramentas utilizadas durante o desenvolvimento do protótipo são listadas. Ao final, é discutido a importância do estudo da Aprendizagem de Máquina.

Palavras-chave: aprendizagem de máquina, predição de dados, supervisionada, máquina de vetores de suporte, multiplataforma.

Abstract

This project aims to present the various methods used for Machine Learning to predict data mainly for the stock market and to apply the method to a prototype system that is intended to perform predictions for study purposes. This paper exposes the popular methods for predicting data, focusing on stock exchanges using automated systems. It also highlights data processing techniques prior to prediction, pre-processing and supervised machine learning, such as Support Vector Regression (SVR), 1-nearest neighbours and k-nearest neighbours. This work also aims to justify the creation of multiplatform applications, not only with focus on algorithms, but also in methods to facilitate the understanding of studies on Machine Learning and a better user experience, emphasizing the use of python as platform language. The tools used during prototype development are listed. In the end, the importance of the study of Machine Learning is discussed towards the end of the paper.

Keywords: machine learning, data prediction, supervised, support vector machine, multiplatform.

Lista de ilustrações

Figura 1 – Tipos da Aprendizagem de Máquina.	15
Figura 2 – Ilustração de hiperplano canônicos e separador.	25
Figura 3 – Exemplo ilustrativo do algoritmo 1-NN.	27
Figura 4 – Impacto do valor de k no algoritmo K-NN.	28
Figura 5 – Ilustração do conjunto de dados.	32
Figura 6 – Ilustração do conjunto de dados com curva de separação.	32
Figura 7 – Ilustração do conjunto de dados com linha de separação do hiperplano.	33
Figura 8 – Tela Inicial do <i>Google Finance</i>	34
Figura 9 – Campo de Pesquisa do <i>Google Finance</i>	34
Figura 10 – <i>Download</i> dos arquivo <i>CSV</i> através do <i>Google Finance</i>	34
Figura 11 – Tela Inicial do Programa de Predição.	36
Figura 12 – PASSO 1: Seleção do arquivo para o Programa de Predição.	37
Figura 13 – PASSO 2: Seleção do método do modelo para o Programa de Predição.	37
Figura 14 – PASSO 3: Escolha do dia do mês e execução do Programa de Predição.	37
Figura 15 – Arquivo "Tesla.CSV"aberto pelo Bloco de Notas	38
Figura 16 – A tela do inicial do programa com os dados preenchidos	38
Figura 17 – Gráfico gerado após execução do Programa	39

Lista de tabelas

Tabela 1 – Tabela de Atributo/Valor	14
Tabela 2 – Conjunto Hospital	17
Tabela 3 – Tipo de atributos do Conjunto Hospital	18
Tabela 4 – Escala dos atributos do Conjunto Hospital	19

Sumário

1	INTRODUÇÃO	12
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	Conjunto de Dados	16
2.1.1	Análise de Dados	17
2.2	Pré-processamento de dados	19
2.2.1	Eliminação de Dados Manuais	19
2.2.2	Integração de Dados	20
2.2.3	Amostragem de Dados	20
2.2.4	Limpeza de Dados	20
2.3	Aprendizagem Supervisionado	21
2.3.1	Máquina de Vetores de Suporte (SVM)	22
2.3.2	Máquina de Vetores de Suporte: Regressão (SVR)	25
2.3.3	1-vizinhos Mais Próximos (1NN)	26
2.3.4	K-vizinhos Mais Próximos (KNN)	26
2.3.5	K-vizinhos Mais Próximos (KNN): Aspectos positivos	28
2.3.6	K-vizinhos Mais Próximos (KNN): Aspectos negativos	29
2.4	Bolsa de Valores	29
2.4.1	National Association of Securities Dealers Automated Quotations (NASDAQ)	29
2.5	Ferramentas Computacionais	30
2.5.1	<i>Python</i>	30
2.5.2	wxFormBuilder	30
3	SOBRE A APLICAÇÃO	31
3.1	Método escolhido	31
3.2	Dados de Treinamento	33
3.3	Desenvolvimento	35
3.3.1	Pacote <i>Numpy</i>	35
3.3.2	Pacote <i>WX</i>	35
3.4	Visualização: Interface gráfica do utilizador (GUI)	36
3.5	Exemplo: Tesla, Inc.	38
4	CONCLUSÃO	41
4.1	Trabalho Futuro	42
	REFERÊNCIAS	43

1 Introdução

A ciência da Inteligência Artificial (IA), teve seu sonho inicial com a vontade do homem de compreender seu raciocínio lógico, isto é, como ele pode condensar um punhado de dados e compreendê-los, percebê-los, manipulá-los e realizar conexões muito além dele próprio. Foi na Antiguidade Clássica que este conceito teve início, porém foi no Renascimento que começaram a surgir os mecanismos, como por exemplo o relógio e sendo assim, o homem começou a perceber o poder de facilitação que uma máquina pode promover no dia-a-dia. Entretanto, os maiores avanços deste conceito aconteceram nos trabalhos matemáticos do século XVII em diante. No século XIX surge um matemático chamado Alan Turing, o pai dos computadores e da Inteligência Artificial. A medida que o ser humano vem buscando formas de facilitar suas atividades e simplificar cada vez mais o seu cotidiano a IA pode ter um papel importante para isto. IA foi definida por Haugeland como “o novo e interessante esforço para fazer os computadores pensarem... máquinas com mentes, no sentido total e literal” (HAUGELAND, 1985). Com a invenção do computador, o cotidiano tornou-se mais ameno mesmo com as dificuldades que antes eram consideradas problemáticas. Contudo, durante anos o ser humano não parou de procurar meios que facilitam sua vida, e acabaram desenvolvendo a ideia de uma máquina que tenha características de seres vivos. Assim, ocorreu o desenvolvimento da Inteligência Artificial, na qual, está cada vez mais parecida com o ‘pensar’ humano.

De acordo com o dicionário *Oxford*, *artificial intelligence* (inteligência artificial) corresponde a uma área de pesquisa sobre computadores simulando o comportamento humano inteligente (WEHMEIER, 2000). A IA mantém-se presente, constantemente, no nosso cotidiano em todos os setores, na automatização de indústrias, nos celulares, nos computadores, na medicina, entre outros.

Nas últimas décadas com a crescente complexidade dos problemas a serem tratados computacionalmente e do volume de dados gerados por diversos setores, tornou-se clara a necessidade de ferramentas computacionais mais sofisticadas, que fossem mais autônomas, reduzindo a necessidade de intervenção humano e dependência de especialistas (CARVALHO, 2015, p. 129)

A Aprendizagem de Máquina (AM) surgiu para suprir esta necessidade pois ela possibilita por meio de experiências passadas criar hipóteses e funções que irão resolver problemas futuros. Esta é uma subárea da IA que é capaz de criar modelos e aprender sozinho com base nas informações disponíveis. "Aprendizado de máquina é o campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados"(SAMUEL, 1959). Aprendizagem de Máquina (AM) nada mais é do que a habilidade do computador obter conhecimento com acontecimentos passados para criar tendências e assim resolver problemas

e analisar dados futuros, desta forma, o seu desempenho em determinada tarefa é medido pelo maior nível de experiência que adquire para executá-la. "Um sistema capaz de aprender é aquele que se modifica automaticamente, no sentido de que ele possa fazer as mesmas tarefas sobre um mesmo domínio de conhecimento, de uma maneira cada vez mais eficiente"(SIMON, 1983). Para ter um ótimo desempenho, a experiência é fornecida pela quantidade de dados processados pelo algoritmo de aprendizagem. Em tempos passados, onde as mídias físicas predominavam, o AM era menos preciso em razão da dificuldade de obtenção dos dados para o treinamento destes.

Em 1997, uma reportagem intitulada "*Information overload causes stress*" publicada pela *Reuters Magazine* baseada em estatísticas apontadas pela universidade Humboldt, nos Estados Unidos, que demonstrava que a quantidade de informação escrita disponível no mundo dobrava a cada 5 anos (WILSON, 2001). O avanço do meio de telecomunicação, trouxe uma Era em que o volume de dados gerado é colossal e os dados são armazenados de forma digital, o que representa um bom desempenho para as máquinas que tiverem acesso aos dados. A AM é uma ferramenta poderosa para obtenção de conhecimento e análise de dados, porém, deve-se ressaltar que não existe um algoritmo que seja melhor do que outro para resolver um problema específico, cada um tem suas próprias características. "As técnicas de aprendizado de máquinas empregam um princípio de inferência denominado indução, no qual é possível obter conclusões genéricas a partir de um conjunto particular de exemplos." (LORENA A. C.; CARVALHO, 2007).

Para que AM tenha uma alta performance e qualidade em suas análises, é necessário que o treinamento seja realizado com a maior quantidade de dados possíveis, quanto mais, melhor e isto tornará a predição mais precisa. Uma das maiores empresas de tecnologia do mundo, a Google, tem em seu poder um volume consideravelmente grande de dados e com isso suas ferramentas tornam-se mais precisas, parecendo 'mágica', porém, esses dados servem para o treinamento e uma melhora gradual da qualidade do algoritmo de aprendizagem. As técnicas de AM são divididas em duas principais: supervisionado e não supervisionado. A técnica de aprendizagem supervisionada com o uso da Regressão e da Classificação é uma ótima opção para predição de dados, e normalmente é usada com o objetivo de encontrar o resultado a ser previsto e foram utilizadas neste trabalho. Com base neste cenário o estudo a seguir visa o desenvolvimento de um software de AM que tenha a habilidade de analisar e suprir a necessidade latente das companhias que analisam grande volumes de dados, seja eles multidisciplinares.

2 Fundamentação Teórica

Será declarados alguns termos e conceitos de aprendizado de máquina aplicados, utilizando a notação do próprio autor durante as aulas ministradas pelo Professor Doutor João Paulo Papa e o Professor Andrew Ng (Coursera – “*Machine Learning*”). Como introdução ao tema é apresentado uma tabela no padrão de Atributo/Valor, cujo a acepção é de um conjunto de n_{ex} exemplos x_i com $i = 1; \dots; n_{ex}$. Cada exemplo é composto por atributos A_j , com $j = 1; \dots; n_{at}$. Sendo o valor de cada atributo, o elemento x_{ij} . Outro atributo, além do A_i , é especial e não é obrigatório de cada exemplo, ele é denominado classe ou rótulo e representado pela letra y e pertencente do conjunto Y de classes discretas y_h , com $h = 1; \dots; n_{cl}$. Na Tabela 1 é descrito a representação de dados no formato de tabela atributo/valor, com conceito de exemplo x_i , atributos A_j , valores dos atributos x_{ij} , atributos A_{cl} e os valores dos atributos classe y_i .

Tabela 1 – Tabela de Atributo/Valor

	A_1	A_2	...	$A_{n_{at}}$	A_{cl}
X_1	x_{11}	x_{12}	...	$x_{1n_{at}}$	y_1
X_2	x_{21}	x_{22}	...	$x_{2n_{at}}$	y_2
...
$X_{n_{ex}}$	$x_{n_{ex}1}$	$x_{n_{ex}2}$...	$x_{n_{ex}n_{at}}$	$y_{n_{ex}}$

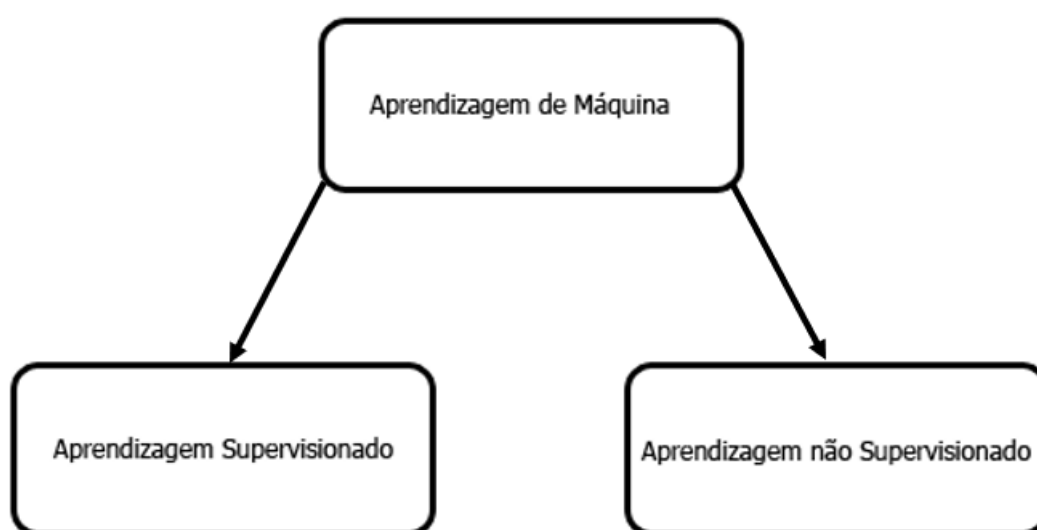
Fonte: Elaborada pelo autor.

A representação de dados é composta por conjuntos de dados que são fornecidos como entrada para os algoritmos de aprendizado de máquina. Os conjuntos de aprendizado são divididos em dois:

- **Conjunto de treinamento:** Formado por exemplos rotulados que servem como *Input* do algoritmo de aprendizagem para que seja realizada a indução (treinamento através de aprendizagem) do Classificador. Os dados que compõem o conjunto de treinamento que serão utilizados para que possa construir hipóteses do mundo real do qual deseja-se abstrair o conhecimento e realizar a predição.
- **Conjunto de teste:** Composto por exemplos já rotulados que serão utilizados para testar o classificador já treinado. Nenhum dos objetos de exemplos que compõe o conjunto de teste deve ser um mesmo já apresentado ao algoritmo durante o processo de treinamento, ou seja, os conjuntos de treinamento e teste devem estar separados. Desta maneira torna-se possível realizar os testes e medir a qualidade do classificador, uma vez que ele não conhece os objetos de exemplos apresentados.

O conjunto de treinamento é utilizado para adquirir um bom conhecimento sobre o problema, criando-se um modelo, para que posteriormente seja realizada a predição dos novos objetos, aqueles que fazem parte do conjunto de teste. Os objetos de treinamento podem ser considerados como rotulados e os não rotulados. Os conjuntos rotulados são quando os atributos estão submetidos a classe, já os não rotulados são aqueles em que seus atributos não estão submetidos a nenhuma. Então que o conceito denominado de grau de supervisão, que classifica o aprendizado de máquina devido a quantidade de objetos de exemplos de treinamento rotulados no conjunto de treinamento. O aprendizado de máquina, contém dois ramos a serem seguidos, o aprendizado supervisionado e não supervisionado (WEISS S.M.; KULIKOWSKI, 1991), como mostrado na figura 1.

Figura 1 – Tipos da Aprendizagem de Máquina.



Fonte: Elaborada pelo autor.

Independentemente do tipo de aprendizado no qual o problema de predição estiver inserido, a etapa de treinamento de classificadores possui alguns conceitos importantes a serem definidos, para o melhor entendimento do conteúdo:

- **Indutor:** Algoritmo de aprendizado que utiliza um processo de indução para gerar uma hipótese ou modelo, que é denominado como classificador.
- **Classificador:** Modelo que representa as classes de um problema já proposto e pode ser utilizado para rotular os objetos de teste (exemplos).
- **Overfitting (Super Ajuste):** Tipo de situação em que o classificador passar por um processo em que precisa se ajustar aos exemplos de treinamento. Isso ocorre geralmente quando o modelo não é capaz de generalizar e abstrair o conceito dos exemplos do conjunto, normalmente ocorrem devido à complexidade do modelo proposto. Consequentemente, o desempenho torna-se muito ruim para o teste, e deixa de ser aceitável para o

treinamento. Um exemplo de super ajuste é a rede neural com uma quantidade grande de neurônios.

- ***Underfitting* (Sobre ajuste):** Geralmente ocorre quando o classificador é muito simples e não tem base para sustentação em representar se conceito pode ter ocorrido. Para caso de *Underfitting* a classificação não agrega nenhum conforto para o conjunto de treinamento e nem para o de teste. Um exemplo é quando uma árvore de decisão gerada tiver apenas um nível.

Alguns métodos preditivos serão apresentados neste trabalho, estes métodos são relacionados ao aprendizado Supervisionado.

2.1 Conjunto de Dados

Todos os dias, uma enorme quantidade de dados é gerada. Existe uma estimativa de que a cada 20 meses dobra a quantidade de dados armazenada nos bancos de dados do mundo (WITTEN, 2011).

Diariamente são gerados grande quantidade de dados durante as transações financeiras, captura de imagens, navegação na internet, entre outros. Estes dados são armazenados em formatos diferentes, como séries temporais, *itemsets*, transações, grafos, textos, imagens, áudios e vídeos. Essa quantidade de crescimento exponencial de informações gera uma distância gigante entre os dados gerados e analisados dos dados já compreendidos. Os conjuntos de dados são formados por objetos que podem representar objetos físicos, como um banco, ou ainda mais abstrato como anotações de dados financeiros de uma empresa, ações da Bolsa de Valores. De forma geral, os objetos possuem características próprias, chamadas de atributos que servem para o algoritmo como conjunto de entrada.

Para utilizar um algoritmo de AM com técnicas de indução, não basta apenas ter um grande volume de dados que, as vezes, tem a necessidade de passar por uma fase conhecida por pré-processamento de dados. As técnicas de pré-processamento são frequentemente utilizadas para transformar os dados em uma linguagem de entendimento a serem aplicados nos algoritmos de AM. As tarefas mais conhecidas durante essa etapa são:

- Eliminação manual de atributos;
- Integração de dados;
- Amostragem de dados;
- Balanceamento de dados;
- Limpeza de dados.

As organizações em geral possuem, muitas vezes, mais de uma base de dados, e essas informações quando vindas de lugares diferentes para serem a entrada do algoritmo de AM, podem trazer problemas, e tem a necessidade de serem integradas, e este volume de informações devem possuir o menor problema de redundância e inconsistência, pois isto pode gerar dificuldade quando existe um grande número de dados para a interpretação. Vários algoritmos de AM têm muita dificuldade de utilizar os dados no formato original que se é armazenado, e para tratar o problema é preciso passar pela transformação de valores simbólicos para valores numéricos para o fácil entendimento do algoritmo.

2.1.1 Análise de Dados

A análise das características de um conjunto de dados proporciona a descoberta de padrões e de tendências que fornecem informações valiosas para o entendimento do problema e do processo que gerou os dados. Essas características, muitas vezes são obtidas por meio de aplicação de fórmulas estatísticas simples, outro método de obter é com o uso de técnicas de visualização. Os dados podem ser apresentados na forma de uma matriz M_{ij} , onde i são a quantidade de objetos e j o número de características. Considerando o conjunto *Hospital*, representado pela Tabela 2:

Tabela 2 – Conjunto Hospital

Id.	Nome	Idade	Sexo	Peso	Manchas	Temp.	# Int.	Est.	Diagnóstico
129534	Ricardo	23	M	79	Concentrada	38,0	2	SP	Doente
792018	Paula	26	F	67	Inexistente	39,5	4	MG	Doente
235122	Manoella	21	F	52	Espalhado	38,0	2	RS	Saudável
533441	Elisa	48	F	72	Inexistente	38,5	8	SP	Doente
978490	Luis	32	M	43	Uniforme	37,6	1	PE	Saudável
003471	Jaime	59	M	92	Inexistente	38,0	3	GO	Doente
252172	João	25	M	67	Espalhadas	39,0	6	AM	Doente
683339	Paloma	25	F	87	Uniformes	38,4	2	RJ	Saudável

Fonte: Elaborada pelo autor.

Cada objeto da tabela é a representação de pacientes, cujo as características descritas nas linhas da tabela são colocadas como os atributos que serviram de entrada para o algoritmo e também vai representar os pacientes do Hospital. Os atributos são apresentados como: identificação, nome, idade, sexo, peso; e dados do resultados clínicos como: manchas encontradas, temperatura, quantidade de internações, estado de origem e o diagnóstico da doença. Dentre as características existe um atributo alvo, denominado atributo meta ou de saída, que representa o interesse sobre qual deseja-se realizar a predição. As tarefas preditivas focam e se baseiam na presença deste atributo especificamente.

Os valores que um atributo pode assumir podem ser definidos de diversas formas, duas delas são: tipo e escala. O tipo de um atributo representa o grau de quantização dos dados, e

a escala indica o valor relativo. O conhecimento destes dois, facilita e auxilia para identificar a melhor forma de preparação dos dados para poderem ser modelados posteriormente.

O tipo de atributo define se aquele em específico é quantitativo, ou qualitativo. Um exemplo para atributos cujo o valor é qualitativo são apresentados pelo conjunto {pequeno, médio, grande}, esses atributos não podem ser utilizados em operações aritméticas. O conjunto {20,56,83} representam valores quantitativos e podem ser utilizados em operações aritméticas. Se considerar os tipos das colunas da tabela do conjunto do Hospital (Tabela 2), pode ser indicados na Tabela 3:

Tabela 3 – Tipo de atributos do Conjunto Hospital

Atributo	Classificação
Id.	Qualitativo
Nome	Qualitativo
Idade	Quantitativo
Sexo	Qualitativo
Peso	Quantitativo
Manchas	Qualitativo
Temp.	Quantitativo
# Int.	Quantitativo
Estado	Qualitativo
Diagnostico	Qualitativo

Fonte: Elaborada pelo autor.

A escala quem define as operações que podem ser realizadas sobre os valores dos atributos. A escala dos atributos pode ser classificada como nominais, ordinais, intervalares ou racionais.

Na escala nominal, correspondente aos nomes, cujo carrega a menor quantidade possível de informação, não possuindo uma relação de ordenação e desta maneira acaba diminuindo as operações que podem ser realizadas nos atributos. As operações mais comuns realizadas são de igualdade ou desigualdade de valores. Exemplos de atributo são os nome, CPF, cores, sexo, CEP, entre outros.

A escala ordinal, é aquela representada por atributos que podem ser colocados em ordem. Os operadores de comparação matemáticos $>$, $<$, \leq , \geq podem ser utilizados para os atributos de escala ordinal. Por exemplo, a temperatura que pode ser definida como fria, natural e quente.

A escala intervalar os atributos são representados por números que variam dentro de um intervalo, sendo assim possível definir tanto a ordem, quanto à magnitude entre dois valores. Um exemplo para escala intervalar seria a temperatura, que possui diferentes tipos escala a em *Celsius* (C) e *Fahrenheit* (F), por exemplos, o valor de uma temperatura de 30 graus Celsius é diferente do valor de 30 graus *Fahrenheit*. Porém, os valores podem ser ajustados se

transformados para uma terceira escala de temperatura, a escala Kelvin (K), cujo o valor Zero de temperatura é o real.

A escala racional é aquela que carrega a maior quantidade de informações que são representados por números que possuem um significado absoluto. Por exemplo, utilizando a ideia de uma empresa em que um colaborador recebe o salário R\$2.000,00 e outro tem o seu em R\$6.000,00, desta maneira podemos afirmar que o segundo colaborador recebe 3 vezes mais que o primeiro.

Baseando-se na tabela 2 do conjunto Hospital podemos classificar os atributos por escala, como indicado na tabela 4:

Tabela 4 – Escala dos atributos do Conjunto Hospital

Atributo	Classificação
Id.	Nominal
Nome	Nominal
Idade	Racional
Sexo	Nominal
Peso	Racional
Manchas	Nominal
Temp.	Intervalar
# Int.	Racional
Estado	Nominal
Diagnostico	Nominal

Fonte: Elaborada pelo autor.

2.2 Pré-processamento de dados

A forma que os dados são apresentados para o algoritmo de AM influência diretamente em seu desempenho. O conjunto de dados, como dito anteriormente, podem ser apresentados com diferentes características, dimensões e/ou formatos. Existem diferentes técnicas de pré-processamentos para melhorar a qualidade dessas informações e minimizar ou eliminar problemas existentes nos conjuntos. As técnicas podem facilitar o uso da AM criando modelos com maior precisão e mais fiél ao que realmente representa. Estes procedimentos são necessários uma vez que os algoritmos de AM trabalham apenas com valores numéricos.

2.2.1 Eliminação de Dados Manuais

As eliminações de dados manuais devem ser feitas por especialistas da informação obtida, os atributos que serão removidos são aqueles que não tem nenhuma significância para o algoritmo que vai realizar a predição. Um exemplo, para um conjunto de dados, como o do Hospital, cujo os dados relevantes que irão fazer parte da predição da amostra de teste, é se o

paciente está ou não doente, os atributos como nome e identificação do paciente é totalmente dispensada.

2.2.2 Integração de Dados

Para o início da utilização da técnica de AM, a integração dos dados é uma fase muito importante, como já dito anteriormente, os dados podem vir de fontes de diferentes ambientes e isso significa que precisam e devem ser tratados. Os atributos passam por uma identificação para que sejam posteriormente combinados. Durante esta identificação são realizadas buscas para encontrar atributos com valores em comum, isto pode ter de ser dificultoso quando os dados podem vir com nomes, bases de dados e com tempo de atualizações diferentes.

2.2.3 Amostragem de Dados

Algoritmos de AM supervisionados, como K-Vizinhos Mais Próximo (KNN - *k-nearest neighbours* (FIX, 1951)) que é baseado em instâncias podem ter dificuldade de lidar com grande volume de dados e podem apresentar saturação de memória. Existe um balanço que deve ser considerado entre processamento computacional e acurácia (taxa de predição corretas). Quanto maior a quantidade de dados utilizados maior será a acurácia e menor vai ser a eficiência computacional durante o processo indutivo. Para que isto não seja um problema, utiliza-se a divisão do conjunto original de exemplos em amostras menores, porém os dados devem obedecer a mesma distribuição estatística que gerou o conjunto de dados originais, desta maneira, seria capaz de fornecer uma estimativa da informação contida nos dados inicial, isto é, permitindo tirar conclusões do todo a partir de uma parte.

2.2.4 Limpeza de Dados

Os conjuntos originais podem apresentar dificuldades para o algoritmo devido a sua qualidade. Os problemas com dados mais frequentes são:

- Ruídos: Possuem erros ou valores que são diferentes do esperado;
- Inconsistente: Contradiz valores de outros atributos do mesmo objeto;
- Redundantes: Quando dois ou mais objetos tem os mesmos valores para todos os atributos;
- Incompletos: Ausência de valores para alguns atributos em parte de dados do objeto.

A presença destas deficiências em um conjunto de dados pode acarretar em estatísticas e análises incorretas do algoritmo. Cada problema tem uma forma diferente de ser tratada, são elas por meio de algoritmos, ou até mesmo de forma manual.

2.3 Aprendizagem Supervisionado

Para melhor entendimento do aprendizado de máquina supervisionado, será realizada uma análise de uma simples tarefa que pode ser facilmente resolvida por este tipo de aprendizado. Para realizar a predição de dados na AM, o problema do mundo real deve ser apresentado de forma abstrata e compreensível para as máquinas, para que possa ter o entendimento do problema. Dados como velocidade e distância, por exemplo, são transformados em valores numéricos analíticos, que por sua vez, são tabulados e armazenados. O valor de um determinado bem, por exemplo a de um imóvel pode sofrer variação devido diversos parâmetros como às condições deste, localidade, quantidade de quartos, vagas na garagem, idade da obra, entre outros.

A etapa de processamento dos dados adquiridos do mundo real é denominada como pré-processamento, como já apresentada na Sessão 2.2 . A etapa de pré-processamento tem a entrada de dados que são as informações adquiridas no mundo real, após o pré-processamento a saída de dados são conjuntos de exemplos, que foram transformados para que a máquina possa entender e serem utilizados para a fase de treinamento. Os dados que foram abstraídos do mundo real é convertidos a linguagem de entendimento das máquinas e posteriormente são rotulados de acordo com o problema de predição para o qual foram indicados. Um rótulo é a característica adquirida a partir dos dados processados que se deseja ter um maior conhecimento, acarreta ou não a dispensa a rotulação manual de um especialista. Por exemplo, a classificação de plantas com relação ao tipo de superfície das folhas, problema no qual são utilizadas as classes GLABA, PILOSA, LISA e RUGOSA. Para a obtenção dos rótulos das plantas, basta saber se a folha tem ou não tricomas (semelhante à pêlos), se o limbo é liso ou enrugado da folha e definir o adito para agrupá-los e atribuir o mesmo rótulo as folhas com tipo de superfície entre dois áditos.

O algoritmo de AM supervisionado faz uma busca para encontrar padrões para representar as possíveis classes do problema apresentado, e desta maneira, os padrões reconhecidos podem passar para um especialista para análise e extração de padrões interessantes para o problema apresentado. Para a etapa de Rotulação tem a entrada exemplos e a saída são os exemplos já rotulados, desta forma, pode-se seguir em frente com diferentes algoritmos de AM supervisionado para a indução de Classificadores. Os classificadores são capazes de rotular os exemplos que são diferentes daqueles já apresentados ao classificador durante a etapa de indução, isto é, o conjunto de teste. A entrada dos classificadores é o conjunto de exemplos, e saída consiste em classes que categorizam cada um dos exemplos, especialmente em problemas de predição, por exemplo, para a bolsa de valores como as ações. Os algoritmos de AM mais conhecidos e usados para a predição são os Vetores de Suporte (SVM) e K-vizinho Mais Próximo (KNN).

2.3.1 Máquina de Vetores de Suporte (SVM)

As máquinas de vetores de suporte (*support vector machine* – SVM) está cada vez mais valorizada dentro da comunidade de AM. Os resultados da aplicação desta técnica mostram uma grande superioridade em cima dos demais algoritmos mais populares de aprendizado, tais como as RNAs (Rede Neurais Artificiais). O SVM possui uma grande vantagem, pois suas aplicações são multidisciplinar, tais como, gerenciamento de relacionamento com o cliente, reconhecimento facial ou de outras imagens, bioinformática, extração de conceito de mineração de texto, detecção de intrusão, predição de estrutura proteica e reconhecimento de voz. Os vetores de suporte são sustentados pela teoria de aprendizado estatístico que foi desenvolvido e implementado por (VAPNIK, 1995) através dos estudos de (CHERVONENSKI A. Y.; VAPNIK, 1971). Levantada a teoria que é estabelecida por uma série de princípios, que devem ser sequenciais para a obtenção de classificadores com alta capacidade de generalização. As SVM pode ser consideradas o estado da arte em diversas tarefas que tem como propósito a classificação e regressão. São baseadas na Teoria de Aprendizagem Estatística (TAE) de (VAPNIK, 1995).

Para um melhor entendimento, consideramos h como um classificador e H o conjunto de todos os classificadores que um algoritmo pode gerar. Durante o processo de aprendizado utiliza-se um conjunto de treinamento X , composto por n pares de (x_i, y_i) para gerar classificadores particulares h pertencente a H . A teoria de aprendizado estatístico contribui na escolha de um classificador partindo do conjunto de dados de treinamento. Algumas considerações deve ser feitas sobre a escolha de um classificador, que ao aplicar a TAE, tende-se assumir que os dados do domínio em que o aprendizado esta sendo processado os classificadores vão ser gerados de forma independente e identicamente distribuída, que de acordo com uma distribuição da probabilidade $P(x, y)$, descrita conforme a relação de objetos e os seus respectivos rótulos. O risco já esperado do classificador h para todos desse domínio é representado pela a expressão 2.1. (MÜLLER, 2001).

$$R(h) = \int (c(h(x), y) dP(x, y) \quad (2.1)$$

Onde, $c(h(x), y)$ é uma função de custo, representada pela equação 2.2, que geralmente surgem em problemas de classificação.

$$c(h(x), y) = \frac{1}{2} |y - h(x)| \quad (2.2)$$

A função de custo definida pela Equação 2.2 retorna o valor 0 se X é classificado e 1 caso contrário.

Pode-se usar a minimização do risco esperado sobre os dados de treinamento. Fazendo o uso do princípio da indução para inferir uma função h que minimize o erro sobre os dados

de treinamento, e tem como esperado um procedimento apazível e também gere a menor quantidade de erros possíveis nos novos dados. Essa estratégia é adotada na maioria dos algoritmos de AM supervisionados. Outro risco conhecido é o empírico, em que pode ser representado por h que é provido pela Equação 2.3, no qual, é responsável por medir o desempenho dos classificadores nos dados de treinamento, através de taxa de classificações incorretas obtidas em X .

$$R_{emp}(h) = \frac{1}{n} \sum_{i=1}^n c(h(X_i), y_i) \quad (2.3)$$

O processo de indução com base de dados para treinamento conhecidos, estabelece o princípio de minimização do risco empírico (SCHOLKOPF, 2002). Com $n \rightarrow \infty$, torna-se possível fundar condições para que o risco empírico converge para o um risco esperado (SCHOLKOPF, 2002). Em casos em que a base de treinamento forem sucintos, não pode sempre trazer isso como garantia, isto é, nem sempre a minimização do risco empírico pode trazer o menor risco esperado. Tomamos por exemplo um classificador binário que memoriza todos os objetos de treinamento e faz as classificações aleatórias para outros exemplos (SHUURMANS, 1999). Embora seu risco seja Nulo, o esperado é 0.5.

Um limite esperado importante é apresentado pela TAE que relaciona o risco esperado de uma função ao seu risco empírico à um termo de capacidade. Esse limite é apresentado na Equação 2.4, em que é garantido a probabilidade $1 - \theta$, em que $\theta \in [0, 1]$.

$$R(h) \leq R_{emp}(h) + \sqrt{\frac{VC(\ln(\frac{2n}{VC}) + 1) - \ln(\frac{\theta}{4})}{n}} \quad (2.4)$$

Para a Inequação 2.4, VC denota a dimensão Vapnik-Chervonenkis (VAPNIK, 1995) da classe de funções H à qual h faz parte, n representa a quantidade de exemplos no conjunto de treinamento T e a parcela de raiz na soma é referenciada como termo de Capacidade. A dimensão VC mede a capacidade do conjunto de funções H , quanto maior o seu valor, mais complexas são as funções de classificação que podem ser induzidas a partir de H . Dado um problema de classificação binário, essa dimensão é definida como o número máximo de exemplos que podem ser particionados em duas classes pelas funções contidas em H para todas as possíveis combinações binárias desses dados.

Para funções de decisão lineares do tipo $h(x) = w \cdot x$ perdura resultados alternativos que relacionam o risco esperado ao conceito de margem. A margem de um objeto de exemplo tem relação com sua distância à fronteira de decisão induzida, sendo uma medida da confiança da previsão do classificador. Para um problema binário, temos $y_i \in -1, +1$, dada uma função h e um exemplo x_i , a margem $\varrho(h(x_i), y_i)$ com que esse dado é classificado por f pode ser calculada pela Equação 2.5. Prontamente, se o valor for negativo de $\varrho(x_i, y_i)$ implicará em uma

classificação incorreta.

$$\varrho(h(x_i), y_i) = y_i h(x_i) \quad (2.5)$$

Para obter a margem geométrica de um x_i em relação a uma fronteira linear $h(x) = w \cdot x + b$, o qual mede efetivamente a distância de x_i à fronteira de decisão, divide-se o termo à direita da Equação 2.5 pela norma de w , ou seja, por $|w|$. Para exemplos que foram classificados de forma errônea, o valor adquirido equivale à distância com sinal negativo. Para realizar uma diferenciação, tomamos a margem da Equação 2.6 será identificada como margem de confiança. A partir do conceito conhecido, torna-se possível então definir o erro marginal de uma função $h(R_\rho(h))$ sobre um conjunto de treinamento. Esse erro propicia a proporção de exemplos de treinamento cuja margem de confiança é inferior a uma determinada constante $\rho > 0$.

$$R_\rho(h) = \frac{1}{n} \sum_{i=1}^n I(y_i h(x_i) < \rho) \quad (2.6)$$

Na Equação 2.6, $I(q) = 1$ se q é verdadeiro e $I(q) = 0$ se q é falso. Uma constante c tal que, a probabilidade $1 - \theta \in [0, 1]$, para todo $\rho > 0$ e H correspondendo à classe de funções lineares $f(x) = w \cdot x$ com $\|x\| \leq e \|w\| \leq 1$, demonstrado pela Equação 2.7, o seguinte limite se aplica (SHUURMANS, 1999):

$$R(h) \leq R_\rho + \sqrt{\frac{c}{n} \left(\frac{R^2}{\rho^2} \log^2\left(\frac{n}{\rho}\right) + \log\left(\frac{1}{\theta}\right) \right)} \quad (2.7)$$

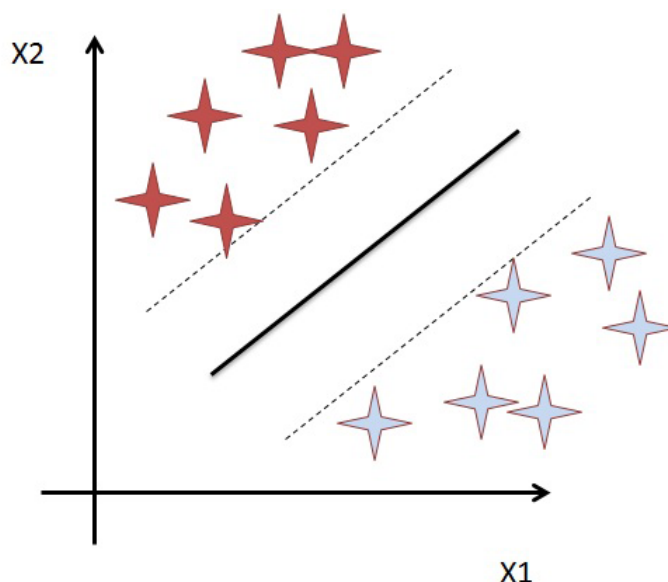
Como na Inequação 2.4, tem-se na Expressão 2.7 novamente o erro esperado limitado pela soma de uma medida de erro no conjunto de treinamento, o erro marginal, a um termo de capacidade. A interpretação do limite é de que quanto maior a margem ρ , implicará em uma menor capacidade. Contudo, a maximização da margem pode causar o aumento da taxa de erro marginal, pois torna-se mais difícil de obedecer às restrições de todos os dados de treinamento para aqueles distantes da margem maior em relação ao hiperplano separador. Se aplicado um baixo valor para ρ , implicará a um erro marginal menor, mas isto fará com que aumente o termo de capacidade. Obrigatoriamente, deve-se buscar um compromisso entre a maximização da margem e a obtenção de um erro marginal baixo.

Ao minimizar o risco para os classificadores, para que seja feita a classificação, um novo exemplo é mapeado para um ponto no hiperespaço e recebe a classe de acordo da sua posição em relação ao limiar de decisão, representadas na Figura 2. Para obras encontradas na literatura sobre predição de 'quotes' da bolsa, alguns chegam a aplicar SVM na indução dos classificadores para classes discretas, como (CHOUDHRY, 2008) e (WANG, 2007).

(CHOUDHRY, 2008) faz a representação dos atributos com indicadores técnicos e correlação entre ações e realiza seleção de atributos com algoritmo genético. Seus resultados

chegaram a proximadamente 61% de taxa de acerto. Já (WANG, 2007) tenta comparar o desempenho das SVM com e *Autoregressive Integrated Moving Average*, representando os atributos com séries temporais financeiras. Nesse caso, o SVM superou os outros dois métodos com 1.82% e 83.33%, de *Root Mean Square Error* e Estatísticas de Direção respectivamente.

Figura 2 – Ilustração de hiperplano canônicos e separador.



Fonte: <https://goo.gl/Ajho6b> - Wikipédia.

Um método matemático para realizar a transformação de curvas de classificação para a linha do hiperplano é conhecido como kernel, este método é realiza um mapeamento a ponto de simplificar o modelo. Um kernel é uma função que recebe dois pontos, em um espaço e realiza o cálculo do produto escalar dos objetos no espaço. Normalmente utiliza-se um kernel quando não se tem o conhecimento do plano. A simplicidade por seu cálculo e capacidade de representar espaços abstraídos fazem com que seja muito utilizado. A escolha do kernel afeta diretamente o desempenho do classificador. Os kernels mais usados são o Polinomial e RBF.

2.3.2 Máquina de Vetores de Suporte: Regressão (SVR)

A SVM são utilizadas em também em problemas de regressão, apesar de não se tratar de algoritmo supervisionado, o algoritmo SVR (*support vector regression* (VAPNIK, 1995) utiliza-se de uma função $h(x)$ que tenha como *output* de forma continua para os dados de treinamento desviem ao máximo de E do rotulo desejado, de forma que, a função seja a mais regular e uniforme possível. O problema de otimização, procura uma regularidade que pode ser encontrada por com a minimização da norma de $\|w\|$, demonstrada na Equação 2.8.

$$Min = \frac{1}{2} \|w\|^2 \quad (2.8)$$

Com as restrições:

$$\begin{cases} y_i - w \cdot x_i - b \leq \varepsilon_i \\ w \cdot x_i + b \leq \varepsilon_i \end{cases} \quad (2.9)$$

2.3.3 1-vizinhos Mais Próximos (1NN)

Para dar o entendimento ao K-NN, temos que conhecer o algoritmo 1-NN. Neste Algoritmo, cada objeto representa um ponto em um espaço definido pelos atributos denominado por espaço de entrada. Se definido uma métrica nesse espaço torna-se possível calcular as distâncias entre dois pontos. A métrica mais usual para realizar o cálculo é a distância Euclidiana, dada pela Equação 2.10, em que x_i e x_j são dois objetos representados por vetores no espaço \mathcal{R}^d , e x_i^l e x_j^l são elementos desses vetores, que correspondem aos valores da coordenada l , os atributos.

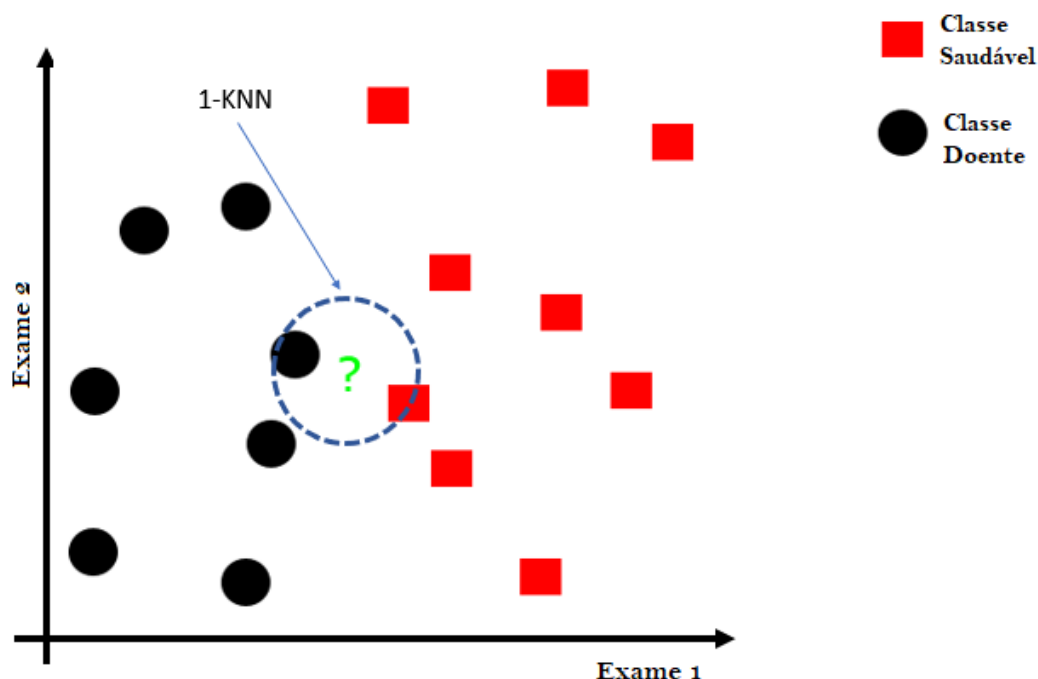
$$d(x_i, x_j) = \sqrt{\sum_{l=1}^d (x_i^l - x_j^l)^2} \quad (2.10)$$

O algoritmo de 1-NN é de fato bem simples. Na fase de treinamento o algoritmo memoriza os exemplos já rotulados do conjunto de treinamento. Para realizar a classificação de um exemplo não rotulado, ou seja, aquele que faz parte do conjunto de testes e sua classe não é conhecida, é calculada a distância entre o vetor de valores de atributo dos exemplos já rotulado em memória. O rótulo de classe a ser associado, sempre será ao exemplo de treinamento mais próximo do exemplo de teste e utilizado para classificar os novos exemplos. A Figura representa um exemplo do algoritmo de 1-NN. Neste exemplo, demonstraremos que existe dois tipos de objetos que podem ser classificados, os saudáveis e os doentes. A entrada para o algoritmo seria o resultado de dois atributos que são 2 exames realizados pelos pacientes. O ponto classificado como “?” é o objeto de teste, que será classificado pelo algoritmo. Os demais pontos são objetos já conhecidos e classificados, cujo os saudáveis são os quadrados e os doentes os círculos. Utilizando a distância Euclidiana, o objeto de treinamento mais próximo do que deve ser classificado é um doente, que é então atribuído ao objeto de teste.

2.3.4 K-vizinhos Mais Próximos (KNN)

O K-vizinhos mais próximos, proposto por (ALBERT, 1991), é um algoritmo de aprendizado de máquina simples, baseado em instâncias e seu processamento é moroso até o momento da classificação. É dito simples porque na etapa de treinamento apenas armazena os exemplos, diferentemente do SVM por exemplo, que constrói um modelo indutor com os exemplos e o utiliza para classificar novos exemplos. O KNN é uma extensão imediata ao algoritmo 1-NN, que em vez de ter apenas 1 vizinho, trata-se de k vizinhos, que são objetos do conjunto de treinamento mais próximos do ponto de teste X_t , em que k é um parâmetro do algoritmo. Para quando o valor de K é maior de 1, para cada ponto de teste, são obtidos k

Figura 3 – Exemplo ilustrativo do algoritmo 1-NN.



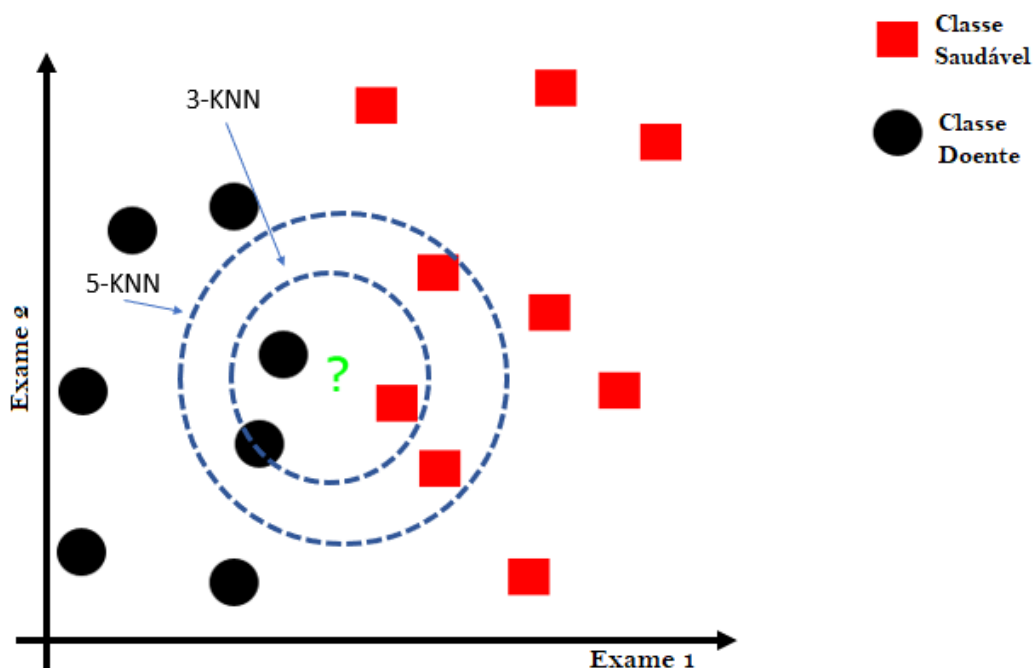
Fonte: Elaborada pelo autor.

vizinhos. Para cada vizinho pode-se atribuir um peso para a classe. As previsões dos diferentes vizinhos agregam de certa forma, a tarefa de classificar o ponto de teste. A agregação é realizada de divergentes formas para os problemas de regressão e classificação. Em caso de problemas de classificação, a classe recebe valores de conjuntos discreto, e que cada vizinho vota em uma classe. O objeto de teste então é classificado à aquela classe mais votada. Em termos matemáticos, esse processo é dito por $f(x_t) \leftarrow \text{moda}(f(x_1), f(x_2), \dots, f(x_k))$, o qual pode ser justificado por que a constante que minimiza a função de custo $0 - 1$ é a moda. Em caso dos problemas de regressão pode-se utilizar duas estratégias dependendo da função de custo usada. Se a função de custo minimizar o erro quadrático, deve-se utilizar a média dos valores obtidos para cada um dos k vizinhos, formalmente obtemos $f(x_t) \leftarrow \text{média}(f(x_1), f(x_2), \dots, f(x_k))$. Segundo caso, se a função de custo a ser considerada é do desvio absoluto, neste caso, deve ser utilizar a mediana, formalmente temos $f(x_t) \leftarrow \text{mediana}(f(x_1), f(x_2), \dots, f(x_k))$. A justificativa para o uso deste processo é devido a média é a constante que minimiza o erro quadrático, enquanto temos a mediana para minimizar o desvio absoluto.

Utilizando o mesmo caso da figura 3, iremos propor que o valor de k seja 3, assim o objeto de teste seria classificado seria classificado como “doente”, mas o valor proposto for k igual a 5, o objeto seria classificado como “saudável”, representado pela Figura 4.

A escolha de um valor para k mais apropriado para um caso de decisão específico pode não ser trivial. O valor de k geralmente é ímpar e baixo: $k = 1, 3, 5, \dots$. Em problemas

Figura 4 – Impacto do valor de k no algoritmo K-NN.



Fonte: Elaborada pelo autor.

de classificação deve-se evitar usar números com valores pares, por exemplo, $k = 2, 4, \dots$, dificultando a geração de empates.

Uma estratégia que pode ser utilizada para que diminua o risco de empates, é a utilização de pesos. Associar um peso à contribuição de cada vizinho. A contribuição de cada um dos k vizinhos é pesada de forma inversamente proporcional a distância ao ponto de teste. Assim, é possível utilizar o valor de k seja n (todos os objetos de treinamento).

2.3.5 K-vizinhos Mais Próximos (KNN): Aspectos positivos

- O algoritmo é simples
- O k-NN constrói aproximações locais da função objetivo, diferentes para cada novo dado a ser classificado. Essa característica pode ser vantajosa quando a função objetivo é complexa, mas ainda pode ser descrita por uma coleção de aproximações locais de menor complexidade (MICHELL, 1997)
- Aplicável em problemas complexos
- É incremental: quando novos exemplos de treinamento estão disponíveis, basta armazená-los na memória.

2.3.6 K-vizinhos Mais Próximos (KNN): Aspectos negativos

É um algoritmo preguiçoso “*lazy*” o que não obtém uma representação compactada dos objetos. Durante a fase de treinamento não se exige muito esforço computacional. Porém classificar um objeto de teste, é necessário realizar o cálculo da distância para todos os objetos de treinamento, sendo assim, torna-se a predição um procedimento de alto custo, que em casos de grande volume de objetos de treinamento o processo torna-se demorado. Outro problema deste algoritmo é a dimensionalidade dos exemplos, isto é, quanto maior número de atributos, maior será definido o número de dimensões do espaço, ou seja, o espaço que é definido pelos atributos de um problema cresce de forma exponencialmente ao número de atributos.

2.4 Bolsa de Valores

Bolsa de Valores é o nome dado ao mercado organizado que se realiza negociações de ações de empresas que possuem o capital aberto. A primeira bolsa de valores, mais parecida com as atuais, foi na Bélgica no século XV durante a expansão comercial. Vendedores e comerciantes se reuniam para a troca de Moedas e metais preciosos. A ideia foi difundida pela Europa durante a Revolução Comercial. O comércio de ações teve início no século XIX. Nas primeiras décadas do Século XX, as negociações de ações eram feitas por “Pregão”, nome dado a gritaria dos operadores. Recentemente foi introduzido ao mercado de ações o *trading* algorítmico que utiliza plataformas eletrônicas para ordens de compra e venda. O algoritmo executa instruções de negociação pré-programadas.

2.4.1 National Association of Securities Dealers Automated Quotations (NASDAQ)

National Association of Securities Dealers Automated Quotations conhecida como NASDAQ é a das maiores bolsa de valores do mundo, e a segunda maior dos Estados Unidos, ficando atrás apenas da NYSE (*New York Stock Exchange*), sua diferença é que realiza todas as operações por meio eletrônico. Considerada a mais moderna do mundo devido suas atividades serem realizadas 100% sem a utilização de operadores, devido a forma de operação serem realizadas por transações automatizadas pela internet. As negociações são feitas igualmente como são realizadas em outras mais antigas como NYSE, apesar do seu modo de operação. As *quotes* mais encontradas na bolsa NASDAQ são empresas de pequeno e médio porte de capitalização, o que é historicamente entendido por atrair as empresas de tecnologia.

Fundada em 1971, após um estudo elaborado pela *United States Securities and Exchange Commission* (SEC) a NASDAQ pelo presente motivo de que as operações de alguns setores, principalmente as de tecnologia, tinham como necessidade uma maior regulamentação e transparência, pois então, foi criado o documento em que recomendava que o controle e

sua execução fosse realizada por meios de dispositivos eletrônicos. Dessa maneira, desde sua fundação a NASDAQ é uma bolsa de valores totalmente virtual.

Com exceção da empresa que desenvolveu um dos mais famosos aplicativos do mercado o *Snapchat*, a Snap Inc. não possui as ações na NASDAQ, e sim na NYSE, todas as as outras empresas de tecnologia de grande capital, como a *Facebook* deram a preferência por utilizar a NASDAQ. Segundo o relatório da *World Federation of Exchanges* de 2016, a soma de valores de mercado de mais de 2400 empresas listadas na NASDAQ é de US\$ 7,3 trilhões.

2.5 Ferramentas Computacionais

Para desenvolver a aplicação, foram realizadas uma série de pesquisas para selecionar as ferramentas ideais. Levando em consideração que a aplicação seja multiplataforma e que possa vir a ser utilizado em todos os Sistemas como *Windows*, *Linux* e *Macintosh*, as linguagens de programação selecionadas são orientadas para este propósito e são bem difundidas na comunidade computacional para tal propósito. Para um melhor entendimento desta seleção foi dividido em subcapítulos e cada deste possui o resumo sobre as suas funções.

2.5.1 *Python*

A linguagem de programação *Python* é atualmente a mais difundida do mundo, pela facilidade de aprendizagem, simplicidade e a grande quantidade de bibliotecas para desenvolvimento. Afins de estudo da AM, *Python* mostrou-se uma ferramenta importante para criação de aplicações. É uma linguagem multiplataforma que funciona igualmente bem em plataformas como *Windows*, *Linux* e *Macintosh*. A linguagem pode ser utilizada para desenvolver diversas aplicações de pequeno porte e protótipos rápidos, mas escala bem para permitir o desenvolvimento de programas robustos e complexos. O *Python* possui bibliotecas para o desenvolvimento de aplicações de AM de alto nível, além de, a facilidade de criação de *Interface* para uma melhor experiência para o usuário.

2.5.2 *wxFormBuilder*

O *wxFormBuilder*, é uma ferramenta que foi apresentada recentemente e recomendada pela comunidade por agregar facilidade para a criação de GUIs. A ferramenta agiliza o processo de criação de interfaces para diversas linguagens que são multiplataforma utilizando a aparência nativa do sistema operacional, a IDE utiliza a biblioteca *wxPython* para aplicações na linguagem *Python*.

3 Sobre a Aplicação

Baseado no método de problemas de otimização Lineares e Não Lineares, tais como, SVM e SVR, a apresentação da construção do trabalho foi dividida em quatro etapas. A primeira etapa trata-se da escolha do algoritmo de aprendizagem e qual a melhor forma de desenvolvimento. A segunda, é a escolha dos dados de treinamento, e a escolha do objetivo foco da aplicação, cujo é a Bolsa de Valores, e as ações do mercado financeiro. A terceira etapa, trata-se da escolha da linguagem de programação para melhor desenvolvimento do método já selecionado durante a primeira etapa do projeto. E por fim, a última etapa trata-se da solução encontrada para um melhor entendimento das informações necessárias para o realizar a predição, e apresentação das informações para o usuário final, isto é, a criação de uma interface gráfica que transmita a fácil compreensão.

3.1 Método escolhido

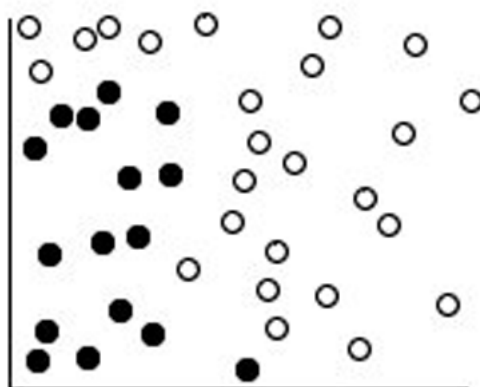
Qualquer pessoa que deseja fazer um investimento na Bolsa de Valores deve ter capital para tal, e ter consciência de que aquele dinheiro investido pode gerar lucros e prejuízos. A utilização de AM para predir uma ação na bolsa é uma técnica que pode servir bem, se usada de forma correta. A SVM (*Support Vector Machine*) é uma ótima opção para realizar este tipo de predição, e está na categoria de AM supervisionado, que realiza análises dos dados e reconhecimento de padrões, a fim de serem utilizados para a classificação e análise de regressão.

O SVM tem o *input* um conjunto de dados e de forma que realize a predição para cada *input*, no qual faz a identificação das duas possíveis classes aquele *input* faz parte. Pode-se dizer que este método é considerado um classificador linear binário não probabilístico. A criação do modelo SVM classifica o objetos do conjunto de treinamento como pertencente a uma de duas categorias, então, o algoritmo de treinamento do SVM cria um modelo, que servirá para atribuir aos novos exemplos as categorias.

Como já visto na Figura 2, a ideia consiste em encontrar uma linha de separação em um hiperplano entre duas classes. Essa linha busca maximizar a distância entre os pontos mais próximos em relação a cada uma das classe. A distância entre o hiperplano e o primeiro ponto de cada classe é a margem. Inicialmente, a SVM realiza a classificação das classes definindo qual classe cada ponto pertence. E posteriormente faz a maximização da margem.

A escolha de *Support Vector Machine* (SVM) para o projeto foi realizada por se tratar de uma técnica robusta de classificação e regressão que maximiza a precisão preditiva de um modelo, e particularmente é adequado para realizar análise de dados com números muito grandes de campos preditores. Por exemplo, um conjunto dado é representado pela a figura 5.

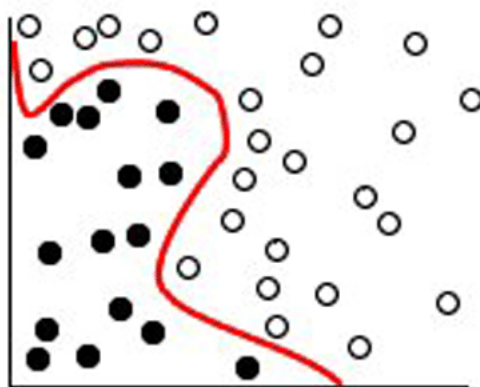
Figura 5 – Ilustração do conjunto de dados.



Fonte: IBM SPSS Modeler 17.1.0.

Como já descrito anteriormente, o conjunto de dados passa pela a classificação dos itens de teste, e após o entendimento cria-se uma curva de separação, representado pela figura 6.

Figura 6 – Ilustração do conjunto de dados com curva de separação.



Fonte: IBM SPSS Modeler 17.1.0.

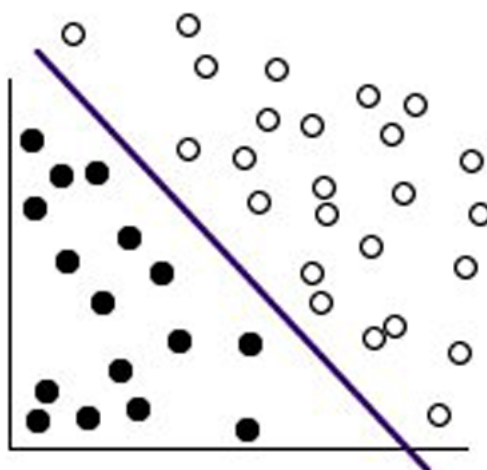
O processamento do algoritmo SVM, faz com que após a classificação, o tratamento realizado nos dados do conjunto de treinamento gere um linha de separação no hiperplano, representado pela a figura 7. A função matemática usada para realizar a classificação é conhecida como Kernel, no qual, três foram selecionados para serem aplicadas no desenvolvimento do protótipo, tais como:

- Linear

- Polinomial
- Função de Base Radial

O objetivo do kernel é encontrar um equilíbrio quase que ideal entre uma margem larga e um número pequeno de pontos de dados classificados incorretamente. A função kernel possui um parâmetro de regularização que controla e realiza os *trade-off* entre esses dois valores.

Figura 7 – Ilustração do conjunto de dados com linha de separação do hiperplano.



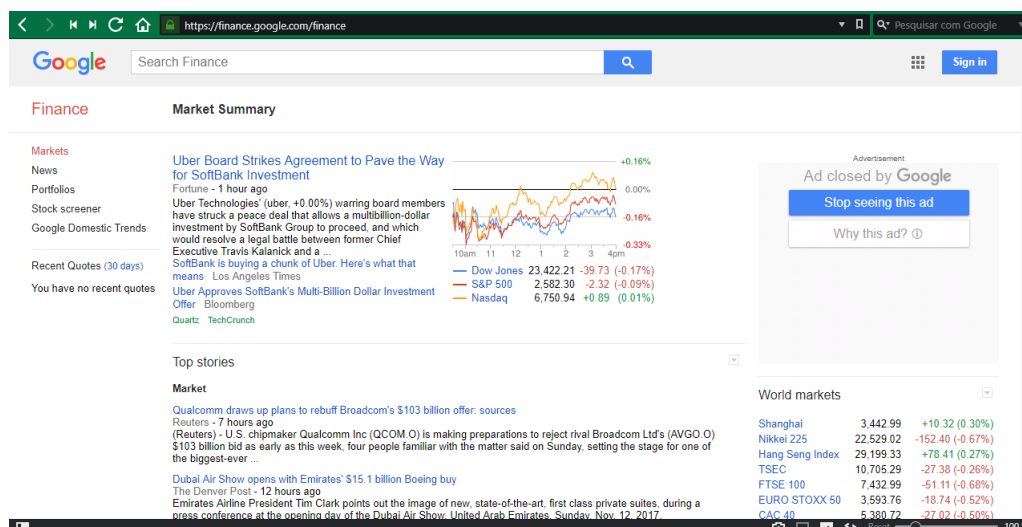
Fonte: IBM SPSS Modeler 17.1.0.

3.2 Dados de Treinamento

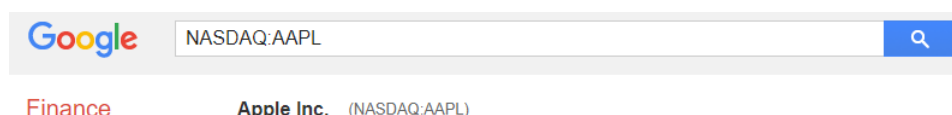
Esta etapa é de grande importância, pois trata-se do coração da aprendizagem de máquina, os dados que compõem o conjunto de exemplos de treinamento. A predição de dados é considerada importante ferramenta para o grande volume de dados que são criados todos os dias. Inicialmente, o propósito do projeto trava-se de auxílio as grandes corporações com o uso da predição de dados destinada área de atuação da mesma. Por fim, a fins de estudo e conhecimento, a escolha foi mais genérica e visa a o auxílio a qualquer pessoa que deseja investir, através de análises de dados anteriores do valor de uma ação em específico da bolsa de valores NASDAQ e realizar predições para um futuro investimento, já que, a aplicação do investimento realizado de forma não tão randômica cria-se maior conforto para investir e gerar mais lucros do que prejuízos.

Os dados selecionados para o projeto são adquiridos pela Internet, no site do *Google Finance*, demonstrado pela Figura 8 que mostra a página inicial de onde foi realizada a extração das informações.

No campo de pesquisa, é adicionado a Bolsa de Valores que deseja e a ação, como é conhecida no mercado (por exemplo, AAPL (*Apple*)), como é demonstrado na Figura 9.

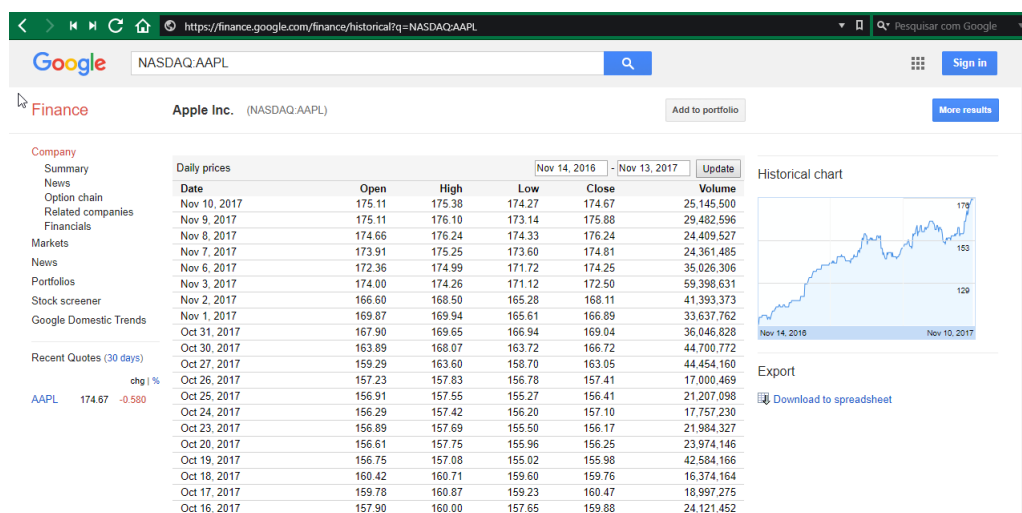
Figura 8 – Tela Inicial do *Google Finance*.

Fonte: Elaborada pelo autor.

Figura 9 – Campo de Pesquisa do *Google Finance*.

Fonte: Elaborada pelo autor.

Os dados podem ser baixados na aba 'Historical', e clicando no campo 'baixar dados...', como demonstrado na figura 10, os dados são coletados e salvos no formato *Comma-separated values* (CSV), que são arquivos de texto que realiza uma ordenação de *bytes* ou um formato de terminador de linha.

Figura 10 – *Download* dos arquivo CSV através do *Google Finance*.

Fonte: Elaborada pelo autor.

3.3 Desenvolvimento

Esta etapa visa a escolha da tecnologia a ser usada para o desenvolvimento e como utilizada. Linguagens de programação como C, Java e HTML são consideradas mais atrativas e tradicionais para a comunidade. Contudo, existe uma linguagem de programação que particularmente trata-se de um modelo mais limpo e de fácil entendimento e produção para programadores, a linguagem *Python*. O crescente uso desta fez com que emergisse uma volumosa quantidade de bibliotecas que podem e auxiliam no desenvolvimento de qualquer tipo de programa, desde simples protótipos à programas mais robustos e complexos com dificuldade de desenvolvimento em outras tecnologias. O protótipo foi desenvolvido totalmente em *Python*, em sua versão 3.6.

Durante o desenvolvimento foram utilizadas bibliotecas como a *Numpy*, *WX*, *CSV* e *matplotlib*. As bibliotecas *CSV*, é responsável pela manipulação de arquivos com este formato, e *matplotlib* agrega recursos para a geração de gráficos 2D a partir de *arrays*, são nativos da linguagem *Python*. A *Numpy* e *WX* são pacotes que podem ser adquiridos por meio da *Internet*, para realizar o *download* e fazer o uso destas bibliotecas foram necessário entrar no 'prompt de comando (cmd) do *Windows* e instalar os pacotes através dos comandos: "*pip install numpy*" e "*pip install -u wx*", respectivamente.

3.3.1 Pacote *Numpy*

O *Numpy* é uma biblioteca do *Python*, que auxilia no trabalho com *arrays*, vetores e matrizes de N dimensões. A eficiência dos recursos da biblioteca são de alta qualidade, quando comparado a programa que fazem o mesmo. Algumas utilidades do pacote são listadas abaixo, vale frisar que não são todas:

- Objeto array para a implementação de arranjos multidimensionais
- Objeto matrix para o cálculo com matrizes
- Ferramentas para Aprendizagem de Máquina
- Transformadas de Fourier básicas
- Ferramentas sofisticadas para geração de números aleatórios

3.3.2 Pacote *WX*

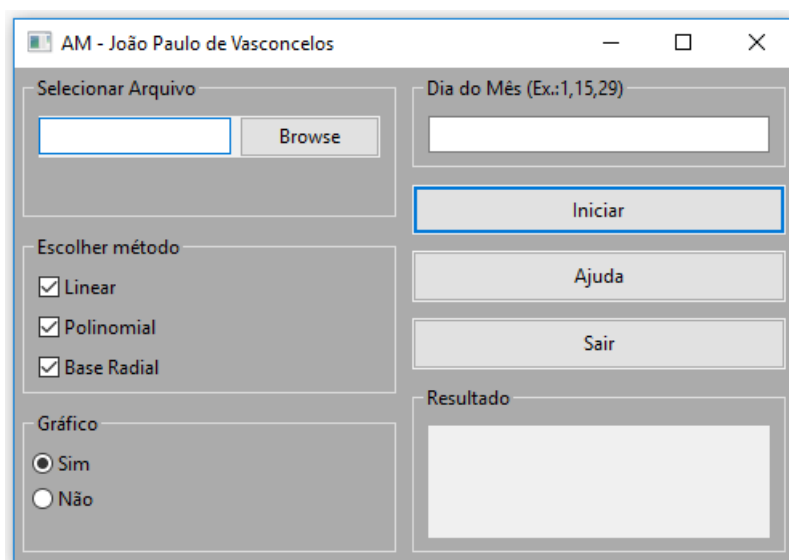
A *WX* é uma popular biblioteca que auxilia na criação de *Interface* (GUI), possuindo um *kit* de ferramentas. Abrange diversas outras tecnologias e não apenas o *Python*. Foi desenvolvido por Robin Dunn juntamente com Harri Pasanen, na linguagem de programação *C++*. O Pacote criado para o *Python* ficou conhecido como *wxPython*. O pacote auxilia

com os desenvolvimento da GUIs em geral, desde a criação do *Forms*, separadores, criações de botões, *labels*, entre outros.

3.4 Visualização: Interface gráfica do utilizador (GUI)

Esta etapa do projeto trata da apresentação das informações para o usuário, desde como foi realizada e quais recursos o usuário dispõe para navegar na aplicação. Para realizar uma predição o usuário deve possuir sua máquina o *Python* com os pacotes mencionados na sessão anterior (3.3). A GUI foi totalmente desenvolvida para trazer facilidade de entendimento ao método aplicado para predição. A tela inicial do programa é composta por objetos simples que transmitem um fácil entendimento, como mostra a figura 11.

Figura 11 – Tela Inicial do Programa de Predição.



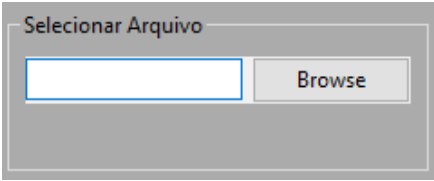
Fonte: Elaborada pelo autor.

Para um melhor entendimento, será demonstrado as etapas para realizar uma predição utilizando a GUI desenvolvida. Para dar inicio a predição das *quotes* (ações) o usuário deve-se clicar no botão "*Browser*" e selecionar o arquivo *CSV* que possui os dados que compõe o conjunto de treinamento do algoritmo. O campo a ser preenchido é demonstrado na Figura 12. Frizando que, o arquivo *CSV* pode ser adquirido no *Google Finance*, como já mencionado na Sessão 3.2 sobre Dados de Treinamento.

O usuário deve selecionar a função kernel do modelo a ser tratado pelo algoritmo SVM, pode ser selecionado entre Linear, Polinomial e Base Radial. Todos os *CheckBox's* podem ser selecionados para participarem do processo de predição, vide a Figura 13, e de forma binária, se deseja ter o gráfico do modelo selecionado.

Um dia que deseja realizar a predição de dados precisar ser escrito no campo "Dia do Mês", por fim, para realizar a predição o usuário deve clicar no botão "Iniciar", vide Figura 14.

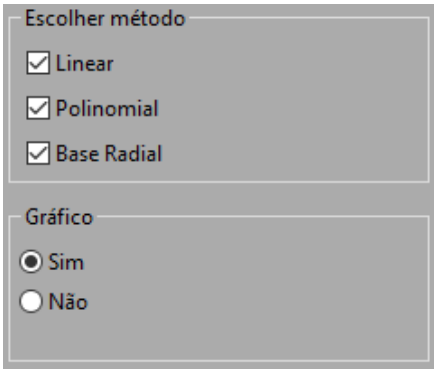
Figura 12 – PASSO 1: Seleção do arquivo para o Programa de Predição.



The image shows a small window titled "Selecionar Arquivo". Inside, there is a text input field with a blue border and a button labeled "Browse" to its right.

Fonte: Elaborada pelo autor.

Figura 13 – PASSO 2: Seleção do método do modelo para o Programa de Predição.

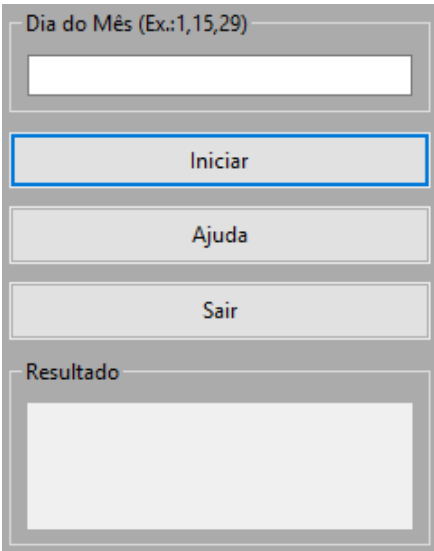


The image shows a window titled "Escolher método". It contains three checkboxes, all of which are checked: "Linear", "Polinomial", and "Base Radial". Below these, there is a section titled "Gráfico" with two radio buttons: "Sim" (which is selected) and "Não".

Fonte: Elaborada pelo autor.

Quando houver o termino da predição, será apresentado um gráfico na tela, e no campo de "Resultado" o valor da predição da ação para o dia escolhido.

Figura 14 – PASSO 3: Escolha do dia do mês e execução do Programa de Predição.



The image shows the main application window. At the top, there is a label "Dia do Mês (Ex.:1,15,29)" above a text input field. Below this are three buttons: "Iniciar", "Ajuda", and "Sair". At the bottom, there is a section titled "Resultado" containing a large, empty rectangular area for displaying the prediction result.

Fonte: Elaborada pelo autor.

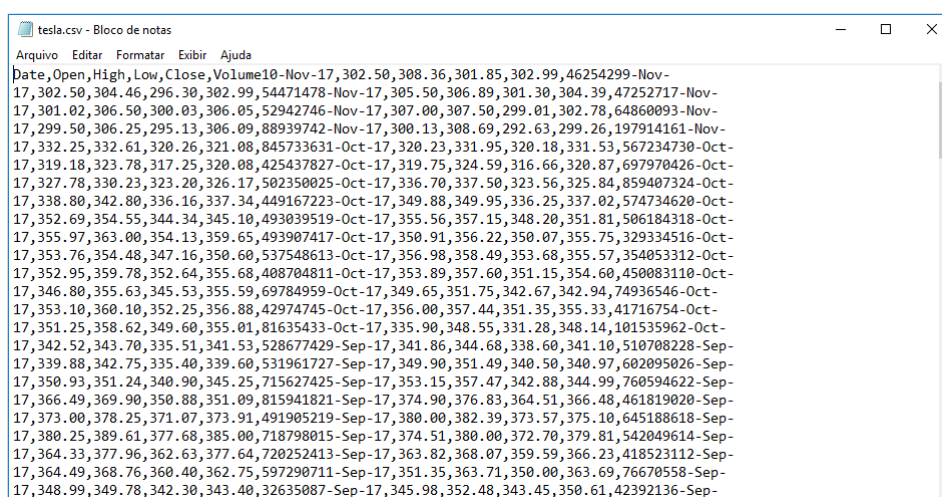
Foi desenvolvido um botão de "Ajuda" para auxiliar o usuário para o preenchimento dos dados e não ter nenhum problema para realizar a predição. Este botão abre uma nova janela que possui tais instruções de auxílio, e o botão "Sair" que fecha a aplicação.

3.5 Exemplo: Tesla, Inc.

Para confirmação de que é possível predir os dados com o protótipo, e obtenção de resultados durante o desenvolvimento tomamos como exemplo a empresa '**Tesla, Inc.**'. A Tesla é uma empresa do setor automobilístico e de armazenamento de energia, localizada nos Estados Unidos da América, que desenvolve, produz e vende automóveis elétricos de alto desempenho, além de, baterias de alta performance. A empresa possui suas ações na Bolsa de Valores NASDAQ, e está cotada no dia 10 de Novembro de 2017 em US\$302,99.

Para realizar a predição para o dia 13 de Novembro de 2017 com a aplicação foi necessário obter o arquivo CSV, adquirido no *Site do Google Finance*, temos o CSV representado pela figura 15:

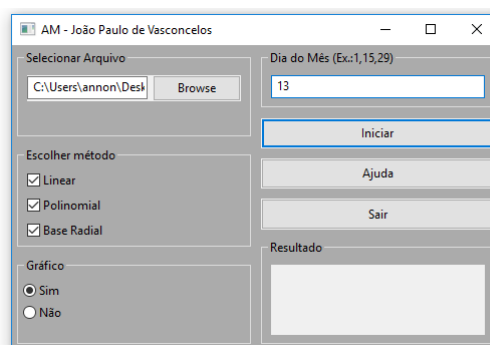
Figura 15 – Arquivo "Tesla.CSV"aberto pelo Bloco de Notas



Fonte: Elaborada pelo autor.

Coletadas as informações para o conjunto de treinamento, deve ser preencher todos os campos necessários da aplicação, tais como demonstrado na Figura 16.

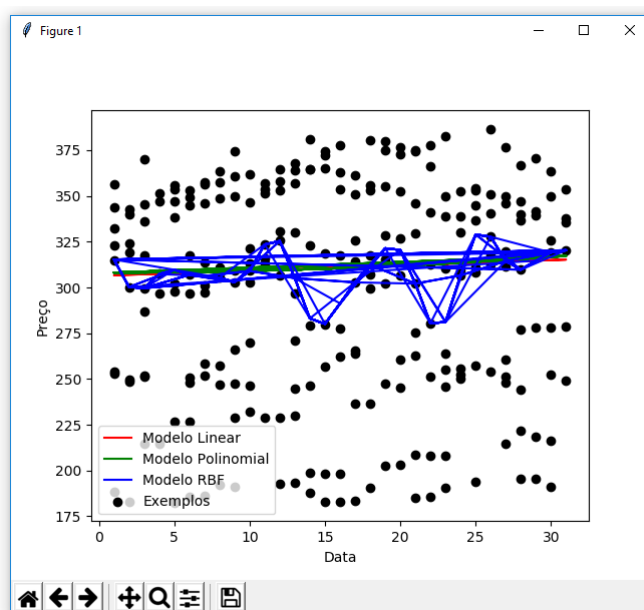
Figura 16 – A tela do inicial do programa com os dados preenchidos



Fonte: Elaborada pelo autor.

Com tudo preenchido, basta colocar o mouse sobre o botão Iniciar e clicar. Os dados preenchidos ficaram da seguinte maneira, o primeiro campo do CSV ficou com o caminho para o arquivo: *"tesla.csv"*; os *CheckBox's* de modelos a serem usados: *"Linear, Polinomial e Base Radial"*; opção de gráfico: *"sim"*; Dia do Mês: *"13"* (Frisando que leva em consideração o próximo dia 13 do ano). Com tudo preenchido ao clicar no botão "Iniciar" o algoritmo realiza as predições. Ao término dos cálculos realizados é apresentado o gráfico dos modelos selecionados, como demonstrado na Figura 17.

Figura 17 – Gráfico gerado após execução do Programa



Fonte: Elaborada pelo autor.

Como pode-se observar no gráfico, os objetos foram representados da seguinte maneira, os pontos pretos são os objetos do conjunto de treinamento e os traçados das cores Vermelha, Verde, Azul são as linhas de limite do hiperplano de cada um dos modelos selecionados durante a classificação do exemplo.

O resultado das predições realizada para o dia 13 de Novembro de 2017 da empresa Tesla, Inc., são demonstrados no campo destinado aos Resultados. Os resultados do exemplo são:

- Linear: US\$: 300.2299
- Polinomial: US\$: 306.9700
- Base Radial: US\$: 308.1375

Em conclusão, o kernel RBF é geralmente mais flexível do que os kernels lineares ou polinomiais, na medida em que pode-se modelar muito mais funções com seu espaço de função.

Deste modo pode-se considerar a Base Radial um resultado melhor para o conjunto de dados que foi utilizado durante o trabalho. Portanto, o melhor resultado encontrado foi 308.1375 da Base Radial.

4 Conclusão

Através do desenvolvimento realizados durante o projeto, de fato a utilização do método de *Support Vector Machine* pode ser considerada uma boa opção para realização as previsões de dados, em frente a outros métodos mencionados. A SVM é uma técnica realmente robusta em termos de classificação e regressão.

Como objetivo do trabalho foi construir um protótipo utilizando os estudos em análise de dados e utilizar a predição supervisionados de dados como *Support Vector Machine*, 1-vizinho Mais Próximo, K-vizinhos mais próximos. O uso de Aprendizagem de Máquina para prever dados do mercado como ações da bolsa de valores, cria-se um maior conforto a terceiros para realizar a aplicação do seu dinheiro de forma não randômica, e com uma base maior de segurança. A técnica usada para a classificação e predição, com princípios embasados na teoria do aprendizado estatístico, pode apresentar uma boa capacidade de generalização. As SVM suportam dados de grande dimensão sobre os quais outras técnicas de aprendizado comumente obtêm classificadores super ou sub ajustados. Entre os kernel utilizados, vale observar o desempenho do Base Radial para a geração do modelo, sendo uma ótima opção aos demais.

O estudo da Aprendizagem de Máquina é de grande importância para os dias atuais. A quantidade de dados geradas hoje é muito maior que de 10 anos atrás, e está com crescimento exponencial. A análise de dados é automatizada para serem aplicadas no desenvolvimento de modelos analíticos. O uso de algoritmos de AM faz com que os computadores aprendem interativamente a partir de dados sendo eles simples ou complexos. O aspecto de interatividade do AM é considerado de grande importância porque, conforme os modelos criados são expostos a novos dados, eles são capazes de se adaptar sem que tenha um especialista alterando, ou seja, é independente. Os computadores aprendem com os cálculos feitos anteriormente para produzir decisões e resultados confiáveis e reproduzíveis.

Além do conceito de grande destaque contemporâneo de um dos ramos da área de Inteligencia Artificial, a Aprendizagem de Máquina, temos a tecnologia que proporcionou garantir ótimas ferramentas para manipulação, construção e alteração de dados, além de, pacotes que permitem a melhor visualização dos modelos aplicados com gráficos e o conceito de multiplataforma. O protótipo pode ser rodado em qualquer plataforma *Windows*, *Linux*, *Macintosh*, desde que tenha um compilador Python.

A utilização aplicação não deve ser única para realizar investimentos na Bolsa NASDAQ, uma vez que apenas o histórico das *quotes* de uma empresa não é único fator para fazer o investimento em uma Bolsa de Valores. Os dados gerados pelo apesar de realizar uma predição através do histórico, a aplicação tem tem como objetivo o estudo educacional.

4.1 Trabalho Futuro

Devido o que foi apresentado durante o trabalho, como trabalhos futuros pode-se identificar ajustes para melhorar o protótipo. Estes trabalhos futuros que podem ser realizados temos:

- Otimizar o algoritmo de predição.
- Aumentar o número de kernels de regressão.
- Adicionar algoritmos de Aprendizagem de Máquina de predição para aumentar as opções do usuário.
- Suportar outras Bolsa de Valores além da NASDAQ.

Importante a expansão de conhecimento que esse projeto e os futuros podem proporcionar e uma ênfase no último item de "melhoramentos" que é de extrema importância para que o projeto torna-se internacional e que esteja o usuário possa investir em qualquer Bolsa de Valores.

Referências

- ALBERT, D. A. M. K. Instance-based learning algorithms. *Machine Learning*, 1991.
- CARVALHO, A. C. P. de Leon Ferreira de. In: *Inteligência Artificial: Uma abordagem de Aprendizado de Máquina*. [S.l.]: LTC, 2015. p. 129–133.
- CHERVONENSKI A. Y.; VAPNIK, V. N. On the uniform convergence of relative frequencies of events to their probabilities. 1971.
- CHOUDHRY, K. G. R. A hybrid machine learning system for stock market forecasting. *Proceedings of World Academy of Science, Engineering and Technology*, 2008.
- FIX, E. Discriminatory analysis: Nonparametric discrimination, consistency properties. *Project 21-49-004*, USAF School of Aviation Medicine, n. 4, 1951.
- HAUGELAND, J. Artificial intelligence: The very idea. *Massachusetts: The MIT Press*, MIT Press, 1985.
- LORENA A. C.; CARVALHO, A. C. P. L. F. Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada*, v. 14, n. 2, p. 43–67, 2007.
- MICHELL, T. Machine learning. McGraw-Hill, 1997.
- MÜLLER, S. M. e. B. S. K. R. In: *An introduction to kernel-based learning algorithms*. [S.l.]: IEEE Transactions on Neural Networks, 2001. p. 181–201.
- SAMUEL, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, IBM, v. 3, n. 3, p. 210–229, 1959.
- SCHOLKOPF, A. S. e B. Learning with kernels. The MIT Press, 2002.
- SHUURMANS, A. S. B. Advances in large margin classifiers. *Introduction to large margin classifiers*, MIT Press, n. 2, p. 1–28, 1999.
- SIMON, H. A. Search and reasoning in problem solving. *Artificial Intelligence*, North-Holland, v. 21, n. 2, p. 7–29, 1983.
- VAPNIK, V. N. The nature of statistical learning theory. 1995.
- WANG, W. X. S. A new method for crude oil price forecasting based on support vector machines. *Computational Science ICCS*, Springer, 2007.
- WEHMEIER, S. In: *Oxford Advanced Learner's Dictionary*. [S.l.]: Oxford University Press, 2000.
- WEISS S.M.; KULIKOWSKI, C. Computer systems that learn. Morgan Kaufmann Publishers Inc., 1991.
- WILSON, T. D. Information overload: implications for healthcare services. *Health Informatics Journal*, v. 7, n. 7, p. 112, 2001.
- WITTEN, I. Data mining: Practical machine learning tools and techniques with java implementations. Morgan Kaufmann Publishers Inc., v. 3, 2011.