Sumário

[**OBJETIVOS** 2](#_Toc19977000)

[**JUSTIFICATIVA** 3](#_Toc19977001)

[**CAPITULO 1** 4](#_Toc19977002)

[**1 ESTADO DA ARTE** 4](#_Toc19977003)

[**1.1** **O Motor de Indução Trifásico** 7](#_Toc19977004)

[1.1.3 Rotor 9](#_Toc19977005)

[1.1.4 Categorias 10](#_Toc19977006)

[1.1.4 Falhas no Motor de Indução Trifásico 11](#_Toc19977007)

[**CAPITULO 2** 15](#_Toc19977008)

[**2 CONCEITOS DA COMPUTAÇÃO** 15](#_Toc19977009)

[**2.1 Aprendizado de Máquina** 15](#_Toc19977010)

[2.5.1 Aprendizado supervisionado 17](#_Toc19977011)

[2.5.2 Aprendizagem não supervisionada 18](#_Toc19977012)

[2.5.3 Aprendizagem por reforço 18](#_Toc19977013)

[**2.2 Computação na Nuvem** 19](#_Toc19977014)

[**2.3 IBM Cloud®** 21](#_Toc19977015)

[**2.4 IBM Cloud Services** 22](#_Toc19977016)

[2.3.1 Cloud Stourage Object(DUVIDA COLOCAR IMAGEM) 24](#_Toc19977017)

[2.3.2 Watson Studio e Watson Machine Learn 25](#_Toc19977018)

[2.3.3 IBM SPSS Modeler 30](#_Toc19977019)

[**2.5 Medidas de Precisão** 46](#_Toc19977020)

[**CAPITULO 3** 49](#_Toc19977021)

[**3 MÁQUINAS DE VETORES DE SUPORTE** 49](#_Toc19977022)

[**3.1 Revisão Da Literatura** 49](#_Toc19977023)

[**3.2 MVS margens rígidas** 51](#_Toc19977024)

[**3.3 MVS margem suave** 55](#_Toc19977025)

[**3.4 Funções Kernel** 58](#_Toc19977026)

[**CAPITULO 4** 64](#_Toc19977027)

[**4 MATERIAIS e MONTAGEM** 64](#_Toc19977028)

[**4.1 System On Chip Esp 8266** 64](#_Toc19977029)

[5.1.1Comunicação SPI 66](#_Toc19977030)

[**4.2 Accelerometer Mems Adxl 345** 67](#_Toc19977031)

[4.2.1 Spi No Adxl345 74](#_Toc19977032)

[**4.3 Sct-013** 75](#_Toc19977033)

[**CAPITULO 5** 77](#_Toc19977034)

[**5 METODOLOGIA** 77](#_Toc19977035)

[**5.1 Sistema de aquisição** 77](#_Toc19977036)

[**5.2 Sistema de Implementação** 82](#_Toc19977037)

[5.2.1 Preparação de dados 83](#_Toc19977038)

[5.2.2 Execução do método 83](#_Toc19977039)

[**5.3 Modelos Preditivos Gerados** 85](#_Toc19977040)

[5.3.1 *Dataset* 1 – Condição Normal 86](#_Toc19977041)

[5.3.2 *Dataset* 2 – Condição de Desbalanceamento 88](#_Toc19977042)

[5.3.3 *Dataset* 3 - Condições de ligação com somente duas fases 90](#_Toc19977043)

[5.3.4 *Dataset* 4 - Condições de Desnível na base 91](#_Toc19977044)

[**5.4** **Discussão dos resultados obtidos no estudo do modelo físico do sistema de monitoramento de vibração e corrente aplicado em IBM® Cloud** 93](#_Toc19977045)

[**6 Conclusão e Trabalhos Futuros** 94](#_Toc19977046)

[6.1 Conclusões 94](#_Toc19977047)

[6.2 Trabalhos Futuros 95](#_Toc19977048)

[**7 REFERÊNCIAS** 96](#_Toc19977049)

[**8 APÊNDICES** 99](#_Toc19977050)

# **OBJETIVOS**

**Objetivos Gerais**

Este trabalho tem como objetivo adquirir dados de condições de falhas externas e condições normais de um motor de indução em laboratório utilizando um sistema de aquisição embarcado controlado por um ESP8266 para obter dados das seguintes condições de funcionamento: normal, desbalanceado, duas fases e desnível na base. Para que através dos dados obtidos possa ser implementado os algoritmos de Máquinas de Vetores de Suporte (MVS, do inglês *Support Vector Machine*) na infraestrutura da IBM Cloud, objetivando realizar predições assertivas sobre cada condição simulada.

**Objetivos Específicos**

* Desenvolvimento de um sistema de baixo custo capaz de monitorar vibração e correntes para máquinas rotativas, objetivando identificar diferentes condições de funcionamento, através do auxílio da implementação de MVS.
* Obter dados para alimentar análises preditivas do condicionamento de funcionamento do motor de indução trifásica.
* Utilizar a infraestrutura disponível gratuitamente de análises estatísticas da Cloud para aplicação de MVS e gerar os modelos preditivos adequados às situações propostas.
* Investigar a aplicabilidade e capacidade dos métodos MVS para modelagem de previsão. Para verificar a aplicação dessa abordagem, os dados de referência são utilizados neste estudo
* Avaliar a variação dos parâmetros dos kernels rbf e sigmoid selecionado para as aplicações e fazer uma discussão do seu impacto na acurácia dos resultados.
* Contribuir com o desenvolvimento, estudo e aplicação de sistemas de investigação de caraterísticas e mineração de dados.

# **JUSTIFICATIVA**

O desenvolvimento de sistemas embarcados vem beneficiando o controle de manutenção na atualidade, de modo a aumentar a disponibilidade de operação das máquinas e controlar melhor as intervenções de manutenção. Porém há de se notar que bons hardwares de sistemas disponíveis no mercado possuem um elevado custo de implementação, não sendo viáveis para uma difusão em larga escala. Este estudo propõe uma alternativa de baixo custo para análise de falhas externas em motor de indução trifásico usando um sistema de captação de dados composto por um Systen on Chip – ESP8266, sensor acelerômetro MEMS e sensor de corrente SCT-013, para gerar o banco de dados necessários para as implementações em IBM Cloud® para utilizar a ferramenta de mineração de dados IBM SPSS Modeler, para aplicar os conhecimentos de MVS.

# **CAPITULO 1**

# **1 ESTADO DA ARTE**

O motor de indução é um dos principais componentes utilizados na indústria. Isso se dá pelo fato da sua alta performance, baixo custo e confiabilidade para gerar energia mecânica a partir da energia elétrica. Tais motores são utilizados em várias áreas da indústria devido sua flexibilidade de aplicação, é encontrado desde aplicações domésticas até aplicações de alta potência e prioridade na indústria. Entretanto, apesar de sua alta confiabilidade, o motor de indução pode ser exposto á diferentes condições de falhas e tais falhas podem levar o motor a um colapso e consequentemente à uma parada não planejada na produção que pode ser bastante disruptivo para o processo devido à escala industrial de produtividade.

A partir disso, é importante que a manutenção esteja continuamente monitorando o motor para que falhas sejam detectadas ainda em seus estados iniciais evitando um colapso. Essa detecção de falhas com antecedência permite que as ações corretivas sejam executadas o quanto antes. Dentre falhas externas as principais são: sobrecarga, falha em uma única fase, tensão de alimentação desbalanceada, rotor travado, reversão de fase e desnível do motor.

A primeira solução encontrada por engenheiros de manutenção para o monitoramento de motores de indução foi a utilização de relés eletromecânicos que protegiam o motor contra falhas (Elmore, 2004). Entretanto, essa solução se mostrou não eficiente devido a características do relé como lentidão na operação, significativo consumo de energia e necessitam de manutenção periódica devido às partes mecânicas envolvidas.

Com o surgimento da tecnologia dos semicondutores a manutenção preditiva obteve uma melhoria na capacidade de proteger um motor de indução. A partir desse momento, os relés eletromecânicos foram substituídos por relés de estado sólido que eram mais eficientes nos quesitos citados acima e ainda menor custo de fabricação e maior confiabilidade. O desenvolvimento da tecnologia de microprocessadores no final dos anos 1970 permitiu que fossem utilizados junto aos relés para proteção de motores de indução (IEEE, 1997) e a partir desse momento a lógica para a proteção passou a ser implementada através de softwares.

Os recentes desenvolvimentos nos softwares baseados em sistemas inteligentes fizeram com que engenheiros passassem a utiliza-los no diagnóstico de falhas de componentes de sistemas elétricos de potência como o motor de indução. (Kezunovic, 1997). Os sistemas baseados em técnicas de inteligência artificial podem substituir um humano, ao prover as informações necessárias sobre a performance de determinado sistema. Algumas dessas técnicas são Redes Neurais Artificiais, Lógica Fuzzy e Máquina de Vetores de Suporte.

As Redes Neurais Artificiais oferecem características sobre o processo interno de predição e de detecção de falhas, e ainda, não possuem algoritmos para treino que maximizam o processo de generalização de forma sistemática o que pode levar à sobreajuste do modelo em relação aos dados em estudo.

Por outro lado, é possível utilizar a Lógica Fuzzy para a detecção antecipada de condições de falha e conseguir interpretar os resultados com uma base teórica. Entretanto a Lógica Fuzzy requer um professional habituado às definições de suas normas e ao processamento dos dados de entrada.

A aplicação da metodologia de Máquinas de Vetores de Suporte-MVS vem sendo mais utilizadas para compor sistemas de detecção de falha devido à sua boa capacidade de generalização e sua alta taxa de acerto.

Existem estudos propostos que obtiveram bons resultados com o objetivo de fazer o diagnóstico de um motor de indução. Podem ser citados trabalhos como o de Kolla e Altmann, onde uma Rede Neural Artificial foi treinada com sinais de corrente e de voltagem para detectar falhas em motores de indução onde somente falhas elétricas foram analisadas.

Baccarini utilizou Maquinas de Vetores de Suporte treinadas com sinais de vibração afim de detectar falhas mecânicas. Desta vez somente falhas mecânicas foram analisadas.

Widodo e Yang, através de uma pesquisa apresentada sobre a detecção de falhas em máquinas utilizando Máquinas de Vetores de Suporte, concluíram que MVS é a técnica mais promissora para diagnóstico de falha e que necessita de trabalhos amplos para a aplicação da ferramenta para o monitoramento da condição e chegar ao diagnóstico de falhas em motores de indução. Brun e Ernst, também encontraram bons resultados em algumas aplicações de MVS.

A maior eficiência na fase de treino obtida com o modelo MVS em relação ao modelo de RNN foi encontrada por (E. Avci & D. Avci,2009). A fase de treino consiste na correlação entre medidas obtidas na aquisição de dados e as correspondentes falhas, e em termos práticos, dependendo da variação da severidade de falha, se não provido de suficientes padrões de falha o modelo poderá não correlacionar de maneira certa um dado de aquisição à uma falha especifica gerando um falso diagnostico. MVS é regido por teorias de aprendizado estatístico onde o modelo tenta desenvolver um sistema de classificação confiável que seja capaz de treinar em menos tempo.

Este trabalho propõe uma aplicação prática de MVS à identificação de falhas mecânicas e elétricas externas à um motor de indução trifásico implementada na infraestrutura de algoritmos na nuvem. A metodologia proposta busca captar as falhas através de um acelerômetro para medir vibrações e um sensor de corrente não invasivo para captar a corrente no estator. MVS será implementado na IBM Cloud através da utilização da ferramenta de mineração de dados SPSS Modeler, que executará a parte de processamento dos sinais obtidos.

## **O Motor de Indução Trifásico**

INTRODUÇÃO..

O fluxo magnético girante aparece no estator devido as correntes alternadas circulantes nas bobinas do estator. Este fluxo magnético do estator se desloca em relação ao rotor, cortando as barras do rotor induzindo tensões (Lei de Faraday e Lenz) que farão circular correntes também alternadas no rotor. Como as correntes do rotor têm polaridades contrárias do estator ,por definição, cria-se também no rotor um campo magnético girante que será atraído e arrastado pelo campo girante do estator. [99]

Desenvolve-se assim um conjugado mecânico no rotor levando o mesmo a girar. A velocidade do rotor () é sempre menor que a velocidade do campo girante do estator (), também chamada velocidade síncrona. Se o rotor fosse levado até a velocidade síncrona ( ), não haveria mais velocidade relativa entre os campos girantes do estator e do rotor e consequentemente a tensão induzida cessaria, não haveria mais corrente no rotor, o conjugado mecânico diminuiria e o rotor automaticamente perderia velocidade (<), então, novamente o rotor iria adquirir o conjugado. [99]et al, concluir construção e referencias.

A operação do motor girando sem carga denomina-se operação em vazio. À medida que se coloca carga no eixo a tendência da velocidade é diminuir para compensar o conjugado resistente da carga. A operação do motor com carga é denominada operação em regime permanente.[99]

A diferença entre a velocidade síncrona e a velocidade do rotor é chamada de velocidade de escorregamento ():

(1.1)

onde velocidade síncrona e velocidade do rotor

Assim, o escorregamento *s* é definido por:

, sendo (1.2)

O valor de s para motores de indução de gaiola é de 2 a 5% de acordo com a literatura[x]. A velocidade síncrona () é dado por:

(1.3)

onde:

= frequência em hertz

= número de pólos.

Podemos variar a velocidade e, consequentemente, variando-se o número de pólos (alterando-se construtivamente as bobinas do estator).

1.1.2 Características Construtivas e Princípios de Funcionamento

Com base em [4], a Figura 1 mostra a estrutura de motor de indução, que compreende:

As máquinas elétricas rotativas são constituídas de duas partes:

a) o estator que é a parte fixa;

b) o rotor que é a parte móvel.

É construído com chapas de material magnético e recebe o enrolamento de campo, cujas espiras são colocadas em ranhuras, como mostra a Figura 2.

Aí se situa o enrolamento de campo, que pode ser mono ou trifásico. A maneira como esse enrolamento é construído determina o número de pólos do motor, entre outras características operacionais. Suas pontas (terminais) são estendidas até uma caixa de terminais, onde pode ser feita a conexão com a rede elétrica de alimentação. [99]



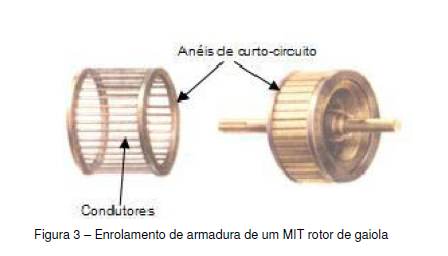
Figura 2 – Enrolamento de campo de um motor de indução: (a) execução dos enrolamentos; (b) núcleo

### 1.1.3 Rotor

O rotor do motor de indução pode ser de 2 tipos:

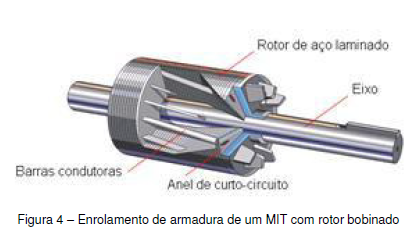
* Rotor em gaiola de esquilo ou rotor em curto: Os condutores são

constituídos por barras de cobre ou alumínio colocadas entre as ranhuras.



Nas duas extremidades das barras existem 2 anéis curto-circuitando todas as barras. Esta estrutura é semelhante a uma gaiola de esquilo (inglês, "*Squirrel Cage*"). Conforme já dito anteriormente é o tipo de rotor mais empregado (mais barato e não requer manutenção elétrica).

* Rotor bobinado ou rotor de anéis: A construção de um rotor bobinado é muito mais cara que um rotor em gaiola, e é executada quando se deseja a variação da velocidade da máquina. Um motor de rotor bobinado é possível controlar-se através da corrente que circula no rotor além da velocidade conjugado do mesmo.

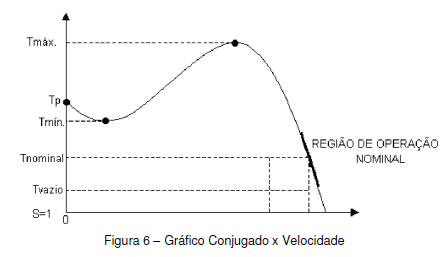


### 1.1.4 Categorias

O motor de indução tem um conjugado nulo à velocidade síncrona (n=ns), s=0 e T=0.

A medida que é aumentada a carga no eixo do motor, a sua velocidade diminui até um ponto onde o conjugado desenvolvido é máximo.

Qualquer acréscimo de carga além desse ponto (Tmáx ® Região de operação instável) faz com a velocidade caia bruscamente, podendo algumas situações travar o rotor.



Sendo:

Tp = conjugado de partida: é o conjugado com o motor travado, ou torque desenvolvido na partida do motor.

Tmín = conjugado mínimo: é o menor valor de conjugado obtido desde velocidade zero até a velocidade correspondente ao conjugado máximo.

Tmáx = conjugado máximo: é o máximo valor de conjugado que o rotor pode desenvolver *sem travar o eixo*.

Tnominal = conjugado nominal: é o conjugado que o motor fornece com carga nominal no eixo. Normalmente Tnominal ocorre com S entre 2 e 5%.

Tvazio = conjugado para o motor operando sem carga. Representa o conjugado sem carga no eixo.

### 1.1.4 Falhas no Motor de Indução Trifásico

As condições diversas as quais está sujeito o motor de indução, o conduz a diferentes maneiras de falhas que se propagam no motor podem ser internas ou externas. Neste trabalho serão abordadas falhas externas, que podem se classificar da seguinte forma:

1. Sobrecarga

2. Fase única

3. Tensão de alimentação desequilibrada

4. Rotor bloqueado

5. Reversão de fase

6. Falta de aterramento

7. Sobretensão

Uma breve descrição dessas falhas e suas características é fornecida abaixo. A proteção desses motores é uma tarefa importante e desafiadora para o âmbito da manutenção. A utilização de relés de proteção foi amplamente difundida, objetivando monitorar essas falhas e desconectar o motor em caso de falha. Porém o avanço das aplicações de monitoramento e diagnóstico de falha vem se modificando de acordo com a evolução das aplicações.

**Sobrecarga:**

A falha de sobrecarga ocorre quando o torque mecânico excede o ponto limite em aplicação de carga mecânica ao motor acima da classificação de carga total. Sobrecarga causa aumento em correntes de fase, superaquecendo a máquina. Em um sistema tradicional de proteção de relé, a sobrecorrente desarma o motor quando encontram sobrecarga de corrente na linha.

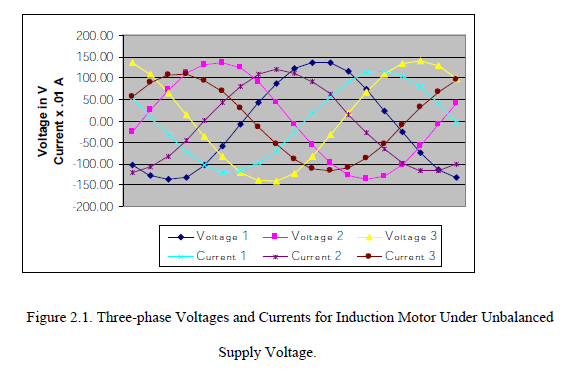
**Fase única:**

Ocorre quando duas das três linhas estão em aberto, desta forma mais corrente flui através das outras duas linhas e mais calor é gerado no enrolamento do estator. Nos sistemas de proteção tradicionais, um relé de unidade de disparo instantâneo de alto nível é usado (Elmore, 2004). A fase única também gera corrente de sequência negativa. Um negativo o relé de seqüência também pode ser usado para proteger contra essa falha.

**Tensão de alimentação desequilibrada:**

Existem muitas causas de tensões de alimentação desbalanceadas, como carga desbalanceada, transformadores a e configuração desigual de distribuição das fases. Esta condição leva à redução da eficiência do motor, aumenta a temperatura do motor e a corrente de carga total desequilibrada excessiva (Sudha & Anbalagan, 2009). Tensões e correntes trifásicas durante uma alimentação desequilibrada são mostradas na Figura 2.1.

O projeto de proteção deve detectar a condição de sobrecorrente durante a alimentação desbalanceada (Elmore, 2004).



**Rotor bloqueado**

O rotor bloqueado ocorre quando a tensão é aplicada a um motor não rotativo. A corrente pode ser quase seis vezes o seu valor nominal durante essa condição (Sudha & Anbalagan, 2009). Existem muitas causas para que essa falha ocorra, por exemplo, se o eixo do rotor estiver conectado a carregar o motor pode experimentar condições bloqueadas do rotor. Devido a essa elevação de corrente leva ao superaquecimento do rotor. Portanto, a condição de rotor bloqueado não pode ser suportada por muito tempo.

**Reversão de fase:**

Ocorre quando qualquer uma das duas fases é revertida da sequência normal, o que leva o motor a girar na direção oposta. Quando o motor começa a girar na direção oposta, pode causar danos intensos, caso essa transição seja feita de forma abrupta. Portanto, essa condição deve ser corrigida imediatamente. Relés de fase reversa e de sequência negativa são usados para a proteção (Elmore, 2004).

**Falta de aterramento: (melhorar estrutura)**

As falhas por ausência de aterramento ocorrem quando qualquer uma das fases toca o solo. São mais frequentes em motores do que qualquer outro sistema de energia, devido à sua condição violenta e frequente começa. Os efeitos dessa falha são intensos, como causar riscos à segurança humana e interferência nas telecomunicações. Pode ser detectado medindo a corrente de fuga ao solo (Elmore, 2004).

**Sobre tensão**

A falha de sobretensão está relacionada com aumentos percentuais no valor nomial da tensão de entrada para o funcionamento do sistema, o que faz com que o motor superaqueça, e perca características como velocidade do rotor. Relés de proteção de baixa tensão são usados em sistemas tradicionais de proteção contra esse tipo de falha (Elmore, 2004).

# **CAPITULO 2**

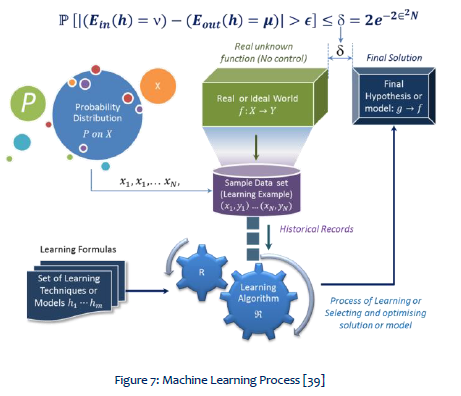
# **2 CONCEITOS DA COMPUTAÇÃO**

## **2.1 Aprendizado de Máquina**

A essência do Aprendizado de Máquina(AM) é um processo automático de reconhecimento de padrões. O principal objetivo do aprendizado de máquina é criar sistemas que possam executar ou exceder a competência no nível humano ao lidar com muitas tarefas ou problemas complexos. O aprendizado de máquina faz parte da Inteligência Artificial (IA). Durante o início da era de pesquisa da IA, o objetivo era construir robôs e simular atividades humanas. Posteriormente, a aplicação da IA foi generalizada para resolver problemas gerais por uma máquina. A solução popular era alimentar um computador com algoritmos (ou uma sequência de instruções) para transformar os dados de entrada em respostas. [ref]

No entanto, para muitos problemas, não podemos encontrar facilmente algoritmos adequados, por exemplo, o reconhecimento da caligrafia humana. Não sabemos como transformar a entrada da letra de escrita manual na saída da letra reconhecida padrão. Uma alternativa é aprender com os dados. Isso significa que, com uma tentativa, há um grande erro, mas se pudermos agregar muitas tentativas, o erro será reduzido para um nível ou convergência aceitável. A Figura 7 ilustra um exemplo típico de processo de aprendizado de máquina ou aprendizado de dados.

Desde o final dos anos 90 o volume de dados se tornou cada vez maior. Uma questão lógica é como lidar com esses grandes volumes de dados e como encontrar padrões úteis ou significativos a partir de um volume maior de dados. Isso leva à “descoberta de conhecimento no banco de dados” (ou KDD- knowledge discovery in database), que também é chamada de mineração de dados. Em outras palavras, queremos cavar no banco de dados e descobrir o significado ou conhecimento para a tomada de decisão. Larose et al. [47] definiram o termo como "o processo de descobrir padrões e tendências úteis em grandes conjuntos de dados". Para descobrir padrões significativos de um conjunto massivo de dados, a estatística é a ferramenta vital para agregar valor à amostragem, modelagem, análise, interpretação e apresentação de dados, assim como Jiawei Han et al. [48] indicaram: “A mineração de dados tem uma conexão inerente às estatísticas”. Isso leva à convergência do sistema de mineração de dados e do sistema especialista difuso sob o grande guarda-chuva do aprendizado de máquina. Do ponto de vista da evolução do aprendizado de máquina, a teoria estatística ou a modelagem de probabilidade mudaram a disciplina de IA de sistemas especialistas baseados em regras ou aprendizado de esquema na gravação, para uma metodologia de esquema na leitura ou orientada a dados, que é resolver o problema de incerteza com probabilidade dos parâmetros de um modelo. Nessa perspectiva, as estatísticas foram incorporadas ao aprendizado de máquina.



Desde a década de 1950, tem havido muitas definições funcionais de AM. Diferentes autores enfatizariam diferentes aspectos do aprendizado de máquina, como processo, aplicativo e utilidade. Por exemplo, a definição de Arthur Samuel enfatizou o "aprendizado automático" de AM. Tom M. Mitchell descreveu todos os componentes do processo de AM [50]. Kevin P. Murphy [51] e Christopher M. Bishop [52], por outro lado, enfatizaram a função do reconhecimento de padrões. Noam Nisan e Shimon Schocken [53] argumentaram que AM poderia transformar pensamentos abstratos em operação física. No resumo de mais de 30 definições, podemos encontrar alguns dos ingredientes essenciais e comuns dessas definições de AM:

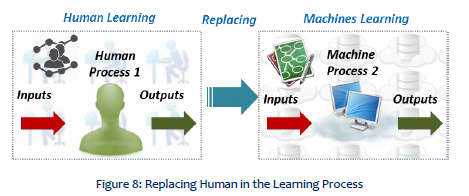
• Treinar a máquina para aprender automaticamente e melhorar os resultados à medida que obtém mais dados.

• Descobrir ou reconhecer padrões com dados de entrada.

• Executar predições através de entradas desconhecidas.

• A máquina irá adquirir conhecimento diretamente dos dados e resolverá problemas propostos.

De acordo com esses elementos, podemos descobrir que, fundamentalmente, o AM é "uma consequência da interseção entre ciência da computação e estatística, visa aprender automaticamente a reconhecer padrões complexos e a tomar decisões inteligentes com base em conjuntos de dados existentes" [54]. O objetivo final do AM é construir sistemas com nível de competência humana (veja a Figura 8) na execução de tarefas complexas.



O aprendizado de máquina permite que os modelos treinem em conjuntos de dados antes de serem implantados. Alguns modelos de aprendizado de máquina são online e contínuos. Esse processo iterativo de modelos online leva a uma melhoria nos tipos de associações feitas entre elementos de dados. Devido à sua complexidade e tamanho, esses padrões e associações poderiam ter sido facilmente ignorados pela observação humana. Depois que um modelo é treinado, ele pode ser usado em tempo real para aprender com os dados. As melhorias na precisão são resultado do processo de treinamento e automação que fazem parte do aprendizado de máquina [ibm knowledge].

O aprendizado de máquina oferece um valor potencial para as empresas que tentam alavancar o big data e as ajuda a entender melhor as mudanças no comportamento, preferências ou satisfação do cliente. Buscando padrões e anomalias ocultos nos dados que podem ajudar ou prejudicar os processos.

Técnicas de aprendizado de máquina são necessárias para melhorar a precisão dos modelos preditivos. Dependendo da natureza do problema de negócios que está sendo resolvido, existem diferentes abordagens baseadas no tipo e no volume dos dados. No tópico seguinte, será discutido as categorias de aprendizado de máquina.

### 2.5.1 Aprendizado supervisionado

O aprendizado supervisionado começa com um conjunto estabelecido de dados e um certo entendimento de como esses dados são classificados. O aprendizado supervisionado tem como objetivo encontrar padrões nos dados que podem ser aplicados a um processo de análise. Esses dados rotularam recursos que definem o significado dos dados. Por exemplo, é possível criar um aplicativo de aprendizado de máquina que distinga entre milhões de animais, com base em imagens e descrições escritas.

### 2.5.2 Aprendizagem não supervisionada

O aprendizado não supervisionado é usado quando o problema exige uma enorme quantidade de dados não rotulados. Por exemplo, aplicativos de mídia social, como Twitter, Instagram e Snapchat, todos têm grandes quantidades de dados não rotulados. Compreender o significado por trás desses dados requer algoritmos que classificam os dados com base nos padrões ou clusters encontrados. O aprendizado não supervisionado conduz um processo iterativo, analisando dados sem intervenção humana. É usado com a tecnologia de detecção de spam de e-mail. Existem muitas variáveis em emails legítimos e de spam para um analista marcar emails em massa não solicitados.

### 2.5.3 Aprendizagem por reforço

O aprendizado por reforço é um modelo de aprendizado comportamental. O algoritmo recebe feedback da análise de dados, guiando o usuário para o melhor resultado. O aprendizado por reforço difere de outros tipos de aprendizado supervisionado, porque o sistema não é treinado com o conjunto de dados de amostra. Em vez disso, o sistema aprende por tentativa e erro. Portanto, uma sequência de decisões bem-sucedidas resultará no reforço do processo, porque melhor resolve o problema em questão.

2.5.4 Aprendizagem profunda

O aprendizado profundo é um método específico de aprendizado de máquina que incorpora redes neurais em camadas sucessivas para aprender com os dados de maneira iterativa. O aprendizado profundo é especialmente útil quando se está tentando aprender padrões a partir de dados não estruturados. As redes neurais complexas de aprendizado profundo são projetadas para imitar como o cérebro humano funciona, para que os computadores possam ser treinados para lidar com abstrações e problemas mal definidos. A criança média de cinco anos de idade pode reconhecer facilmente a diferença entre o rosto do professor e o rosto. da guarda de cruzamento. Por outro lado, o computador deve trabalhar bastante para descobrir quem é quem. Redes neurais e aprendizado profundo são frequentemente usados em aplicativos de reconhecimento de imagem, fala e visão computacional.

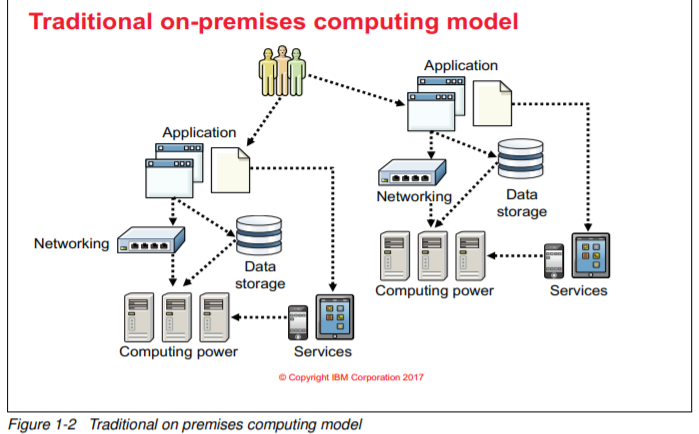
## **2.2 Computação na Nuvem**

O termo nuvem é usado como um termo genérico/metafórico frente a um conjunto virtualizado de hardware e recursos, tornando-se uma abstração para a infraestrutura complexa que que ela possui. Uma boa definição de computação em nuvem vem do Instituto Nacional de Padrões e Tecnologia (NIST). A definição do NIST essencialmente diz que:

“A computação em nuvem é um modelo para permitir acesso conveniente e sob demanda da rede a um conjunto compartilhado de recursos de computação configuráveis que podem ser rapidamente provisionados e lançado com esforço mínimo de gerenciamento ou interação com o provedor de serviços ".

Exemplos de recursos de computação incluem:

* Redes
* Servidores
* Armazenamento
* Aplicações
* Serviços



A computação na nuvem como modelo de implantação está substituindo uma abordagem mais antiga, na qual cada aplicativo com o qual um usuário interage teve seus próprios serviços personalizados, rede, dados armazenamento e poder de computação [ibm docs].

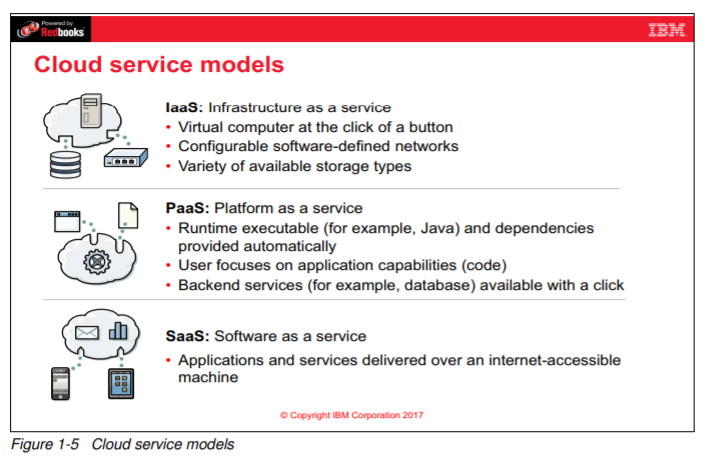
A capacidade de reutilizar e adaptar novamente o hardware rapidamente, além de hospedar vários aplicativos e sistemas em um único conjunto de hardware de maneira isolada, são algumas das principais características que impulsionam a adoção da computação em nuvem.

Na abordagem antiga, a equipe de TI necessitaria gerenciar todos os processos, desde o hardware, até alterações mais recentes do software. Esse modelo não escala tão bem quanto os negócios e as empresas atuais organizações exigem.

Uma contribuição importante para o crescimento da computação em nuvem acontece através da necessidade atual da velocidade da demanda de entrega de serviços, que devem ser entregues rapidamente. Os desenvolvedores são pressionados a colocar seu produto no mercado assim que possível. Necessitando de feedbacks rapidamente e, em seguida, repetem a ideia de tornar o produto melhor e mais rápido.

A nuvem torna os recursos de hardware prontamente disponíveis e rápidos de configurar, desta forma reduzindo o tempo necessário para que seja mostrada uma versão funcional de produtos. Além disso, permite a reutilização dos mesmos recursos para vários projetos sucessivos, o que é mais econômico. Um outro fator que contribui para o crescimento da computação em nuvem é que os desenvolvedores podem usar diferentes linguagens de programação.

Descrição dos Modelos de serviço:



- Infraestrutura como Serviço (IaaS), um conjunto de ativos físicos, como servidores, dispositivos de rede e discos de armazenamento, são oferecidos como dedicados e particulares acessível aos consumidores. Os serviços neste modelo suportam a infraestrutura de aplicativos.

- Plataforma como serviço (PaaS) é um modelo de serviço em nuvem no qual a estrutura de aplicativos e o tempo de execução é uma entidade virtualizada, de autoatendimento. O objetivo do PaaS é permitir o desenvolvedor ou equipe para se concentrar nas funções, código e dados de negócios do aplicativo, em vez de se preocupar com infraestrutura.

- Comumente utilizada por usuários da web o serviço de Software como Serviço (SaaS), embora possam não sabe disso. Os aplicativos no modelo SaaS são fornecidos sob demanda aos usuários através do Internet, em oposição aos aplicativos de desktop. Exemplos de aplicativos SaaS incluem Salesforce.com, Google Apps e Facebook.

## **2.3 IBM Cloud®**

A IBM Cloud**®** é uma plataforma de computação em nuvem aberta que combina a plataforma como serviço (PaaS) com infraestrutura como serviço (IaaS) e inclui um catálogo de diversos serviços, que podem ser usados para criar e implantar rapidamente aplicativos ou infraestruturas diversas.

Como PaaS, fornece aos desenvolvedores acesso ao software IBM**®** para integração, segurança e outras funções-chave. Os tipos de aplicativos podem variar de web, dispositivos móveis, big data e dispositivos inteligentes à Internet das coisas.[ref]

Como IaaS, ele permite que os desenvolvedores controlem detalhadamente a infraestrutura na qual seus aplicativos são implantados. Os desenvolvedores podem implantar servidores de alto desempenho, servidores virtuais e contêineres, utilizando a infraestrutura e sistemas de hardware nos locais dos data centers da IBM Cloud**®** em todo o mundo.[ref]

## **2.4 IBM Cloud Services**

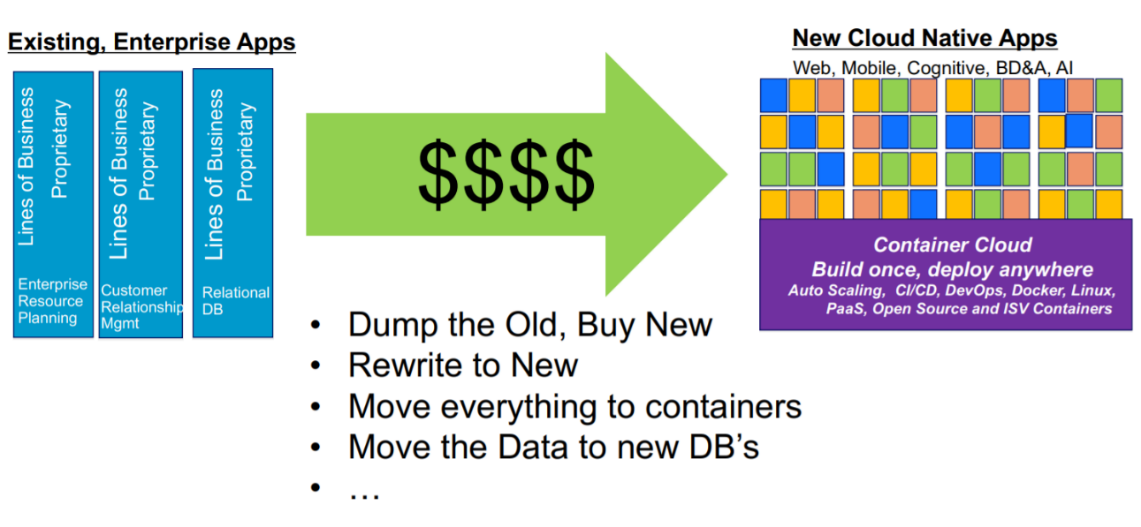
Através da imagem 87 é possível observar a gama de divisões dentre aplicações em IBM Cloud:



O IBM Cloud**®** fornece uma ampla gama de serviços pré-criados (da IBM e de terceiros) que pode ser usado frente a diferentes necessidades de tratamento estatístico.

Além disso, pode-se citar a criação, gerenciamento, e execução de APIS, uso da infraestrutura de *back-end* para criar e testar aplicações. Comunicando-se com dispositivos, sensores e gateways conectados caso sejam implementados. Várias opções de serviços pré-desenvolvidos são oferecidos, como Blockchain. Message Hub, WebSphere Application Server, Business Rules, Watson Studio entre outros serviços na nuvem totalizando mais de 190 serviços disponibilizados em IBM Cloud**®.**

Recomendações comuns para transformação atual das aplicações, na fig 60



A figura 60 ilustra bem o que vem acontecendo com todas as aplicações que buscam inovar e fazer parte dos avanços em quesitos de desenvolvimento junto a investimento na migração para a computação na nuvem, como já mostrado anteriormente.

As documentações disponibilizadas pela IBM Cloud ® fornecem informações base sobre suas aplicações, sendo de fundamental importância ter conhecimento prévio para a seleção correta das ferramentas a serem provisionadas.

Dentre a vasta biblioteca de desenvolvimento oferecida pela IBM Cloud, para o desenvolvimento dessa dissertação foi necessário provisionar os serviços de Watson Studio, Cloud Object Storage e Watson Machine Learn, descritos a seguir, com o objetivo de dar subsídios para a aplicação do processamento da utilização de MVS dentro do fluxo de IBM SPSS Modeler. Figura 91

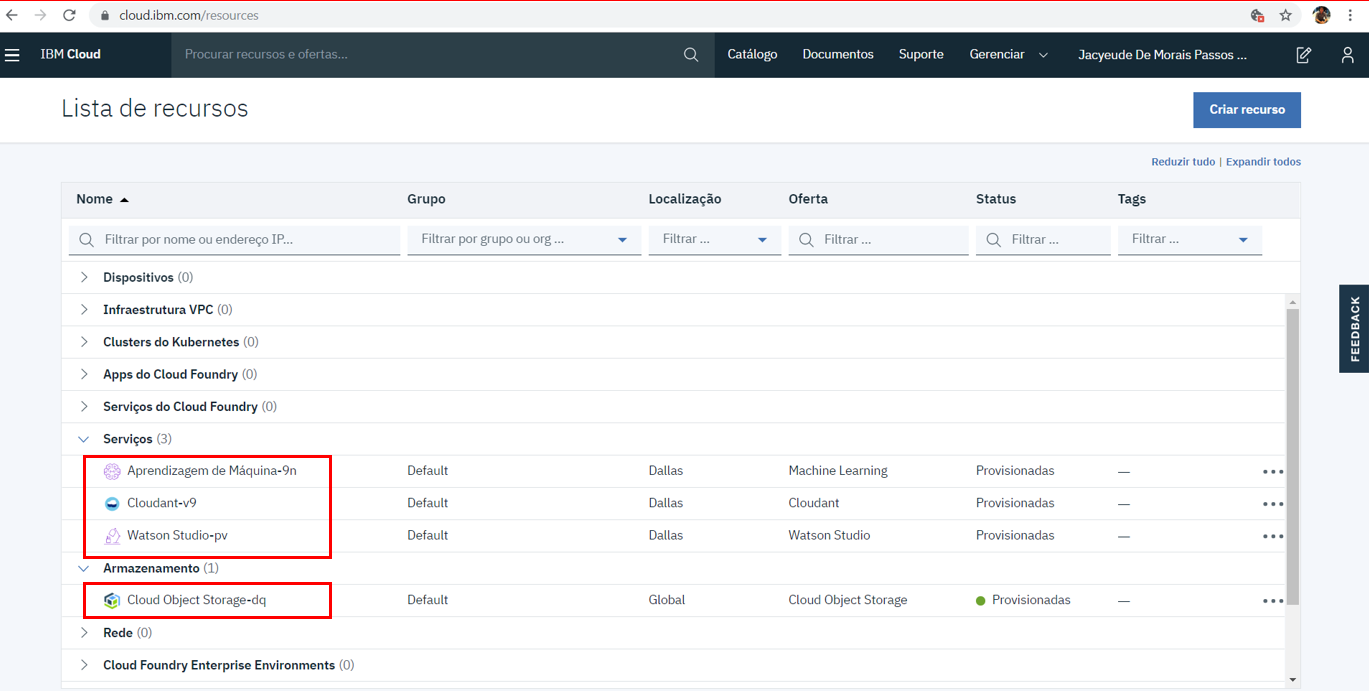


Figura 91

### 2.3.1 Cloud Stourage Object(DUVIDA COLOCAR IMAGEM)

{estrutura e conteúdo a verificar}

O IBM Cloud Object Store (CSO)é uma forma moderna de armazenamento de dados prática e eficiente. IBM COS é um mecanismo de armazenamento que aproveita um cluster de nós de armazenamento para armazenar os dados. O IBM COS usa um Algoritmo de Dispersão de Informações (IDA) para dividir os arquivos em partes que são distribuídas aos nós de armazenamento. [565]

O Cloud Stourage Object (CSO) foi criado para superar vários problemas, dentre os quais pode-se citar limitações estruturais e de segurança em vários níveis dos dados.

Os conceitos de armazenamento de dados do objeto incluem as três seguintes construções:

* Dados são compreendidos como: formato de texto, binários, planilhas, multimídia ou qualquer outro conteúdo gerado por humanos ou máquinas.
* Metadados: são os dados sobre os dados. Inclui alguns atributos predefinidos, como hora e tamanho do upload. O CSO permite que os usuários incluam metadados personalizados contendo informação em pares de chave e valor. Um aspecto exclusivo da manipulação de metadados nos sistemas de CSO é que os metadados são armazenados com o objeto.
* Um identificador universalmente exclusivo (UUID): esse ID é atribuído a todos os objetos em um sistema CSO. Esse UUID permite diferenciar objetos entre si para encontrar os dados sem precisar saber a unidade física exata, matriz de onde estão os dados.

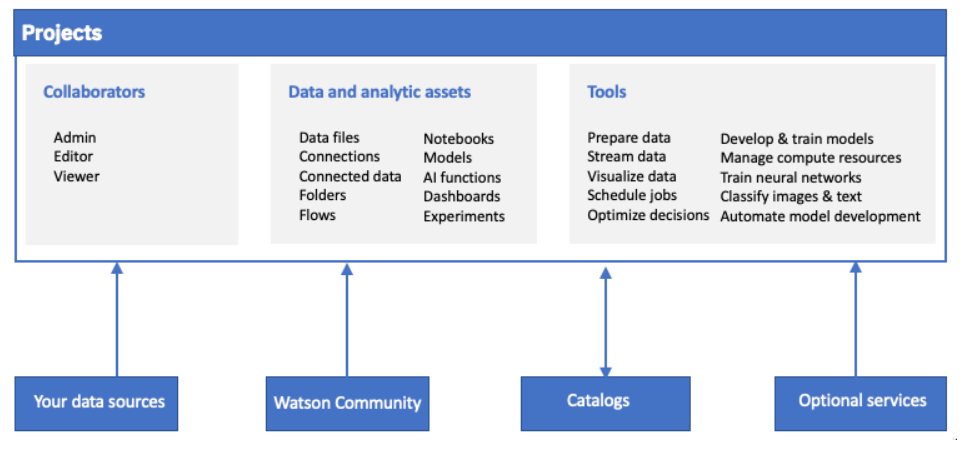
A integridade, escalabilidade, gerenciamento, disponibilidade e segurança são essenciais para os dados, porém as soluções de armazenamento tradicionais simplesmente não conseguem lidar com a escala na qual os dados estão crescendo atualmente. [565]

Sendo desta forma o Cloud Stourage Object oferecido pela IBM Cloud que irá armazenar os dados deste trabalho, tornando-os disponíveis para o fluxo das análises subsequentes dentro da infraestrutura da nuvem.

### 2.3.2 Watson Studio e Watson Machine Learn

O Watson Studio, fornece o ambiente e as ferramentas para análises estatísticas em conjunto de dados. Oferecendo funções de visualização, limpeza, experimentos que aplicam IA, modelagem dados, podendo ainda criar e treinar modelos de aprendizado de máquina [555].

Esta ilustração mostra como a arquitetura do Watson Studio é centralizada em torno de um projeto.



De acordo com [52] O Watson representa um avanço impressionante no design e análise de sistemas. Ele executa a tecnologia DeepQA(arquitetura probabilística paralela maciça baseada em evidências) da IBM, um novo tipo de capacidade analítica que pode executar milhares de tarefas simultâneas em segundos para fornecer respostas precisas a perguntas. Ativado pela tecnologia dos processadores IBM POWER7, o Watson é um exemplo das cargas de trabalho de análise complexa que estão se tornando cada vez mais comuns e essenciais para o sucesso e a competitividade dos negócios no ambiente atual de grande fluxo de dados.

O Watson se aproveita do desempenho de processamento paralelo maciço dos seus processadores POWER7 para executar suas milhares de tarefas da DeepQA simultaneamente em núcleos de processadores individuais. Cada um dos 90 servidores IBM Power7 em cluster do Watson conta com 32 núcleos POWER7, executados a 3.55 GHz. Executando o sistema operacional Linux®, os servidores são abrigados em 10 racks, juntamente com os nós e *hubs* de comunicação associados. O sistema tem um total combinado de 16 terabytes de memória e pode operar a mais de 80 teraflops (trilhões de operações por segundo).[52]

A capacidade do Watson para entender o significado e o contexto da linguagem humana e para processar rapidamente informações para encontrar respostas precisas para perguntas complexas guarda um potencial enorme para transformar a forma na qual os computadores podem ajudar as pessoas a realizar tarefas nos negócios e nas suas vidas pessoais.

Os colaboradores tem acesso as entradas de dados em Data Assets, podendo carrega-los automaticamente, tendo disposição posterior de uma importante ferramenta na aplicação, Data Refinery(anexo X), onde podem ser criados fluxos de tratamentos dos dados incialmente dispostos, podendo retirar espaços vazios, ou realizar visualizações gráficas dos dados de entrada.

Tendo opções de criação, treinamento e testes de modelos de aprendizado de máquina e aprendizado profundo (deep learn).

Sendo possível também a realização de experimentos com modelos de aprendizado profundo em paralelo com outras aplicações. O Watson Studio também pode trazer dados e ativos analíticos da Comunidade IBM Watson, para avaliar testes como exemplos.

É possível encontrar ativos nos catálogos e os adicionar a qualquer projeto. Ao concluir o desenvolvimento de modelos em um projeto, a documentação de IBM CLOUD indica que deve-se promover os modelos no espaço de implantação associado ao projeto. Tornando possível configurar modelos no espaço de implantação.

Algumas ferramentas exigem complementos, havendo necessidade de fazer provisionamento das ferramentas: IBM Cloud Object Stourage e IBM Watson Machine Learn, para dar os subsídios necessários para o upload de dados e análises estatísticas. A seleção de ferramentas complementares desta dissertação se deu através de considerações sobre tipos de dados, tipos de tarefas e quanto de automação pode ser implementado através de editores, dentre os quais cita-se: *Jupyter, RStudio IDE ou SPSS Modeler*.

Estas ferramentais diferenciam-se por serem melhor aplicadas a contextos de dados diferentes como : [dados tabulares ou relacionais](https://dataplatform.cloud.ibm.com/docs/content/wsj/getting-started/tools.html#tab), [dados textuais](https://dataplatform.cloud.ibm.com/docs/content/wsj/getting-started/tools.html#text), [dados de imagem](https://dataplatform.cloud.ibm.com/docs/content/wsj/getting-started/tools.html#image).

Uma tabela disposta pela documentação de IBM Cloud mostra as tarefas que são executadas pelas ferramentas, para análises de dados tabulares ou relacionais (dados do projeto).

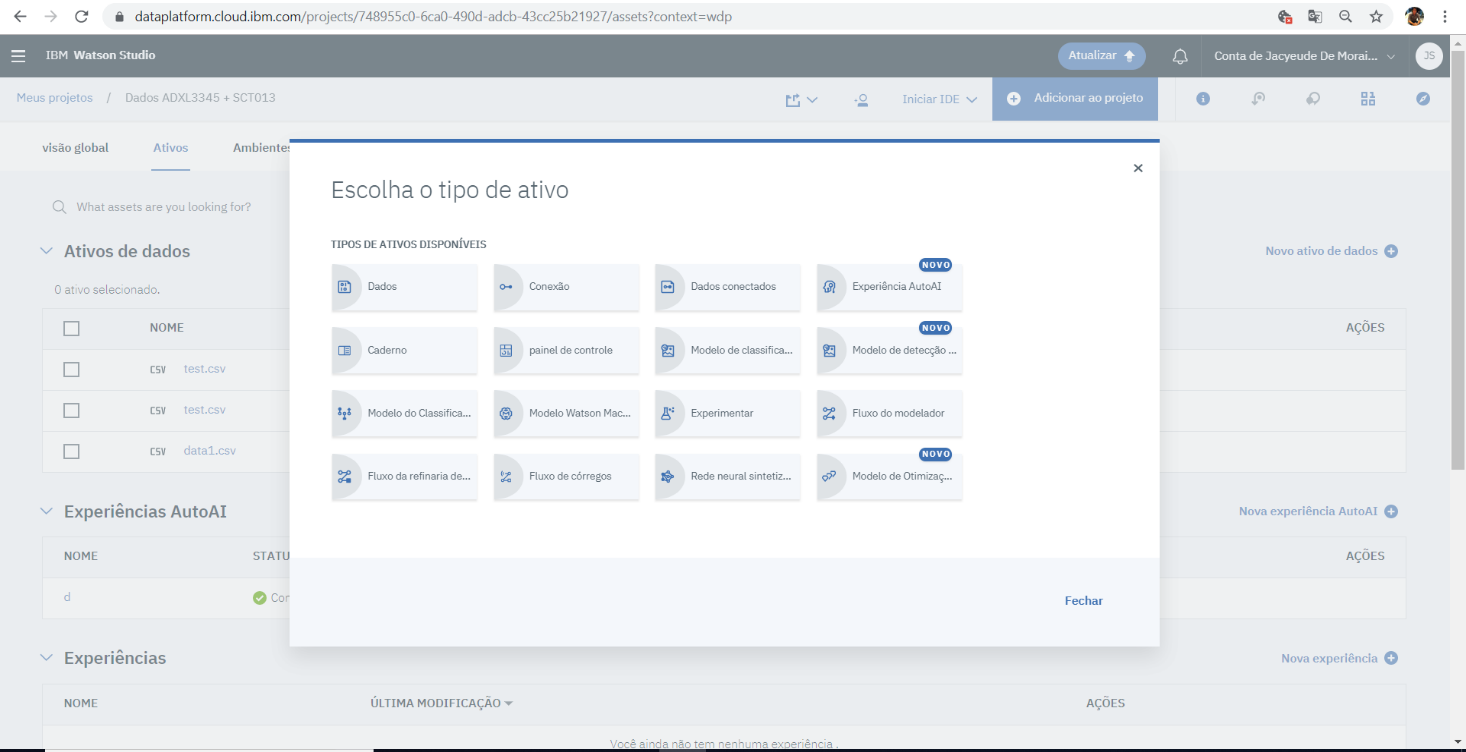


O ambiente de Watson Studio conta com ferramentas gráficas para projetar, treinar, implantar e gerenciar modelos com seus serviços do Watson Aprendizagem de Máquina:

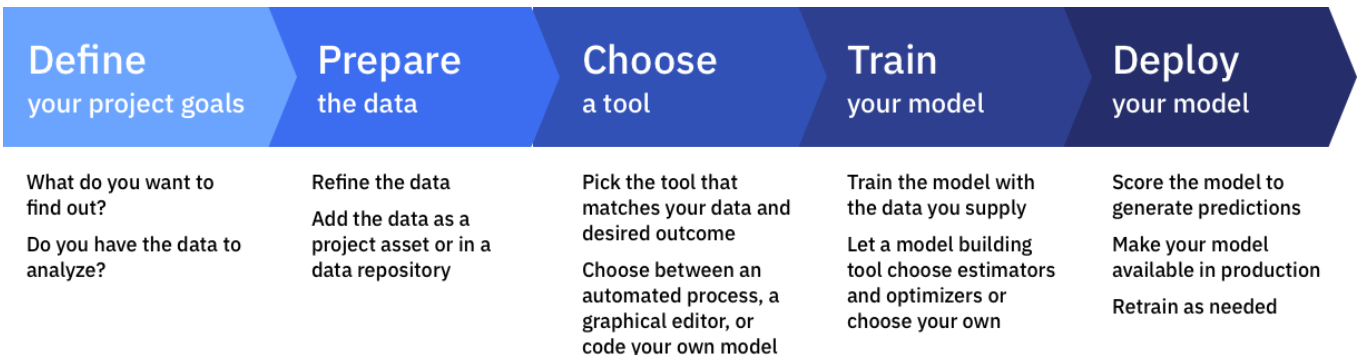
* As experiências da AutoAI processam automaticamente seus dados, selecionam o melhor estimador para os dados e, em seguida, geram pipelines candidatos a modelos para que possam ser revisados e comparados. Implante o pipeline com melhor desempenho como modelo de aprendizado de máquina.
* A Rede Neural Sintetizada sintetiza automaticamente uma rede neural e a treina nos seus dados de treinamento sem que seja necessário projetar ou construir qualquer coisa manualmente.
* O modelador Spark MLlib apresenta uma visualização gráfica do modelo enquanto é construído um combinando nós que representam nós de algoritmo.
* O modelador SPSS apresenta uma visualização gráfica do seu modelo enquanto é construído um combinando nós que representam objetos ou ações, realizando implementações de algoritmos para mineração de dados.
* O modelador de rede neural apresenta uma visualização gráfica do seu modelo enquanto é construído um combinando nós de rede neural).
* O criador de experiências automatiza a execução de centenas de execuções de treinamento enquanto rastreia e armazena resultados.
* O construtor de modelo *Decision Optimization* orienta na construção e resolução de modelos prescritivos.

Através de Watson Aprendizado de Máquina(AM), construído em uma plataforma escalável de código aberto baseada nos componentes Kubernetes e Docker, sendo possível construir modelos analíticos e redes neurais, treinadas com seus próprios dados, que podem ser implementadas para uso em aplicativos. Com isso a seleção do provisionamento de Watson AM é fundamental para este trabalho.

A imagem que mostra tipos de ativos que podem ser implementados em Watson Studio segue na figura 94

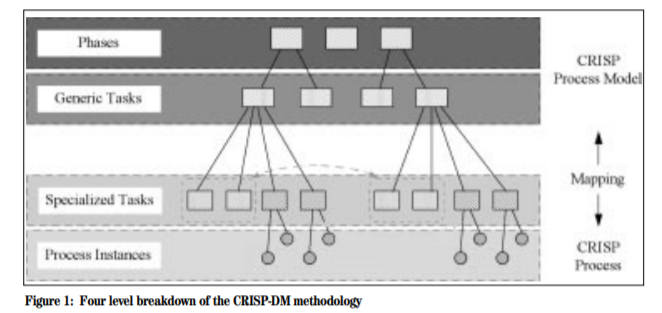


Abaixo segue o fluxo de que ilustra o desenvolvimento de uma aplicação a ser desenvolvida em Watson Studio:



### 2.3.3 IBM SPSS Modeler

O IBM® SPSS® Modeler é um conjunto de ferramentas de mineração de dados que permite desenvolver modelos preditivos usando conhecimentos estatísticos. Projetado com base no modelo CRISP-DM. Este é descrito em termos de um modelo de processo hierárquico, consistindo em conjuntos de tarefas descritas em quatro níveis de abstração, figura 0032: fase, tarefa genérica, tarefa especializada e instância de processo.[555]



É necessário que seja observado o contexto da mineração de dados em tópicos científicos, em que se tornou importante e útil na maioria dos campos com largas databases, sendo hábil para extrair conhecimento de uma massa de dados brutos armazenados [4].

A mineração de dados está encontrando os padrões anteriormente “ocultos” em diferentes dados na forma semiautomática [4] descrendo as informações através de métodos e modelos, como modelos analíticos e de classificação, além de apresentar resultados usando diferentes ferramentas disponíveis.

Seu conceito base fundamenta-se na capacidade de sistematizar processos, podendo obter quesitos relevantes a informações e conhecimento a partir de bancos de dados. É indispensável associar a mineração de dados a eficiência e vantagem competitiva, características indispensáveis para o mercado atual de aplicações. em diferentes contextos estatísticos, invadindo diferentes áreas para buscar informações inerentes a cada processo, necessitando de abstrações particulares para eventos diferentes, respeitando um fluxo de ações a ser tomado. (referência)

É necessário diferenciar conceitos de dados, informações e conhecimento dentro dos contextos de desenvolvimento. Um dado isolado não possui significado relevante e não conduz a nenhuma compreensão, representa algo que não tem sentido a princípio. Portanto, não tem valor algum para embasar conclusões, muito menos respaldar decisões. A informação é a ordenação e organização dos dados de forma a transmitir significado e compreensão dentro de um determinado contexto, seria o conjunto ou consolidação dos dados de forma a fundamentar para obtenção conhecimento. Enquanto conhecimento vai além de informações, pois com ele é possível extrair significados em termos de uma aplicação.

O SPSS Modeler oferece uma variedade de métodos de modelagem retirados do AM, inteligência artificial, além de outros métodos estatísticos.

Ele utiliza nós para ajudá-lo a explorar dados. Vários nós na área de trabalho de sua interface representam diferentes objetos e ações. A aba na parte lateral esquerda da janela do IBM SPSS Modeler contém todos os abas que contém os nós usados na criação de fluxos, como mostra a figura 73.

Existem vários tipos de nós, organizados em campos específicos da aba principal que serão descritos. Conectando os nós, geram-se fluxos que, quando executados, permitem visualizar relacionamentos entre os dados e aplicações. Os fluxos são como scripts, sendo possível salvá-los e reutilizá-los com diferentes arquivos de dados.

Um nó executável que processa dados de fluxo é conhecido como **nó terminal**. Um nó de modelagem ou saída é um nó terminal se estiver localizado no final de um fluxo ou ramificação de fluxo não pode conectar a outros nós a um nó do terminal.

Os métodos e nós disponíveis na aba principal de trabalho são mostrados na figura 73.

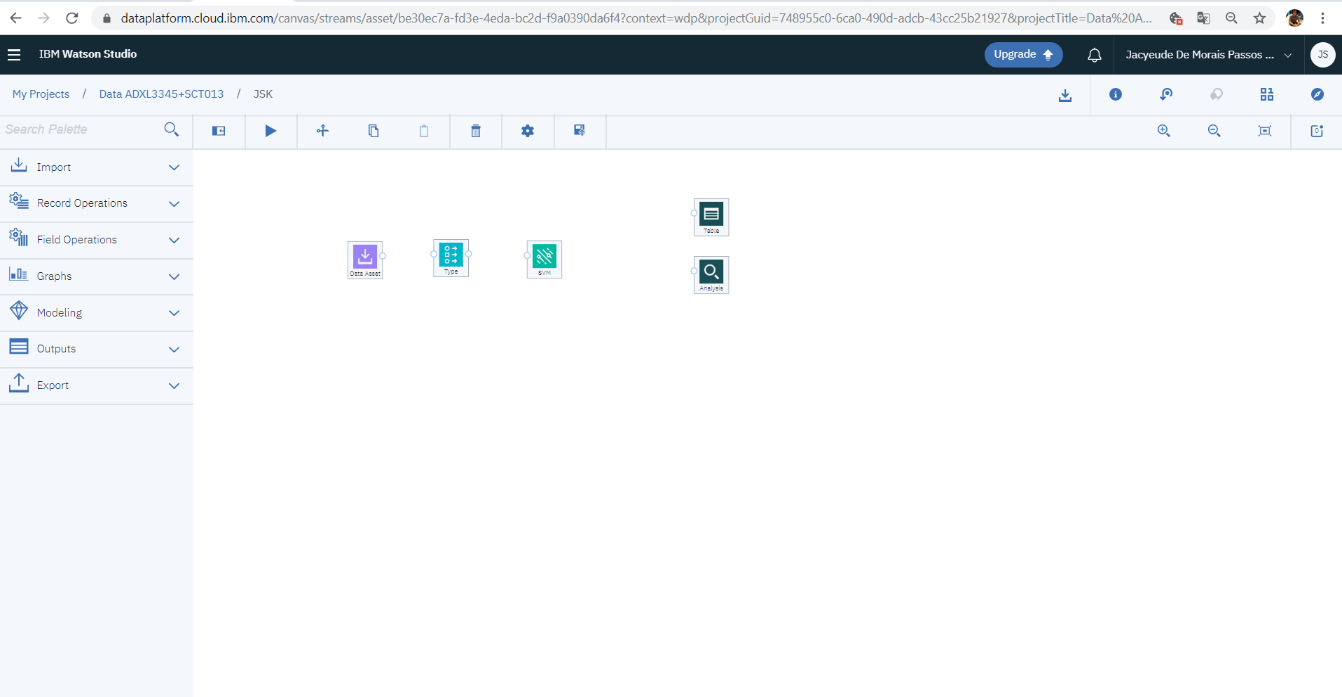


Figura 73

Com o IBM SPSS Modeler, é possível construir modelos de AM com facilidade de arrastar e soltar. Inicialmente carregando os dados, executando etapas do pré-processamento sendo possível transformá-los, antes de aplicar algoritmos e avaliar o desempenho do modelo preditivo criado. Objetivando encontrar padrões ou variáveis ocultas que influenciam os resultados.

A seguir serão mostradas sínteses dos conteúdos dos principais nós de IBM SPSS Modeler:

Através da aba Import é possível inserir o banco de dados a que se deseja analisar, ou até mesmo cria-lo para seguir o fluxo de análises.

Os nós de Record Operations são usados para fazer alterações nos dados, compreendendo a etapa de pré-processamento. Essas operações são importantes durante as fases de mineração de dados para compreensão e preparação de dados, pois permitem sejam adaptados às necessidades devidas, eles serão descritos na tabela 88.

A aba *Record Operations* contém os seguintes nós:

Tabela 88

|  |  |
| --- | --- |
| Nós | Descrição |
|  | O nó Selecionar seleciona ou descarta um subconjunto de registros do fluxo de dados com base em uma condição específica. Por exemplo, é possível selecionar os registros que pertencem a uma região específica da data. |
|  | O nó Amostra seleciona um subconjunto de registros. Uma variedade de tipos de amostra é suportada, incluindo amostras estratificadas, agrupadas e não aleatórias (estruturadas). A amostragem pode ser útil para melhorar o desempenho e selecionar grupos de registros ou transações relacionados para análise. |
|  | O nó Equilíbrio corrige desequilíbrios em um conjunto de dados, para que esteja em conformidade com uma condição especificada. A diretiva de balanceamento ajusta a proporção de registros em que uma condição é verdadeira pelo fator especificado. |
|  | O nó Agregado substitui uma sequência de registros de entrada por registros de saída agregados resumidos. |
|  | O nó Classificar classifica os registros em ordem crescente ou decrescente com base nos valores de um ou mais campos. |
|  | O nó Anexar concatena conjuntos de registros. É útil para combinar conjuntos de dados com estruturas semelhantes, mas com dados diferentes. |
|  | O nó Distinct remove registros duplicados, passando o primeiro registro distinto para o fluxo de dados ou descartando o primeiro registro e passando quaisquer duplicatas para o fluxo de dados. |

Os nós de *Field Operations* contêm ferramentas úteis para essa transformação e preparação de bancos de dados já importados e inicialmente tratados.

A aba de *Field Operations* contém os seguintes nós:

|  |  |
| --- | --- |
| Nó | Descrição |
|  | O nó ADP (Automated Data Preparation) pode analisar seus dados e identificar correções, filtrar campos pouco prováveis de serem úteis, obter novos atributos quando apropriado e melhorar o desempenho por meio de técnicas inteligentes de triagem e amostragem. É possível usar o nó de maneira totalmente automatizada, permitindo que ele escolha e aplique correções ou pode visualizar as alterações antes que elas sejam feitas e aceitá-las, rejeitá-las ou alterá-las conforme desejado. |
|  | O nó Tipo especifica os metadados e as propriedades do campo. Por exemplo, é possível especificar um nível de medição (contínuo, nominal, ordinal ou sinalizador) para cada campo, definir opções para lidar com valores ausentes e nulos do sistema, definir a função de um campo para fins de modelagem, especificar rótulos de campo e valor e especifique valores para um campo. Nó fundamental para identificar e classificar os dados categoricamente. |
|  | O nó Filtro filtra renomeia os campos e mapeia os campos de um nó de origem para outro. |
|  | O nó Derivar modifica os valores dos dados ou cria novos campos a partir de um ou mais campos existentes. Ele cria campos do tipo fórmula, sinalizador, nominal, estado, contagem e condicional. |
|  | O nó Ensemble combina dois ou mais *nuggets* de modelo para obter previsões mais precisas do que as obtidas em qualquer modelo. |
|  | O nó Preenchimento substitui os valores do campo e altera o armazenamento. Pode-se optar por substituir valores com base em uma condição, como @BLANK para @FIELD. Como alternativa, é possível optar por substituir todos os espaços em branco ou valores nulos por um valor específico. Um nó de preenchimento geralmente é usado junto com um nó de tipo para substituir os valores ausentes. |
|  | O nó Anonymize transforma a maneira como os nomes e os valores dos campos são representados a jusante, disfarçando os dados originais. Isso pode ser útil se for necessário permitir que outros usuários construam modelos usando dados confidenciais, como nomes de clientes ou outros detalhes. |
|  | The Reclassify node transforms one set of categorical values to another. Reclassification is useful for collapsing categories or regrouping data for analysis. See the topic [Reclassify Node](https://www.ibm.com/support/knowledgecenter/SS3RA7_15.0.0/com.ibm.spss.modeler.help/reclassify_overview.htm?view=kc) for more information. |
|  | O nó Binning cria automaticamente novos campos nominais definidos com base nos valores de campos contínuos existentes (intervalo numérico |
|  | O nó Partição gera um campo de partição, que divide os dados em subconjuntos separados para os estágios de treinamento, teste e validação da construção do modelo. |
|  | O nó Reestruturar converte um campo nominal em um grupo de campos que podem ser preenchidos com os valores de outro campo. |
|  | O nó Transpose troca os dados em linhas e colunas para que os registros se tornem campos e os campos se tornem registros. |
|  | O nó Intervalos de tempo especifica intervalos e cria rótulos (se necessário) para modelar dados de séries temporais. Se os valores não estiverem uniformemente espaçados, o nó poderá preencher ou agregar valores conforme necessário para gerar um intervalo uniforme entre os registros. |
|  | O nó Histórico cria novos campos contendo dados de campos em registros anteriores. Os nós do histórico costumam ser usados para dados sequenciais, como dados de séries temporais. |

Várias fases do processo de mineração de dados usam gráficos e tabelas para explorar os dados trazidos para o IBM® SPSS® Modeler. Desta forma na aba *Graphs* é possível conectar um nó de plotagem ou distribuição a uma fonte de dados para obter informações sobre os tipos de distribuições de dados com gráficos, podendo verificar a distribuição e os relacionamentos entre os campos recém-criados.

A aba *Graphs* contém os seguintes nós:

|  |  |
| --- | --- |
|  | O nó Graphboard oferece muitos tipos diferentes de gráficos em um único nó. Usando esse nó, é possível pode escolher os campos de dados que deseja explorar e, em seguida, selecionar um gráfico dentre os disponíveis para os dados selecionados. |
|  | O nó Plot mostra o relacionamento entre os campos numéricos. É possível criar um gráfico usando pontos (um gráfico de dispersão) ou linhas. |
|  | O nó Distribuição mostra a ocorrência de valores simbólicos (categóricos). Normalmente, pode-se usar o nó Distribuição para mostrar desequilíbrios nos dados, que podem ser corrigidos usando um nó Equilíbrio antes de criar um modelo. |
|  | O nó Histograma mostra a ocorrência de valores para campos numéricos. É frequentemente usado para explorar os dados antes de manipulações e construção de modelos. |
|  | O nó Coleção mostra a distribuição de valores para um campo numérico em relação aos valores de outro. Ele cria gráficos semelhantes aos histogramas. É útil para ilustrar uma variável ou campo cujos valores mudam ao longo do tempo. Usando gráficos 3D. |
|  | O nó Multiplot cria uma plotagem que exibe vários campos Y em um único campo X. Os campos Y são plotados como linhas coloridas; cada um é equivalente a um nó de plotagem, com Estilo definido como Linha e Modo X definido como Classificação. Multiplots são úteis quando se deseja explorar a flutuação de várias variáveis ao longo do tempo. |
|  | O nó da Web ilustra a força do relacionamento entre valores de dois ou mais campos simbólicos (categóricos). O gráfico usa linhas de várias larguras para indicar a força da conexão. |
|  | O nó Gráfico de tempo exibe um ou mais conjuntos de dados de séries temporais. Normalmente, usando primeiro um nó Intervalos de tempo para criar um campo TimeLabel, que seria usado para rotular o eixo x. |
|  | O nó Avaliação ajuda a avaliar e comparar modelos preditivos. O gráfico de avaliação mostra quão bem os modelos prevêem resultados específicos. Classifica os registros com base no valor previsto e na confiança da previsão. |

O SPSS Modeler Applications Guide fornece exemplos para muitos desses métodos, juntamente com uma revisão geral ao processo desenvolvimento de fluxos. Este guia está disponível como um tutorial on-line e também em formato PDF nas bibliotecas de knowledge center.[555]

Os métodos da aba de *Modeling* são divididos em três categorias de: classificação, associação e segmentação. Descritos abaixo.

* **Modelos de classificação**

Os modelos de classificação usam os valores de um ou mais campos de entrada para prever o valor de um ou mais campos de saída. Alguns exemplos dessas técnicas são: Árvores de Decisão (algoritmos C&R Tree, QUEST, CHAID e C5.0), Regressão (Algoritmos Lineares, Linear Generalizado e Regressão de Cox), Redes Neurais, Máquinas de Vetores de Suporte- MSV e Redes Bayesianas.

Os modelos de classificação ajudam desenvolvedores a prever um resultado conhecido, como se um cliente irá comprar ou sair ou se uma transação ou encaixar os resultados em um padrão conhecido. As técnicas de modelagem incluem aprendizado de máquina, indução de regras, identificação de subgrupos, métodos estatísticos e geração de modelos múltiplos.[55]

|  |  |
| --- | --- |
|  | O nó Classificador automático cria e compara vários modelos diferentes para resultados binários (sim ou não, rotatividade ou não, etc.), permitindo que seja possível a escolha da melhor abordagem para uma determinada análise. Vários algoritmos de modelagem são suportados, possibilitando selecionar os métodos que possam ser usados, as opções específicas para cada um e os critérios para comparar os resultados. O nó gera um conjunto de modelos com base nas opções especificadas e classifica os melhores candidatos de acordo com os critérios especificados. |
|  | O nó numérico automático estima e compara modelos para resultados contínuos de intervalo numérico usando vários métodos diferentes. Ele também permite escolher os algoritmos a serem usados e experimentar várias combinações de opções em uma única passagem de modelagem. Os algoritmos suportados incluem redes neurais, C&R Tree, CHAID, regressão linear, regressão linear generalizada e máquinas de vetores de suporte (MVS). Os modelos podem ser comparados com base na correlação, erro relativo ou número de variáveis usadas. |
| C&RT node | O nó da árvore Classificação e regressão (C&R) gera uma árvore de decisão que permite prever ou classificar observações futuras. O método usa o particionamento recursivo para dividir os registros de treinamento em segmentos, minimizando a impureza em cada etapa, em que um nó na árvore é considerado "puro" se 100% dos casos no nó caírem em uma categoria específica do campo de destino. Os campos de destino e entrada podem ser intervalos numéricos ou categóricos (nominal, ordinal ou sinalizadores); todas as divisões são binárias. |
| QUEST node | O nó QUEST fornece um método de classificação binário para a construção de árvores de decisão, projetado para reduzir o tempo de processamento necessário para grandes análises da Árvore C&R, além de reduzir a tendência encontrada nos métodos da árvore de classificação para favorecer entradas que permitem mais divisões. Os campos de entrada podem ser intervalos numéricos (contínuos), mas o campo de destino deve ser categórico. Todas as divisões são binárias. |
| CHAID node | O nó CHAID gera árvores de decisão usando estatísticas de qui-quadrado para identificar divisões ideais. Diferente da árvore C&R e dos nós QUEST, o CHAID pode gerar árvores não binárias, o que significa que algumas divisões têm mais de dois ramos. Os campos de destino e de entrada podem ser intervalo numérico (contínuo) ou categórico. CHAID exaustivo é uma modificação do CHAID que faz um trabalho mais minucioso ao examinar todas as divisões possíveis, mas leva mais tempo para calcular. |
| C5.0 | O nó C5.0 constrói uma árvore de decisão ou um conjunto de regras. O modelo funciona dividindo a amostra com base no campo que fornece o ganho máximo de informações em cada nível. O campo de destino deve ser categórico. São permitidas várias divisões em mais de dois subgrupos. |
| Decision List node | O nó Lista de Decisão identifica subgrupos ou segmentos que mostram uma probabilidade maior ou menor de um determinado resultado binário em relação à população geral. Os modelos de Lista de Decisão consistem em uma lista de regras nas quais cada regra tem uma condição e um resultado. |
|  | Os modelos de regressão linear preveem um alvo contínuo com base em relacionamentos lineares entre o alvo e um ou mais preditores. |
| PCA/Factor node | O nó PCA/Factor fornece técnicas de redução de dados para reduzir a complexidade de seus dados. A análise de componentes principais (PCA) encontra combinações lineares dos campos de entrada que realizam o melhor trabalho de captura da variação em todo o conjunto de campos, onde os componentes são ortogonais (perpendiculares) entre si. |
|  | O nó Seleção de recurso seleciona os campos de entrada para remoção com base em um conjunto de critérios (como a porcentagem de valores ausentes); em seguida, classifica a importância das entradas restantes em relação a um destino especificado. |
|  | A análise discriminante faz suposições mais rigorosas do que a regressão logística, mas pode ser uma alternativa valiosa ou complementar uma análise de regressão logística quando essas suposições são cumpridas. |
|  | A regressão logística é uma técnica estatística para classificar registros com base nos valores dos campos de entrada. É análogo à regressão linear, mas utiliza um campo de destino categórico em vez de um intervalo numérico. |
|  | O modelo Linear Generalizado expande o modelo linear geral para que a variável dependente seja linearmente relacionada aos fatores e covariáveis através de uma função de link especificada. Além disso, o modelo permite que a variável dependente tenha uma distribuição não normal. |
|  | Um modelo misto linear generalizado (GLMM) estende o modelo linear para que o alvo possa ter uma distribuição não normal, é linearmente relacionado aos fatores e covariáveis por meio de uma função de link especificada e para que as observações possam ser correlacionadas. Modelos mistos lineares generalizados cobrem uma ampla variedade de modelos, desde regressão linear simples a modelos multiníveis complexos para dados longitudinais não normais. |
|  | O nó de regressão de Cox permite criar um modelo de sobrevivência para dados de tempo até o evento na presença de registros censurados. O modelo produz uma função de sobrevivência que prediz a probabilidade de que o evento de interesse tenha ocorrido em um determinado momento (t) para determinados valores das variáveis de entrada. |
|  | O nó Support Vector Machine (MVS- Maquinas de vetores de suporte) é adequado para uso com amplos conjuntos de dados. Podendo usar as configurações padrão no nó para produzir um modelo básico de forma relativamente rápida, ou as configurações de Especialista para experimentar diferentes tipos de modelo MVS. permite que sejam classificados dados em um ou mais grupos sem sobreajuste. Sendo Amplamente utilizado por critérios de assertividade na melhoria de tomada de decisão frente à padrões encontrados nos bancos de dados |
|  | O nó Rede Bayesiana permite criar um modelo de probabilidade combinando evidências observadas e registradas com conhecimento do mundo real para estabelecer a probabilidade de ocorrências. O nó concentra-se nas redes Tree Augmented Naïve Bayes (TAN) e Markov Blanket que são usadas principalmente para classificação. |
|  | O nó SLRM (Modelo de Resposta de Auto-Aprendizagem) permite criar um modelo no qual um único novo caso, ou um pequeno número de novos casos, possa ser usado para reestimar o modelo sem precisar treinar novamente o modelo usando todos os dados. |
|  | O nó K-Nearest Neighbor (KNN) associa um novo caso à categoria ou valor dos objetos k mais próximos a ele no espaço do preditor, em que k é um número inteiro. Casos semelhantes estão próximos e casos diferentes são distantes um do outro |

* **Modelos de Associação**

Os modelos de associação encontram padrões em que uma ou mais atributo está associado a outros atributos. Os modelos constroem conjuntos de regras que definem esses relacionamentos. Aqui, os campos dentro dos dados podem atuar como entradas e destinos. Podendo encontrar essas associações manualmente, mas os algoritmos de regra de associação são muito mais rápidos e podem explorar padrões mais complexos. [555]

Nós de associação:

|  |  |
| --- | --- |
| Apriori node | O nó Apriori extrai um conjunto de regras dos dados, retirando as regras com o maior conteúdo de informações. A Apriori oferece cinco métodos diferentes de seleção de regras e usa um sofisticado esquema de indexação para processar grandes conjuntos de dados com eficiência. |
| CARMA node | O modelo CARMA extrai um conjunto de regras dos dados sem exigir que sejam especificados campos de entrada ou destino. Ao contrário de Apriori, o nó CARMA oferece configurações de construção para suporte a regras (suporte para antecedentes e consequentes), em vez de apenas suporte antecedente. |
| Sequence node | O nó Sequence descobre regras de associação em dados sequenciais ou orientados a tempo. Uma sequência é uma lista de conjuntos de itens que tendem a ocorrer em uma ordem previsível. |

* **Modelos de Segmentação**

Os modelos de segmentação dividem os dados em segmentos ou clusters de registros que possuem padrões semelhantes de campos de entrada. Comumente conhecidos como "modelos de cluster" são úteis nos casos em que o resultado específico é desconhecido. Os modelos de cluster se concentram na identificação de grupos de registros semelhantes e na rotulagem dos registros de acordo com o grupo ao qual eles pertencem. [555]

|  |  |
| --- | --- |
|  | O nó Cluster automático estima e compara modelos de cluster, que identificam grupos de registros que possuem características semelhantes. O nó funciona da mesma maneira que outros nós de modelagem automatizados, permitindo experimentar várias combinações de opções em uma única passagem de modelagem. |
| K-Means node | O nó K-Means agrupa o conjunto de dados em grupos distintos (ou agrupamentos). O método define um número fixo de clusters, atribui iterativamente registros aos clusters e ajusta os centros de cluster até que refinamentos adicionais não possam mais melhorar o modelo. |
| Kohonen node | O nó Kohonen gera um tipo de rede neural que pode ser usada para agrupar o conjunto de dados em grupos distintos. Quando a rede estiver totalmente treinada, os registros semelhantes devem estar próximos no mapa de saída, enquanto os registros diferentes estarão distantes. |
| TwoStep node | O nó TwoStep usa um método de cluster em duas etapas. A primeira etapa faz uma única passagem pelos dados para compactar os dados de entrada brutos em um conjunto gerenciável de sub-clusters. A segunda etapa usa um método de cluster hierárquico para mesclar progressivamente os sub-clusters em clusters cada vez maiores. |
|  | O nó Detecção de anomalias identifica casos incomuns, ou outliers, que não estão em conformidade com os padrões de dados "normais". Com esse nó, é possível identificar discrepantes, mesmo que eles não se ajustem a nenhum padrão conhecido anteriormente. |

Cabe enfatizar que as técnicas de *Modeling* anteriormente citadas têm particularidades em cada técnica, tendo conjuntos de algoritmos processados através de computação na nuvem para gerar os modelos preditivos, sendo uma etapa crucial a seleção correta do algoritmo, necessitando de detalhes a serem considerados. O Guia de Algoritmos do IBM SPSS Modelerm traz descrições dos fundamentos matemáticos dos métodos de modelagem usados no IBM SPSS Modeler

Os nós de *Output* fornecem os meios para obter informações sobre seus dados e modelos. Eles também fornecem um mecanismo para exportar dados em vários formatos para interagir com suas outras ferramentas de software.

A seguir os nós de *Outputs* serão descritos na tabela 200:

|  |  |
| --- | --- |
|  | O nó Tabela exibe os dados em formato de tabela, que também podem ser gravados em um arquivo. Isso é útil a qualquer momento em que é necessário inspecionar valores de dados ou exportá-los de forma facilmente legível. |
|  | O nó Matrix cria uma tabela que mostra os relacionamentos entre os campos. É mais comumente usado para mostrar o relacionamento entre dois campos simbólicos, mas também pode mostrar relacionamentos entre campos de sinalizador ou campos numéricos. |
|  | O nó Análise avalia a capacidade dos modelos preditivos de gerar previsões precisas. Os nós de análise realizam comparações entre valores previstos e valores de entrada para geração do modelo. Ele também pode comparar modelos preditivos entre si. |
|  | O nó Auditoria de dados fornece uma primeira visão abrangente dos dados, incluindo estatísticas resumidas, histogramas e distribuição para cada campo, além de informações sobre valores extremos, valores ausentes e extremos. Os resultados são exibidos em uma matriz de fácil leitura que pode ser classificada e usada para gerar gráficos em tamanho real e nós de preparação de dados. |
|  | O nó Transformação permite selecionar e visualizar visualmente os resultados das transformações antes de aplicá-los aos campos selecionados. |
|  | O nó Estatísticas fornece informações resumidas básicas sobre campos numéricos. Ele calcula estatísticas resumidas para campos individuais e correlações entre campos. |
|  | O nó Means compara as médias entre grupos independentes ou entre pares de campos relacionados para testar se existe uma diferença significativa. |
|  | O nó Relatório cria relatórios formatados contendo texto fixo, bem como dados e outras expressões derivadas dos dados.Especificando o formato do relatório usando modelos de texto para definir as construções fixas de saída de texto e dados. |

A interface exclusiva do IBM® SPSS Modeler permite minerar dados, trabalhando com diagramas de fluxos de dados. No nível mais básico, é possível criar um fluxo de dados usando as seguintes etapas:

• Adição de nós à tela de fluxo.

• Conexão dos nós para formar um fluxo.

• Especificação das opções dos nós.

• Execução do fluxo.

Ao contrário dos métodos estatísticos mais tradicionais, não é estritamente necessário precisar saber o que está procurando quando se inicia uma avaliação de característica entre bancos de dados. Sendo possível explorar os dados, ajustando-se a diferentes modelos e investigar diferentes relacionamentos, até que sejam encontradas informações úteis [IBM KNOWLEDGE CENTER].

## **2.5 Medidas de Precisão**

Levando em consideração que as sistematizações em qualquer software de predição precisam considerar algum marco como referência para avaliar as condições das saídas geradas frente as entradas e as comparações realizadas, tem-se que aplicar alguns conceitos para a interpretação das saídas geradas a partir da execução da etapa de processamento desta dissertação.

Desta forma é necessário avaliar o preenchimento dos valores dos parâmetros da função kernel(tópico55.89) aplicada ao algoritmo de modelagem MVS, pois a seleção do tipo da função e a modificação dos parâmetros do kernel , C e tem influência direta sobre a execução da aplicação de MVS e as formulações dos resultados. Conduzindo a buscas para aplicação de parâmetros “ótimos” para obtenção de resultados satisfatórios, como em casos como [876] e [92] em que houve aplicação de métodos heurísticos de determinação dos parâmetros de kernel, estes parâmetros serão melhor discutidos no tópico 4.4.

Assim como o conhecimento da aplicação das funções kernel, outra característica que deve ser compreendida para avaliar os resultados, é conhecimento do nó *Analysis* (output), que é de fundamental importância para a compreensão das avaliações da aplicação de MVS sobre um banco de dados.

Para avaliar os resultados do nó Analysis tem-se a Fig 24 que mostra um exemplo de resultados obtidos.

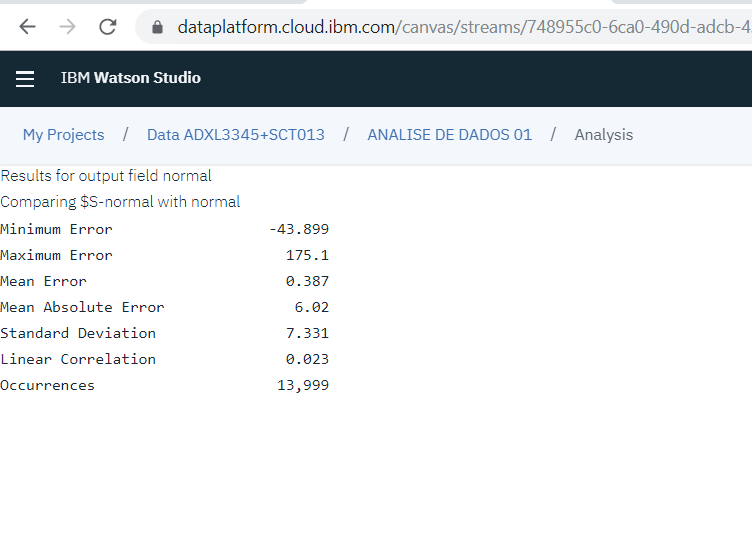
****

Figura 1- Saídas obtidas através de Analysis

O Modeler Applications[77]e [555], fornecem os as informações acerca de como avaliar os resultados de Analysis, que contém uma seção para cada campo de saída para o qual existe um campo de previsão correspondente criado por um modelo gerado.

A seguir as descrições das subseções de Analysis:

• **Erro mínimo:** Mostra o erro mínimo (diferença entre os valores observados e os previstos).

**• Erro máximo:** Mostra o erro máximo. (concatenar informações das ref)

• **Erro médio:** Mostra a média de erros em todos os registros. Indica se há uma tendência mais forte de superestimar do que subestimar o modelo. (verificar se é o MSE)

• **Erro absoluto médio (Mean Absolut Error-MAE):** Mostra a média dos valores absolutos dos erros em todos os registros.

• **Desvio Padrão:** Mostra o desvio padrão dos erros.

• **Correlação linear:** Mostra a correlação linear entre os valores previstos e reais. Essa estatística varia entre -1,0 a 1,0. Valores próximos a 1,0 indicam uma forte associação positiva, de modo que valores previstos são associados a valores reais e baixos valores previstos, associados a baixos valores reais. Valores próximos a -1,0 indicam uma forte associação negativa, de modo que altos valores previstos são associados a baixos valores reais. Valores próximos a 0,0 indicam uma associação fraca, de modo que os valores previstos são um tanto quanto independentes dos valores reais.

Para revisar as descobertas sobre resultados encontrados através do Analysis deve-se compreender as saídas, como na figura x, tomada como exemplo de interpretação, em que:

|  |  |
| --- | --- |
| Correlação Linear | 0.023 |
| MAE | 6.02 |
| MSE | 0.387 |
| Erro máximo | 175.1 |
| Erro mínimo | -43,89 |
| Desvio padrão | 7.331 |

Avaliando os valores encontrados da correlação linear é possível identificar uma baixa relação encontrada entre os valores de predição e os valores reais, devido a correlação linear estar próximo dos 50% do total que se pode obter, trazendo uma avaliação de baixa acurácia do modelo gerado. De acordo com a literatura [876, ABBE et al] para bons valores entre a predição e os valores reais comparados, o valor de correlação linear deve ser superior a 80%, em trabalho [BIBLIOTE] chegou-se a 95% de correlação, observando-se também que os valores relacionados dos erros representam

# **CAPITULO 3**

# **3 MÁQUINAS DE VETORES DE SUPORTE**

## **3.1 Revisão Da Literatura**

O classificador MVS é fundamentado no princípio da Minimização do Risco Estrutural (Structural Risk Minimization-SRM) proposto por (VAPNIK; CHERVONENKIS, 1974). É possível definir o SRM com a função que minimiza o risco médio no conjunto de treinamento.

O princípio da minimização do risco estrutural busca minimizar o erro com relação a um conjunto de treinamento (risco empírico), assim como o erro com relação ao conjunto de testes, isto é, o conjunto de amostras não empregado no treinamento do classificador (risco de generalização). O objetivo do MVS consiste em obter um equilíbrio entre esses erros, minimizando o excesso de ajustes (overfitting) que podem reduzir a capacidade de generalização do classificador. O problema de overfitting está relacionado ao fato de o classificador memorizar os padrões de treinamento, gravando suas peculiaridades e ruídos, ao invés de extrair as características gerais que permitirão o reconhecimento de padrões do classificador, de acordo com (SMOLA et al., 2000).[876]

MVS é um método de aprendizado supervisionado que gera funções de mapeamento de entrada e saída a partir de um conjunto de dados de treinamento rotulados. A função de mapeamento pode ser uma função de classificação ou uma função de regressão.

Esta metodologia vem recebendo crescente atenção da comunidade de Aprendizado de Máquina (AM) [27]. Os resultados da aplicação dessa técnica são comparáveis e muitas vezes superiores aos obtidos por outros algoritmos de aprendizado, como as Redes Neurais Artificiais (RNAs) [4, 14]. Exemplos de aplicações de sucesso podem ser encontrados em diversos domínios, como na categorização de textos [19], na análise de imagens [20, 33] e em bioinformática [30, 34] e identificação de falhas [referencias].

MVS envolve poucos parâmetros livres que precisam ser ajustados pelo usuário e não há uma dependência, pelos menos de uma forma explícita, na dimensão do espaço de entrada do problema, o que sugere que MVS é útil em problemas com um grande número de entradas.

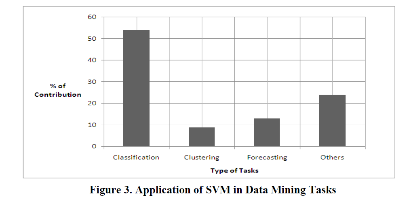
Essa técnica pode ser aplicada ao reconhecimento de padrões (estimar funções indicadores), regressão (estimar funções de valores reais) e extração de características. A caracterização do reconhecimento de padrões de falhas em motor de indução, investigado nesta dissertação, encaixa-se na área de reconhecimento de padrões, este capítulo descreve a teoria que fundamenta MVS dando ênfase a esta área.

A questão da generalização pode ser mais bem avaliada para o caso de duas classes. Assumindo que as amostras de treinamento das duas classes não são linearmente separáveis, a função de decisão mais adequada é aquela para qual a distância entre os conjuntos das amostras de treinamento é maximizada. O MVS é um classificador que utiliza o princípio de maximização da margem. O princípio da margem máxima coloca a superfície de decisão exatamente entre o limite das duas classes e maximiza a distância do limite das classes, figura X. Segundo (HAMEL, 2009), essa abordagem reduz a probabilidade de erro de classificação.[876]

Em relação ao SRM Vapnik (1998), mostra que, desde que o hiperplano de separação não cometa erros empíricos, ou seja, separe corretamente todos os exemplos de treinamento, maximizar a margem equivale a minimizar o limite superior do risco esperado. Desta forma ao separar um hiperplano com margem máxima, o risco de generalização será minimizado.[876]

A busca por uma superfície de decisão com critério ótimo como a margem, implica em um problema de otimização. Construir um classificador de margem máxima é um problema de otimização convexa que pode ser resolvido através de técnicas de programação quadráticas. Problemas de otimização são problemas nos quais desejamos a melhor solução em um número de possíveis soluções. A meta é encontrar a solução que maximiza ou minimiza o valor da função objetivo. A formulação matemática apresentada a seguir é baseada em (ABE, 2005).[876]

Conforme mostrado na Tabela 1, a máquina de vetores de suporte foi aplicada em quase todos os domínios de aplicativos, incluindo problemas de classificação, previsão / previsão, análise de imagens, reconhecimento de padrões, extração de regras e otimização. A maioria das aplicações que analisamos, cerca de 54%, estão na área de classificação. Os trabalhos nas áreas de agrupamento e previsão representam 9% e 13%, respectivamente, e outros representam 24% de cada um dos trabalhos revisados[ref]



## **3.2 MVS margens rígidas**

Seja um conjunto de treinamento em um problema que consiste em duas classes linearmente separáveis . Cada amostra fica associada aos rótulos para a para a . Se esses dados forem linearmente separáveis, a função de decisão pode ser determinada pela equação a seguir:

(4.1)

Nesse caso, é um vetor dimensional e é o termo independente. Onde representa a dimensionalidade dos dados, e é o número de amostras de treinamento, ou seja, os vetores e são representados por para bandas e para amostras.

A função linear divide o espaço em duas regiões e como podemos visualizar na Figura x.x, gerando o que é considerado hiperplano

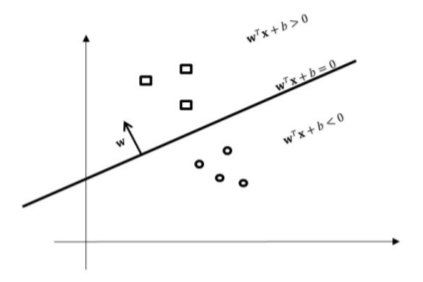


Figura 00: hiperplano separando amostras em duas regiões.

O processo de classificação das amostras pode ser formulado da seguinte forma:

(4.2)

sendo uma constante . As inequações (4.3) podem ser rearranjadas, dividindo ambos os membros por e aplicando uma transformação de escala em e e reescrevendo-a de forma mais compacta obtemos:

(4.3)

A equação acima descrita pode ser reescrita em uma só como se segue:

para (4.4)

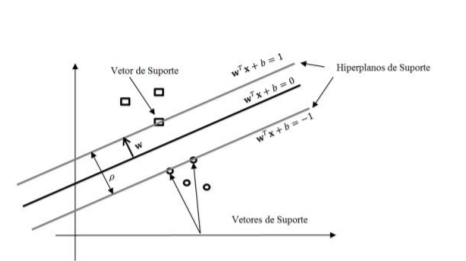


Figura x.3: O hiperplano ótimo separando os dados com margem máxima. Os vetores de suporte são as amostras que satisfazem as equações e . Adaptado de (HAMEL, 2009)[876].

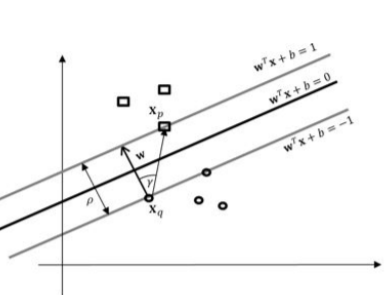


Figura 3.3: Margem entre os dois hiperplanos de suporte. Adaptado de (HAMEL, 2009).

Existem inúmeras funções lineares que separam as amostras de treinamento (Figura3.1). O objetivo do classificador MVS é encontrar um hiperplano ótimo que separe as duas classes de forma ótima. A equação define um hiperplano a meia distância entre os hiperplanos e . A distância entre esses hiperplanos é denominada de margem representada por na Figura3.2. Supondo a existência de pelo menos uma amostra para a qual , e outra amostra para a qual , então o hiperplano representa a melhor superfície de separação dessas amostras. As amostras que se encontram sobre e são os vetores de suporte. O primeiro hiperplano é o hiperplano de suporte para a classe +1 e fica sobre a superfície de decisão, o segundo hiperplano é o hiperplano de suporte para a classe -1 e fica sob a superfície de decisão.[876]

A largura da margem () pode ser calculada pela projeção da diferença entre os vetores de suporte em direção ao vetor normal da superfície de decisão, a construção pode ser visualizada na Figura3.3. A distância pode ser calculada através da seguinte equação:

= = = (4.5)

Como queremos maximizar a margem entre os hiperplanos de suporte isso implica em maximizar . No entanto podemos expressar esse problema de maximização com um problema de minimização , podendo ser reescrito como . De acordo com [876], a otimização sobre valores positivos é invariante sobre a transformação com funções quadráticas. A partir das considerações anteriores, se verifica que a maximização da margem de separação das amostras de treinamento, em relação a , pode ser obtida pela minimização de . A seguir apresentamos a formulação do problema de otimização:

Minimizar , sujeito a (4.6)

Por ser um problema convexo, a resolução de 4.6 pode ser realizada recorrendo aos multiplicadores de Lagrange. Em 4.6, temos que com e a restrição é linear. A restrição é inserida para assegurar que não existam amostras de treinamento na região de separação entre as classes. A introdução das restrições no problema de minimização pode ser resolvida por meio da técnica dos multiplicadores de Lagrange . A nova função objetivo é dada pela diferença entre a função objetivo anterior e o produto das respectivas restrições com os multiplicadores de Lagrange, segue a formulação do Lagrangiano primal:

sendo os multiplicadores de Lagrange um vetor de dimensão m com . A solução para este problema de extremos pode então ser obtida minimizando })com relação a , e maximizando com relação a

acrescidas das condições:

para (4.10)

As equações 4.7, 4.8, 4.9 são conhecidas como as condições de Karush-KunhTucker (KKT), (ABE, 2005)[876]. Substituindo as equações 4.8 e 4.9 em 4.7, obtém-se uma equação expressa em termos de :

O problema de otimização, portanto, se torna maximizar (3.11) com respeito à e sujeito as restrições:

e para

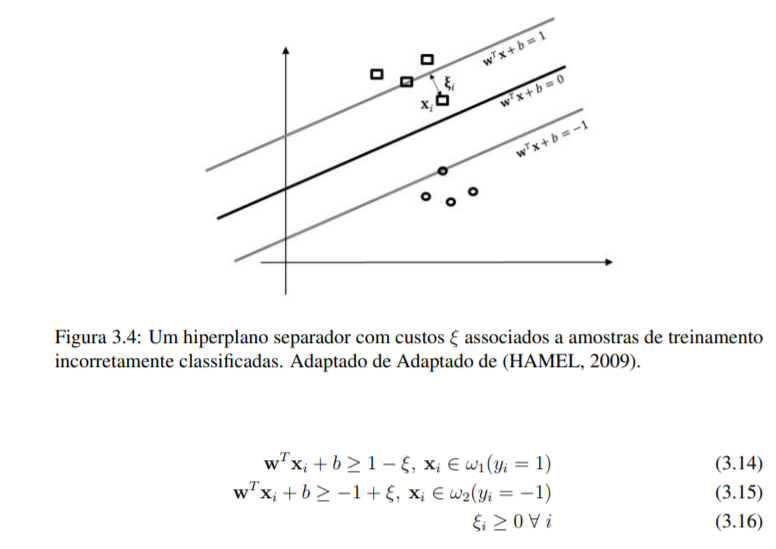
Essa formulação é denominada forma dual. Segundo (HAMEL, 2009), a visão dual do problema tem consequências interessantes. Uma das consequências mais relevantes é que um classificador linear baseado em MVS pode ser facilmente estendido para um classificador não linear. Na formulação dual as amostras de treinamento aparecem na forma de produto interno .

A forma dual apresenta como coeficientes diferentes de zero os vetores de suporte, isto é, para os exemplos de treinamento que verificam a igualdade na equação (4.4). Uma vez estimados os parâmetros que definem o hiperplano ótimo, constrói-se uma função de decisão:

## **3.3 MVS margem suave**

O problema de otimização (4.6) possui solução somente no caso das amostras xi pertencerem a duas classes linearmente separáveis. Em situações reais é pouco provável que duas classes sejam separáveis por um hiperplano no seu espaço original. O caso das classes não linearmente separáveis é tratado de forma idêntica, sendo, porém, necessário introduzir uma penalização às observações que se encontram do lado errado do hiperplano.[876] Os hiperplanos que introduzem essa penalização são conhecidos como hiperplanos de margem suave, conforme a Figura 3.8.

No caso não separável se permite que existam amostras de treinamento dentro da margem máxima. A nova definição dos hiperplanos de suporte é obtida introduzindo a variável de folga para cada ponto:



O procedimento de suavização da margem do classificador linear permite que alguns dados de treinamento permaneçam entre os hiperplanos de suporte. Além disso, permite também a ocorrência de alguns erros de classificação. Para o caso onde , a amostra correspondente não terá margem máxima, mas será rotulada corretamente. No caso de, a amostra será rotulada erroneamente.

Para que ocorra um erro é necessário que o respectivo seja maior que 1, assim é o limite superior para o número de erros de treinamento. Agora o problema de otimização deve minimizar os erros de treinamento. Assim se modifica a função objetivo para minimizar:

Onde é um parâmetro a ser escolhido pelo usuário, quanto maior o valor do parâmetro , maior será a penalização associada aos erros cometidos. A constante é conhecida como “parâmetro de margem” e estabelece a importância relativa das duas parcelas do lado direito da igualdade da equação (4.17) neste processo de minimização. A minimização de resulta na maximização da margem, enquanto que a minimização da segunda parcela resulta na minimização do erro de classificação. A solução de (4.17) é obtida de forma análoga ao caso separável e reescrevendo o problema primal Lagrangiano:[876]

Onde e

Conforme Abe (2005), para a solução ótima as condições de KKT, apresentadas a seguir, devem ser satisfeitas.

para

para

para

Determinam-se os multiplicadores de Lagrange positivos e , igualando as derivadas em ordem , e e, substituindo estas relações em (4.18), obtemos a seguinte formulação:

Maximizar

sujeito a para

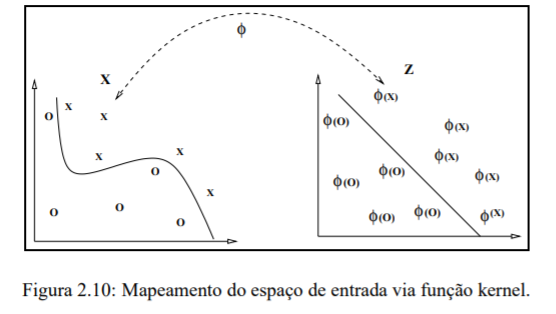
Ainda segundo Abe (2005), a única diferença entre o MVS de margem suave e o MVS de margem rígida é que não pode exceder C. A função de decisão é a mesma para o caso de margem rígida

## **3.4 Funções Kernel**

Apesar de o MVS apresentar bom poder de generalização (ABE, 2005), seu desempenho depende da seleção de parâmetros na função de kernel do classificador. A escolha de parâmetros inadequados pode resultar em decréscimo na acurácia dos resultados. Atualmente não existe um método universal para guiar a seleção de parâmetros do kernel.[876]

As representações Kernel trabalham com a projeção dos dados em um espaço de características com alta dimensão para permitir a classificação em espaços não-linearmente separáveis. Trata-se, em primeira instância, de uma estratégia de pré-processamento que envolve mudar a representação de dados da seguinte forma:

Esse passo é equivalente ao mapeamento do espaço de entrada em um novo espaço , chamado espaço de características em que é uma função kernel. A Figura 2.10 ilustra um mapeamento de um espaço de entrada linearmente inseparável, para um espaço de características de maior dimensão, onde os dados podem ser separados linearmente. É importante observar que ambos os gráficos representam espaços bidimensionais por razões puramente didáticas.



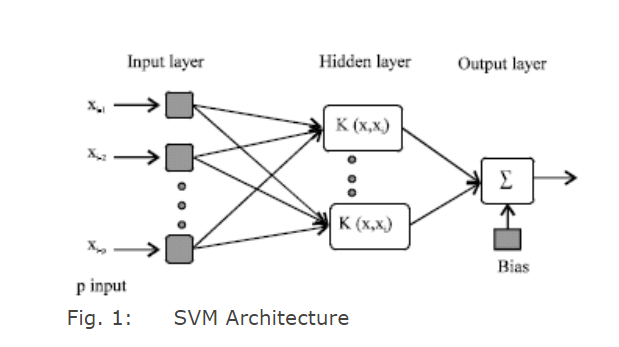
Dada essa representação mapeada, uma classificação ou regressão simples em pode ser feita. Outro ponto de destaque das funções Kernel é que o produto escalar pode ser computado implicitamente em Z, sem usar explicitamente ou mesmo sem conhecer o mapeamento, o que foi destacado na seção anterior.[449]

A escolha da função kernel é de vital importância para MVS. Segundo [171] o problema de escolher uma arquitetura para uma aplicação de Rede Neural é substituído pela escolha do kernel adequado para MVS.

As funções principais de kernel apresentam-se nas seguintes formas:

|  |  |
| --- | --- |
| Função Linear | ) |
| Polynomial Function |  |
| Função de base radial rbf |  |
| Função tangente hiperbólica |  |

É possível notar que γ, C e são parâmetros das kernel. Os parâmetros do kernel devem ser cuidadosamente escolhidos, pois definem implicitamente a estrutura do espaço de característica dimensional alta φ (x) e, portanto, controlam a complexidade da solução final. A arquitetura MVS é mostrada na Fig. 1.

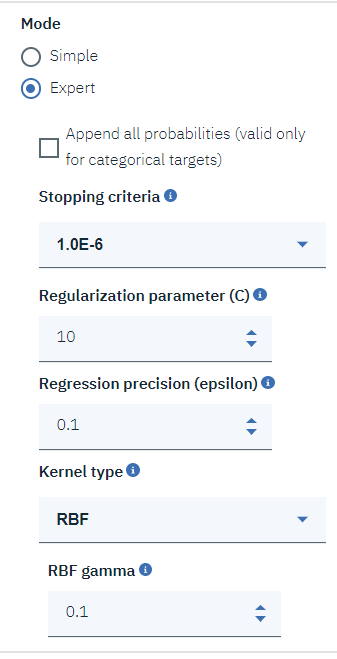


O multiplicador de Lagrange mede a influência do i-ésimo exemplo de aprendizagem no W. funcional. Os exemplos positivos são chamados vetores de suporte, pois definem o hiperplano de separação. é um parâmetro de custo constante, controlando o número de vetores de suporte e permitindo ao usuário controlar o trade-off entre erro de aprendizado e complexidade do modelo, considerado pela margem do hiperplano separador [41]. A regra de separação é dada pela função indicadora, usando o produto escalar entre o padrão a ser classificado (), os vetores de suporte e um limite constante .

Para construir funções de decisão não lineares mais gerais, os MVSs implementam a ideia de mapear os exemplos do espaço de entrada para um espaço de característica de alta dimensão, através de uma função de mapeamento não linear escolhida. A construção de um hiperplano de separação no espaço de características leva a uma superfície de decisão não linear no espaço original; veja a Fig. 7. Deixando os algoritmos praticamente inalterados, isso reduz significativamente a complexidade numérica e permite um aprendizado eficiente do vetor de suporte para exemplos de até centenas de milhares de unidades.

Comparado às redes neurais, o método MVS oferece um número significativamente menor de parâmetros. Uma característica importante é liberdade de modelagem consiste na escolha de uma função do kernel e nos parâmetros correspondentes do kernel, influenciando a velocidade da convergência e a qualidade dos resultados. A determinação dos parâmetros , C e são vitais para obter bons resultados de classificação, embora modificações algorítmicas possam simplificar ainda mais essa tarefa [16].

Dentro da aplicação de MVS no ambiente de IBM SPSS Modeler realizada neste trabalho, as modificações dos parâmetros ocorrem dentro de opções especializadas de MVS node, ainda na etapa de pré-processamento, como mostrado na figura 94, permitindo ajustar o processo de treinamento, após a seleção da ferramenta. Necessitando sair do modo *simple* e selecionar o modo *expert*, para variar valores de parâmetros ou modificar a função kernel usada na aplicação.



A partir dos conceitos obtidos em [modeler modeling nodesibm], é de fácil identificação do padrão de kernel aplicado na modelagem de MVS, o tipo RBF (*Radial Basis Function*). Sendo possível executar modificações diversas entre o tipo de função do kernel usada para a transformação, sendo, portanto, aconselhável experimentar as várias opções de kernel e verificação de validações de acordo com a definição dos respectivos parâmetros.

Serão feitas as seguintes definições desses parâmetros com base em [modeler modeling nodes]:

**Stoping Criteria:** Determina quando parar o algoritmo de otimização. Os valores variam de 1.0E-1 a 1.0E-6; o padrão é 1.0E – 3. Reduzir o valor resulta em um modelo mais preciso, mas o modelo levará mais tempo para treinar.

**Parâmetro de Margem)(C):** Controla o compromisso entre maximizar a margem e minimizar o termo do erro de treinamento. O valor normalmente deve estar entre 1 e 10, inclusive; o padrão é 10. O aumento do valor aprimora a precisão da classificação (ou reduz o erro de regressão) para os dados de treinamento, mas isso também pode levar ao ajuste excessivo.

**Precisão de Regressão ():** Usado apenas se o nível de medição do campo de destino for contínuo. Faz com que erros sejam aceitos, desde que sejam menores que o valor especificado no preenchimento de . Aumentar o valor pode resultar em modelagem mais rápida, mas à custa de perca de precisão.

**RBF Gama():** Ativado apenas se o tipo de kernel estiver definido como RBF. O valor normalmente deve estar entre 3 / k e 6 / k, onde k é o número de campos de entrada. Por exemplo, se houver 12 campos de entrada, vale a pena tentar valores entre 0,25 e 0,5. Aumentar o valor melhora a precisão da classificação (ou reduz o erro de regressão) para os dados de treinamento, mas isso também pode levar ao ajuste excessivo (overfited).

O modelo gerado através da simulação usando MVS cria um ou mais campos extras dependendo da aplicação. A seguir uma tabela de interpretação do novo campo criado após a simulação de MVS dentro do SPSS Modeler

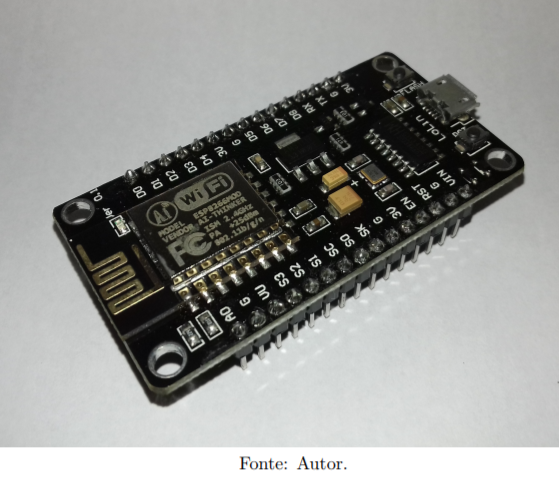
|  |  |
| --- | --- |
| Novo nome do campo | Descrição |
| $ S-Class | Valor para a classe prevista pelo modelo. |
| $SP-Class | propensão da classe SP para esta previsão (a probabilidade dessa previsão ser verdadeira, um valor de 0,0 a 1,0). |

# **CAPITULO 4**

# **4 MATERIAIS e MONTAGEM**

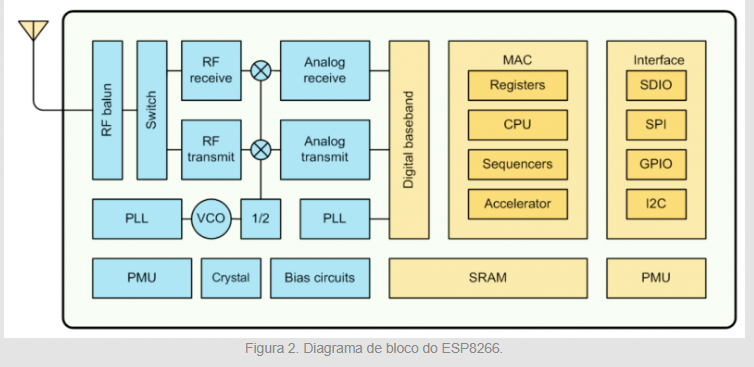
## **4.1 System On Chip Esp 8266**

ESP8266



O ESP8266 é um microcontrolador produzido pela empresa Espressif Systems. Uma característica relevante deste microcontrolador é que ele é capaz de fazer comunicação via Wi-Fi e tem um baixo custo, algo em torno de 3 usd. Além da comunicação por Wi-Fi (seguindo o protocolo IEEE 802.11 bgn), este microcontrolador conta com 16 portas GPIO, comunicação I 2C, UART, SPI, um módulo ADC, um CPU da Tensilica L106 de 32-bit com a arquitetura Xtensa, clock de 80MHz podendo chegar a 160MHz, sendo amplamente utilizado em projetos IoT.

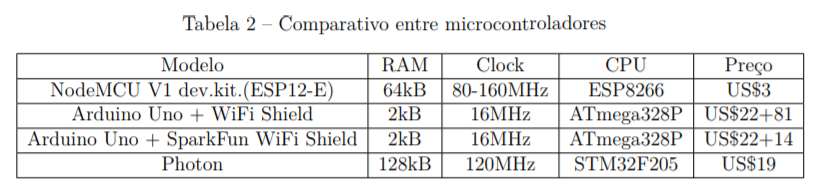
É um conjunto de alto desempenho, alta interação wireless, projetado para espaços pequenos com restrição de consumo de energia para plataformas móveis. Ele fornece a capacidade de incorporar Wi-Fi dentro de outros sistemas, podendo funcionar como aplicativo independente, com menor custo e com um mínimo de espaço. O diagrama de blocos do ESP8266 é ilustrado na Figura 2.



ESP8266 contém todas as ferramentas para se estabelecer uma comunicação Wi-Fi, como; conectores para antenas, RF balun (dispositivo que casa impedâncias), amplificadores, filtros. O microcontrolador pode rodar aplicações que usam Wi-Fi e também pode servir de adaptador Wi-Fi para outro microcontrolador, utilizando uma comunicação como I2C, SPI ou UART. A parte de RF do microcontrolador é formada pelos seguintes principais blocos: receptor, transmissor, gerador de clock de alta precisão, reguladores e gerenciador de energia.

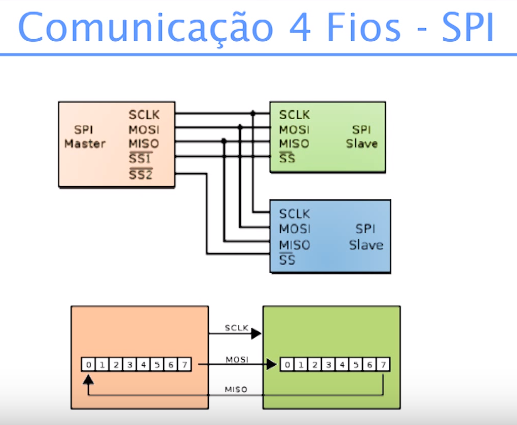
Para contornar os problemas gerados pelas condições adversas do canal de comunicação, são integrados no ESP8266 filtros RF, controle automático de ganho (AGC) e circuitos que cancelam o offset do nível DC. O transmissor converte o sinal em quadratura para um sinal em banda passante de 2,4GHz, e utiliza um amplificador CMOS para ligar a antena. Todos os componentes utilizados pelo gerador de clock estão integrados no microcontrolador.

O principal motivo da escolha deste microcontrolador foi seu preço e capacidade de processamento e forma de comunicação através do protocolo SPI, atendendo as demandas do projeto, na Tabela x é possível verificar algumas especificações e o preço de microcontroladores.



### 5.1.1Comunicação SPI

É um protocolo de dados serial síncrono usado por microcontroladores para se comunicar com um ou mais dispositivos periféricos rapidamente em curtas distâncias. Também pode ser usado para comunicação entre dois microcontroladores. Com uma conexão SPI, sempre há um dispositivo mestre (geralmente um microcontrolador) que controla os dispositivos periféricos. É uma conexão full duplex, o que significa que os dados são enviados e recebidos simultaneamente. A taxa de transmissão máxima é superior à do sistema de comunicação I2C.

A comunicação SPI possui algumas características básicas. Primeiramente os sinais de comunicação possuem uma direção fixa e definida. Isso significa que sempre existem dois transistores que definem o estado de um pin (Push-Pull). Esse recurso é uma das grandes diferenças entre outras comunicações seriais como I2C e OneWire, que possui o mesmo barramento de dados para os sinais de entrada e saída através do esquema de abertura (pull-up).

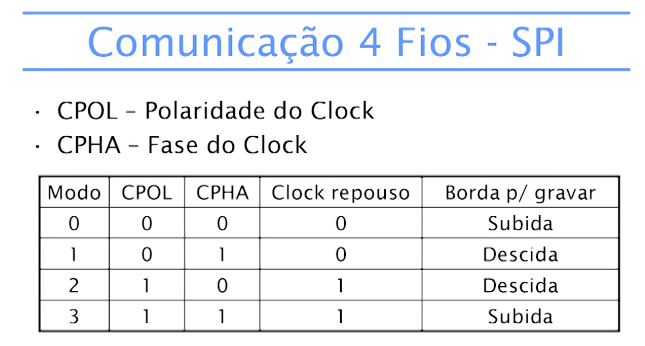
Normalmente, existem três linhas comuns a todos os dispositivos:

* MISO (Master In Slave Out) - A linha Slave para enviar dados ao mestre,

MOSI (Master Out Slave In) - A linha Master para o envio de dados aos periféricos,

* SCK (Serial Clock) - O pulso de clock sincroniza a transmissão de dados gerada pelo mestre.
* SS (Slave Select) - o pino em cada dispositivo que o mestre pode usar para ativar e desativar dispositivos específicos.

Além do clock e sinal de seleção esse tipo de comunicação possui potencial de fazer comunicação simultânea: basicamente a medida que os bits estão saindo do mestre(out), eles podem fazer o caminho inverso ao mesmo tempo. O mais comum é um trânsito de uma via somente.REFERENCIA X



Pode trabalhar em diversas polaridades do clock, desta forma se no repouso do clock está low/high, é possível identificar se o dado está disponível. A mudança do clock significa que tem-se um dado para processar.

Micro wire, trata somente da comunicação em MODE 0 em *half duplex*, enquanto somente um dado transita com dados.

Quando o pino de *Slave Select* está baixo, ele se comunica com o mestre. Quando está alto, ignora o mestre. Isso permite que seja possível vários dispositivos SPI compartilhando as mesmas linhas MISO, MOSI e CLK.

Para escrever uma rotina decódigo para um novo dispositivo SPI,é necessário observar os seguintes quesitos:

* Qual é a velocidade máxima do SPI que o dispositivo pode usar. Isso é controlado pelo primeiro parâmetro em *SPISettings*. Se for usado um um chip classificado em 15 MHz, usa-se 15000000.
* A entrada de dados é controlada pelo parâmetro SPISettings, MSBFIRST ou LSBFIRST. A maioria dos chips SPI usa o primeiro pedido de dados do MSB.

O padrão SPI é flexível e cada dispositivo o implementa de maneira um pouco diferente. Isso significa que deve-se prestar atenção especial as amostras de dados do dispositivo ao desenvolver uma rotina.

## **4.2 Accelerometer Mems Adxl 345**

Basicamente, todos os tipos de acelerômetros traduzem o sinal externo de aceleração em um deslocamento correspondente de sua massa móvel, também conhecida como massa inercial ou de prova. Este deslocamento pode ser detectado através de diferentes esquemas de medição, sendo que os mais comuns são: capacitivo, piezoelétrico, piezoresistivo, ressonante e óptico (YADI; AYAZI; NAJAFI, 1998; KRISHNAN et al., 2007).

As principais especificações que devem ser consideradas na seleção de um acelerômetro são discutidas em seguida.

• Sensibilidade: fator de escala de um sensor, medida em termos de mudança na saída para uma alteração no sinal de entrada. É uma referência à habilidade do sensor em detectar movimento e é normalmente especificada em mV/g;

• Largura de Banda (“Bandwidth”): é a faixa de frequências para a qual o sensor é aplicável. É normalmente especificado em Hertz(Hz), sendo tipicamente limitada a 1/5 da primeira frequência de ressonância (HARRIS; CREDE, 1976);

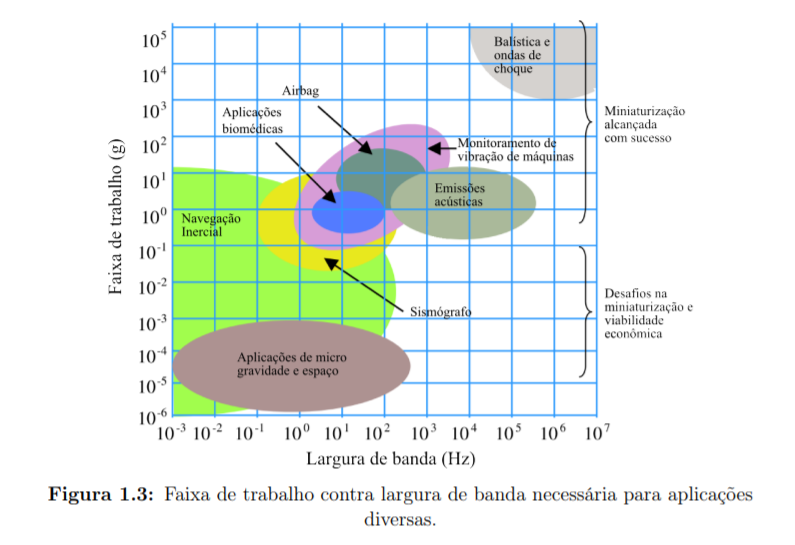
•Estabilidade: define o quão constante é o sinal de saída em condições de entrada constantes, sendo a alteração na saída chamada de deriva (“drift”). Acelerômetros para uso em aplicações de alto desempenho (aeroespaciais), devem apresentar alta estabilidade e por isso são muito mais caros que os utilizados em produtos de consumo, como videogames e celulares. O gráfico da Fig. 1.1, adaptado de Jean-Michel (2004), ilustra a importância desta característica na definição do preço de um sensor;

•Resolução: menor nível de aceleração detectável pelo sensor e é limitada pelo nível de ruído do sensor. É normalmente especificada em mili-g (mg) ou micro-g (µg);

•Alcance Dinâmico (“Dynamic Range”): é a faixa de valores de acelerações que podem ser medidas pelo sensor. O limite inferior é determinado pela resolução do dispositivo e o superior pela sua saturação, que é o ponto a partir do qual o sensor perde a linearidade na sua resposta;

•Confiabilidade: descreve a probabilidade do dispositivo desempenhar adequadamente suas funções, durante um período de tempo especificado e dentro de condições operacionais pré-estabelecidas;

• Custo: tem importância menor para dispositivos de alto desempenho, no entanto, é talvez a característica mais importante em aplicações de consumo.



As origens do uso da tecnologia do sistema micro-eletromecânico (MEMS) é datada de abril de 1954, quando um artigo de Smith (1954), depois do Bell Telephone Laboratories, foi publicado na Physical Review. Quando é descrita nas referências bibliográficas pela primeira vez certos efeitos sensíveis ao estresse no silício e no germânio, denominados piezoresistência. Em meados da década de 1950, os pesquisadores começaram a investigar se as mesmas tecnologias que produziram o transistor, que posteriormente revolucionou a incipiente indústria eletrônica, poderia ser aplicada aos sensores.

Mostrando que os sensores eletromecânicos volumosos anteriormente bastante utilizados podem ser substituídos por dispositivos pequenos e robustos da mesma maneira que o transistor substituiu a válvula termiônica. O artigo de Smith foi seguido no ano seguinte pela provavelmente a primeira publicação a considerar essa possibilidade (Paul e Pearson, 1955) e, no início dos anos 1960, a série de artigos do Honeywell Research Center e do Bell Labs descreveu a primeira pressão do diafragma de silício. sensores e extensômetros (Pfann e Thurston, 1961; Tufte et al., 1962). O interesse na tecnologia de sensores de silício cresceu dramaticamente e, no final dos anos 60, vários pioneiros americanos comercializavam os primeiros sensores de pressão de silício. Esses padrões eram brutos para os padrões atuais, mas no início dos anos 70 os desenvolvimentos em micromáquina, como era então chamado, e as melhorias no processamento de silício levaram a sensores com geometrias de que produziram desempenho superior.

Os Acelerômetros do tipo MEMS (Micro-ElectroMechanical Systems) detiveram grande impacto comercial com aplicação nos mais diversos campos. Sua criação foi motivada além de tudo pelo avanço da indústria para atender a diferentes tecnologias. Os acelerômetros comerciais são baseados na medição das componentes cartesianas do vetor de aceleração gravitacional, comumente encontrados no mercado oferecendo boa resposta à aceleração dinâmica resultante de movimento, baixa demanda de energia e baixa tensão de excitação. Suas dimensões permitem a integração estrutural ao conjunto do dispositivo a ser monitorado.

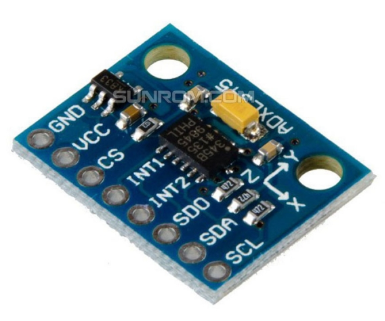
O acelerômetro piezoelétrico é o mais recomendado para aplicações em Manutenção Preditiva (MARTINS, 2000), pois a resposta de frequência estende-se até dezenas de kHz, deslocamento e velocidade podem ser obtidos a partir da integração elétrica da aceleração e, medição de transientes é geralmente melhor relatada por aceleração do que deslocamento ou velocidade. São também comercializados os do tipo IntegratedCircuitPiezoelectric, ICP, que possuem incorporado o circuito integrado de condicionamento do sinal. Este tipo de transdutor necessita, para isso, ser alimentado com uma corrente externa fornecida por outro dispositivo conhecido como SignalConditioner (figura 11). Em seguida, o sinal é enviado para o filtro antialiasing incorporado no CAD. Entretanto, o uso de acelerômetros é limitado devido ao relativo alto custo deste transdutor.



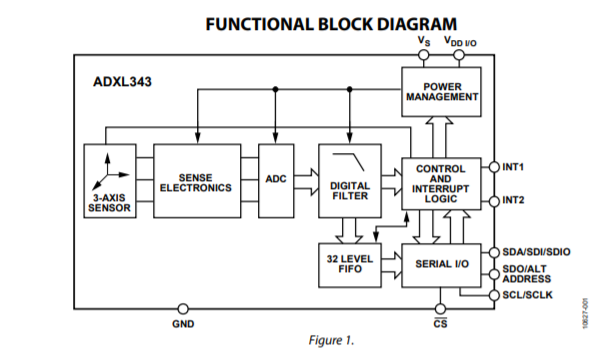
Neste trabalho foi analisado o comportamento do acelerômetro MEMS ADXL345 da AnalogDevice e um sensor sct013.

ADXL 345

O ADXL343 é um MEMS versátil de 3 eixos, saída digital e e facilmente configurado para faixas de trabalho diferentes. Faixas de medição e largura de banda selecionáveis além de detecção de movimento integrada e configurável o tornam adequado para a detecção aceleração em diversas aplicações. Robustez para 10.000 g de choque e uma ampla faixa de temperatura (-40 ° C a + 85 ° C) permitir o uso do acelerômetro mesmo em ambientes agressivos.



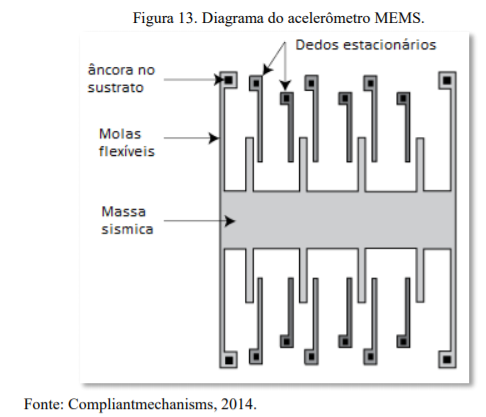
O ADXL345 mede a aceleração com alta resolução (13 bits) medição até ± 16 g. Os dados da saída digital são formatados de16 bits complementam e são acessíveis através de um SPI ou I2C. Ele pode medir a aceleração estática da gravidade em aplicações com sensor de inclinação, bem como a aceleração dinâmica resultante do movimento ou choque. Sua alta resolução (3,9 mg / LSB) permite a medição inclinação varia menos de 1,0 °.

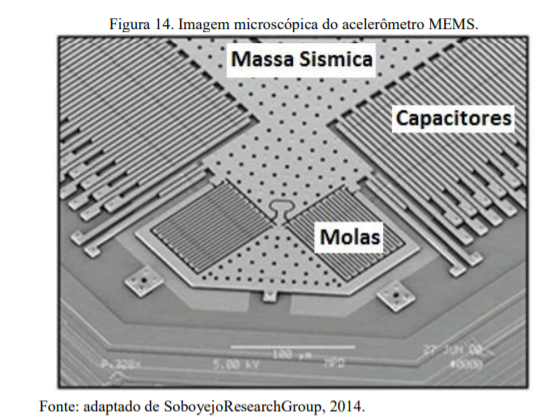


O ADXL343 possui uma faixa de medição selecionável de ± 2 g, ± 4 g, ± 8 g, ou ± 16 g. Ele mede a aceleração dinâmica resultante de movimento ou choque e aceleração estática, como a gravidade, que permite que o dispositivo seja usado como um sensor de inclinação.

O sensor é uma estrutura micro-usinada em superfície de polissilício construído em cima de uma pastilha de silicone. Molas de polissilício suspendem a estrutura sobre a superfície da bolacha e fornecer uma resistência contra forças devido à aceleração aplicada.

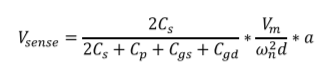
Agindo como um transdutor o qual permite transformar energia mecânica em energia elétrica, sendo sua fonte de informação a aceleração do sistema. É formado por três estruturas fundamentais, a massa sísmica, uma região de molas e as estruturas ou dedos capacitivos os dedos capacitivos se encontram nos dois lados da massa sísmica, e tem a mesma capacitância na posição de equilíbrio mecânico. Ao se aplicar uma aceleração na massa sísmica que conectada fisicamente com as placas dos condensadores, altera a distância ou a superfície entre os dedos capacitivos, alterando a capacitância do condensador. Essa capacitância é proporcional à aceleração aplicada no sistema (TEZ; AKIN, 2013).





A deflexão da estrutura é medida usando capacitores diferenciais que consistem em placas fixas independentes e placas fixadas a massa em movimento. A aceleração desvia a massa de prova e desequilibra o capacitor diferencial, resultando em uma saída do sensor cuja amplitude é proporcional à aceleração. Demodulação sensível à fase é usado para determinar a magnitude e a polaridade da aceleração.

O acelerômetro capacitivo MEMS é governado pela seguinte equação:

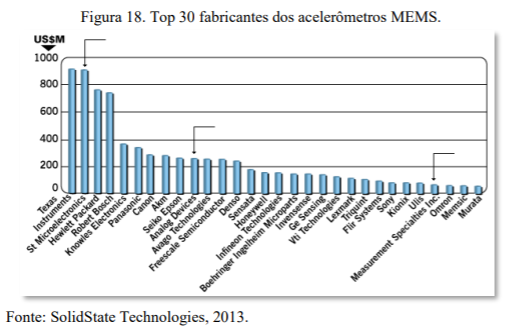


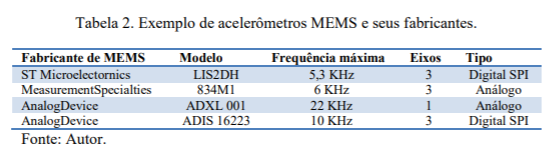
Onde Vsense é a tensão de saída, Cs é capacitância nos dedos capacitivos,Cp é a capacitância parasita, Cgs e Cgd são duas capacitâncias do transistor MOS, Vm é amplitude do sinal modulado, d é a distância entre os dedos capacitivos do sensor, wn é a frequência de ressonância mecânica do transdutor e, a é aceleração a ser medida.

Um acelerômetro típico MEMS tem frequência de ressonância 6 kHz, 1,5 µm na separação dos dedos capacitivos e uma sensibilidade de capacitância de 0,4 fF/g. Em um sinal de modulação de um 1 V de amplitude, a sensibilidade da tensão global é de cerca de 1 mV/g (JIANGFENG, 2004).

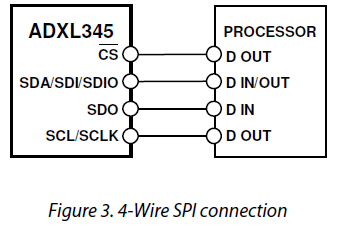
No caso da Manutenção Preditiva, as vibrações das máquinas rotativas geram sinais estacionários, que normalmente resultam da superposição de sinais senoidais com amplitudes e frequências diferentes. As características do acelerômetro que tem relação com esses dados são:

* Faixa de frequência (Bandwith) de medição.
* O limite de amplitude (Acceleration Range) que especifica o valor máximo de aceleração que pode ser medido com precisão, em “g”.





### 4.2.1 Spi No Adxl345



For SPI, either 3-wire or 4-wire configuration is possible, as shown in the connection diagrams in Figure 3 and Figure 4. Clearing the SPI bit in the **DATA\_FORMAT** register selects 4-wire mode while setting the SPI bit selects 3-wire mode. The maximum SPI clock speed is 5 MHz.

CS is the serial port enable line, and is controlled by the SPI master. It must go low at the start of transmissions and back high at the end as shown in Figure 5. SCLK is the serial port clock and is supplied by the SPI master. It is stopped high when CS is high, during period of no transmission. SDI and SDO are the serial data in and out respectively. Data should be sampled at the rising edge of SCLK.

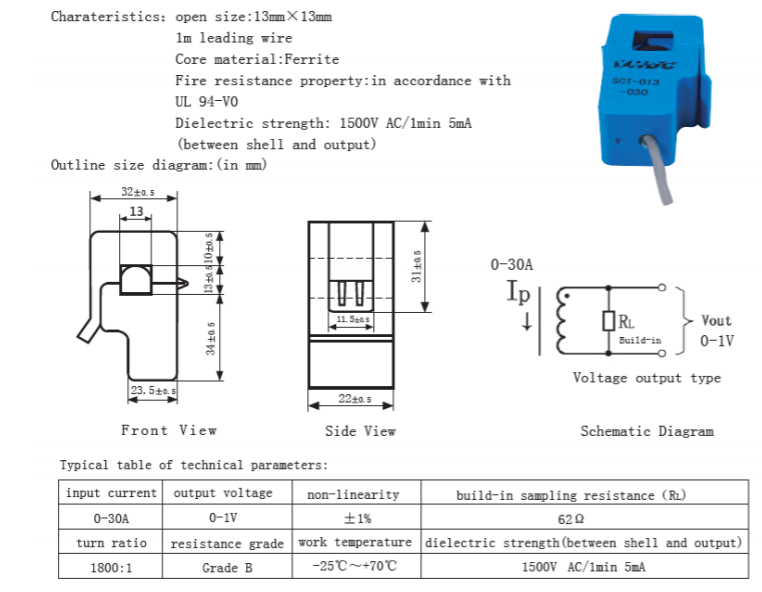
To read or write multiple bytes in a single transmission, the

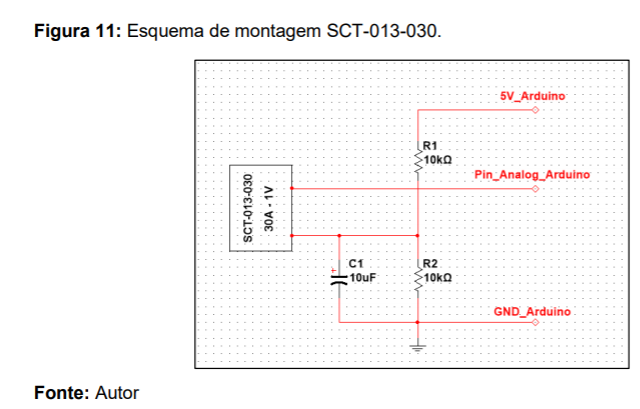
Multi-Byte bit, located after the R/W bit in the first byte transfer, must be set. After the register addressing and the first byte of data, continued clock pulses will cause the ADXL345 to point to the next register for read or write. Continued clock pulses will continue to shift the register that is pointed to until the clock pulses are ceased and CS is de-asserted. To perform reads or writes on different, non-sequential registers, CS must be de-asserted between transmissions and the new register must be addressed separately.

## **4.3 Sct-013**

Sensor SCT 013 (PODE IR NO APENDICE TB )

Sensor de corrente SCT-013-000 é uma ótima opção onde pode-se verificar correntes de até 100A e que não seja invasivo. Muito usado em projetos de automação residencial como medidores de corrente elétrica, proteção de motores AC/DC, iluminação e dentre outros.



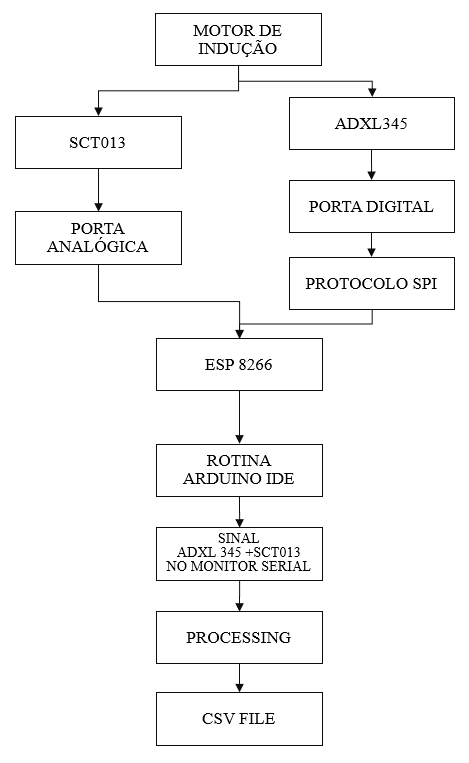


# **CAPITULO 5**

## **5 METODOLOGIA**

# **5.1 Sistema de aquisição**

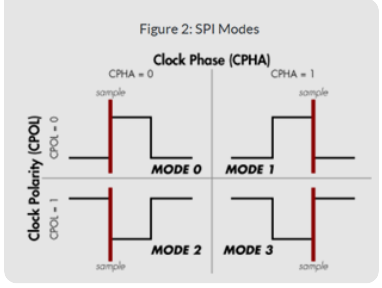
A metodologia deste trabalho consiste no treinamento de um modelo estatístico com base em dados de um motor de indução capturados por um sistema de aquisição embarcado. Os dados obtidos foram os de aceleração nos eixos x,y e z e os valores de corrente na porta analógica do esp8266.



Os dados do acelerômetro ADXL345 foram obtidos através de uma comunicação serial via protocolo SPI, a comunicação SPI foi escolhida em face ao protocolo I2C pois uma taxa de saída de dados deve ser selecionada apropriadamente de acordo com o protocolo de comunicação e a frequência escolhida afim de evitar que amostras de forças de aceleração sejam descartadas. No caso deste trabalho a mais rápida frequência, de 3200hz, é solicitada para obter-se uma maior quantidade de valores de forças de aceleração para um mesmo período de tempo. Portanto, segundo recomendação do fabricante no datasheet do ADXL345 quando uma taxa de transmissão de dados de 3200hz é escolhida o protocolo SPI deve ser utilizado.

Para prosseguir a configuração da comunicação entre o controlador e o sensor definiu-se o modo, sugerido pelo fabricante no datasheet do sensor. Esse modo define qual estado, dentre os 4 possíveis, que são obtidos a partir da combinação dos valores possíveis de polaridade do clock(CPOL), high ou low, e os valores possíveis da fase do clock (CPHA), high ou low.

Afigura representa os 4 possíveis estados para esses parâmetros e o correspondente modo SPI.

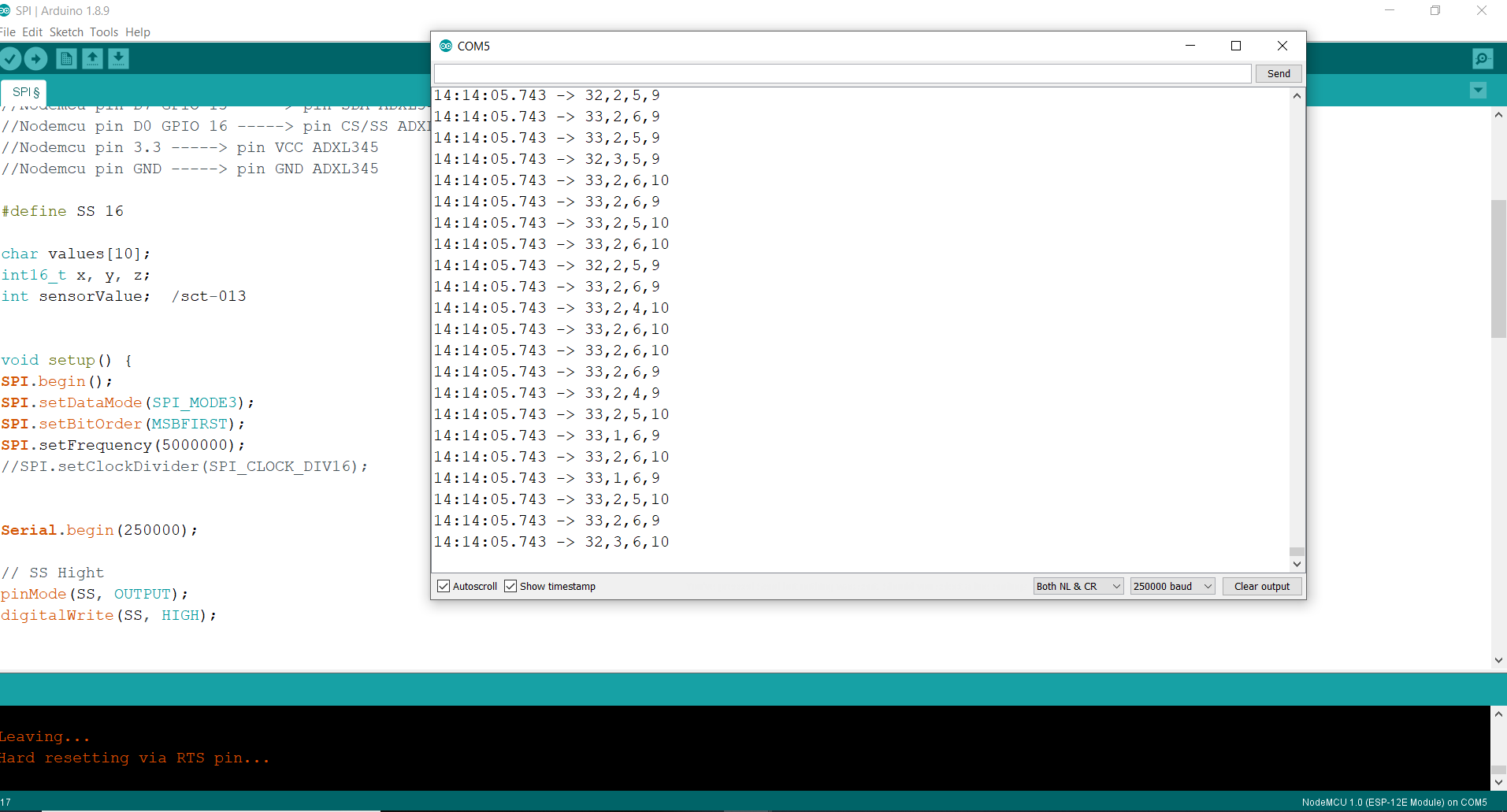


O Mode 3 foi escolhido pela análise do diagrama de timing do ADXL345.

O diagrama mostra que o estado *idle* do clock está no nível *high*, a borda de transmissão é a do nível *high* para nível *low* e a borda válida SDO é a borda de nível l*ow* para nível high.

O desenvolvimento do sistema objetivando a aquisição do sinal de corrente e vibração se deu através de um código escrito na Arduino IDE para o ESP8266 o qual faz a leitura dos registros do ADXL345 e a leitura dos valores obtidos na porta analógica na qual o SCT-013 é conectado e os envia para uma porta serial do computador. A partir desta etapa um código escrito no software *Processing*(APENDICE90) lê a porta em que o ESP8266 encontra-se conectado e constrói um arquivo CSV.

Após a concretização do sistema de monitoramento de vibração e corrente, foram realizados os primeiros testes do funcionamento do sistema. Segue a foto da visualização dos resultados obtidos da captação de sinais dos sensores ADXL345+SCT-13 controlados do ESP 8266, tendo a visualização dos resultados no Serial da IDE(*I*[*ntegrated Development Environment*](https://en.wikipedia.org/wiki/Integrated_development_environment)) do Arduino. A rotina de implementação consta no anexo 10.



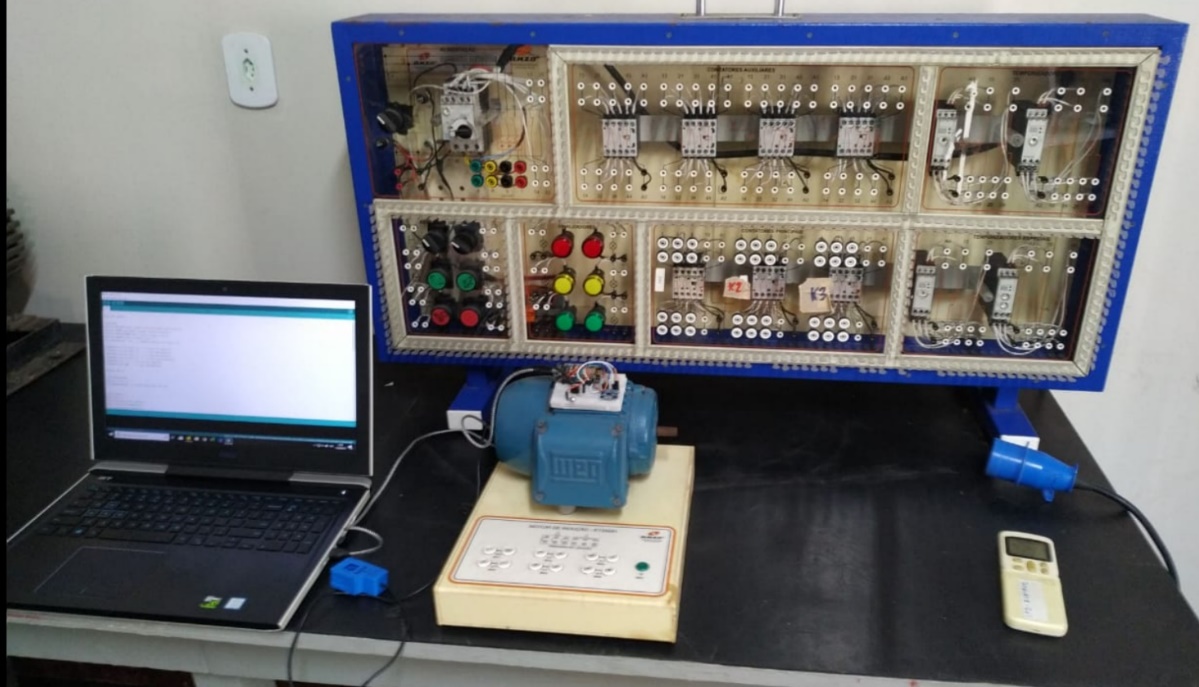
Com a ferramenta de medição de vibração e corrente operando, tornou-se necessário elencar as condições em que seriam executados os testes em um sistema físico. Uma revisão bibliográfica foi realizada para verificação dos principais tipos de falhas que podem ser avaliados segundo análises de sinais de corrente e vibração, como verificado no cap 02, objetivando, fundamentar a seleção de condições de avaliação do motor para este estudo. Os principais tipos das falhas externas sofridas descritas por motores serão listadas a seguir[referencias]:

* Desbalanceamento no rotor.
* Alimentação de tensão inferior.
* Alimentação de tensão superior.
* Alimentação de tensão desbalanceada.
* Sobrecargas no rotor.
* Rotor bloqueado.
* Inversão de fase.
* Desníveis nas instalações de fixação do motor.
* Falta de aterramento das instalações do motor.

Sendo desta forma selecionadas as seguintes condições do motor de indução trifásico, ajustadas as condições de realização dos testes em laboratório:

* + - Normal.
    - Desbalanceamento do rotor.
    - Alimentação de Tensão por Duas Fases.
    - Desnível na base do motor.

O Instituto Florence de Ensino disponibilizou suas instalações laboratoriais para que pudessem ser realizados os testes. A realização de testes e gravação dos dados de cada uma das condições acima citadas foi realizada, o tempo de cada ensaio durou 10 segundos (verifique apêndice testes).As figuras 41 e 42 mostram as instalações de onde ocorreram os testes.



Painel de instalação elétrica para ligação de partida direta no motor de x cv(descrições do motor no anexo x).

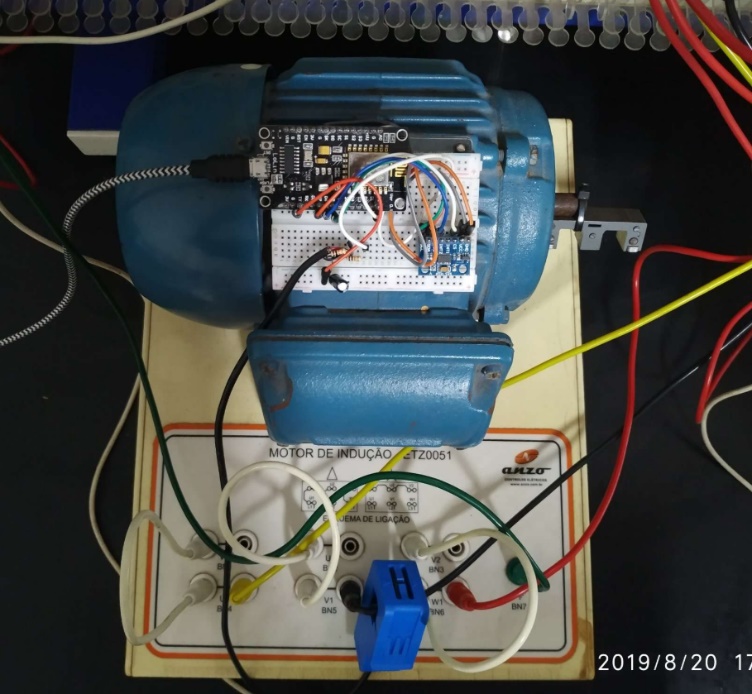


Figura 1 legeda da legenda

O ADXL345 fixado sobre a parte superior do motor, enquanto o SCT 013 fora conectado a um fluxo de fase de corrente de entrada do motor.

Para armazenar os dados em um arquivo no formato CSV, foi implementado o software *Processing*, que forneceu subsídios necessários para a indexação do fluxo dos dados a um novo arquivo, enquanto fosse requerida essa ação, em períodos de tempo determinados. Estes dados foram organizados em colunas em uma planilha sendo que para cada teste descrito foi gerado um arquivo.

A disposição das leituras do sistema de monitoramento de vibração e corrente gravou os dados de acordo com a foto 81

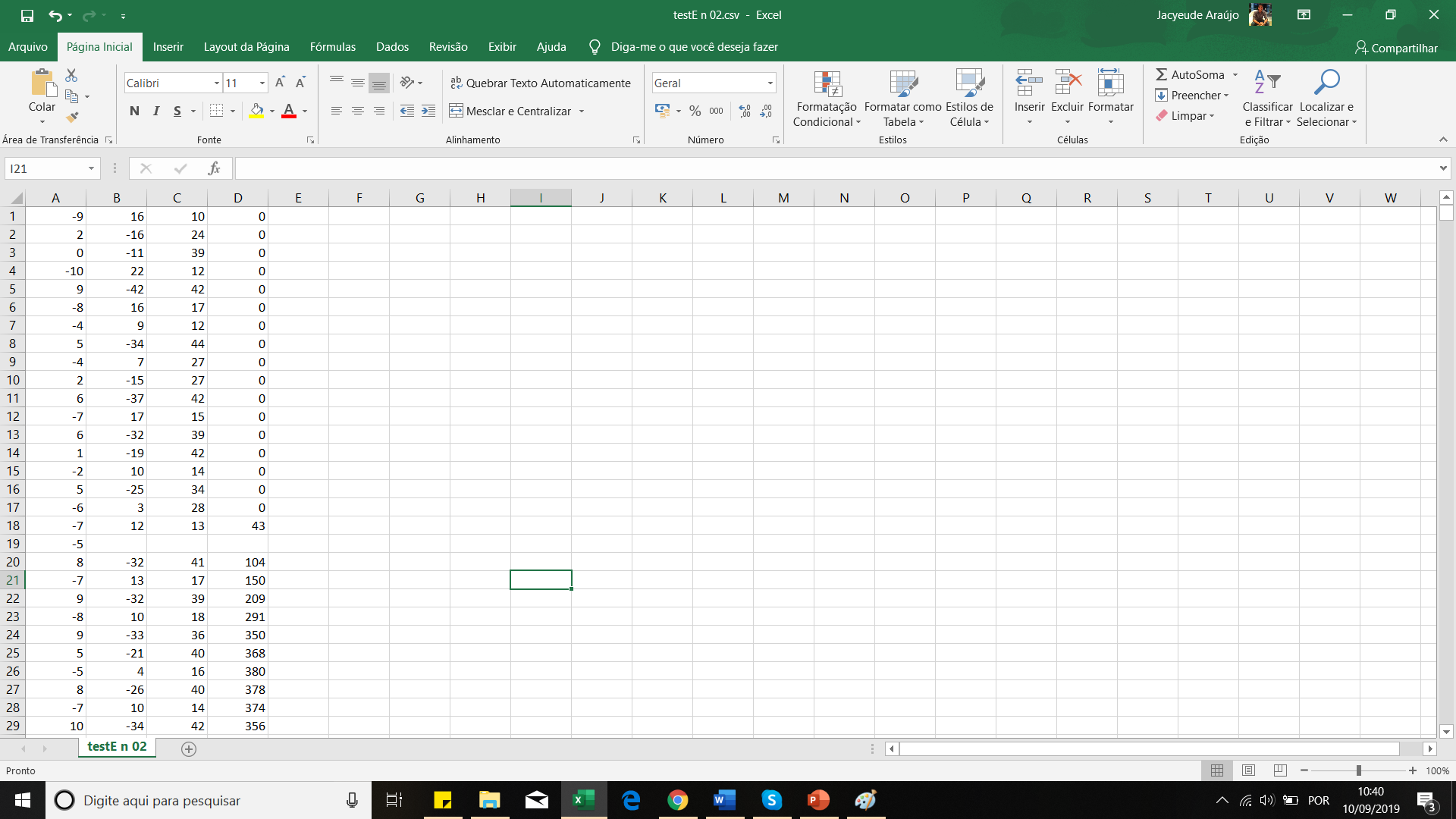


Figura 2 - leganda

As aquisições dos dados foram realizadas, salvas em formato, sendo as três primeiras colunas, representação de dados obtidos de x, y e z respectivamente, e a quarta coluna representa os dados de fluxo da corrente elétrica obtidos da leitura da porta analógica do ESP8266 conectado ao sensor SCT013. E desta forma foi concebido o material fundamental para dar sequência à etapa de implementações em computação na nuvem.

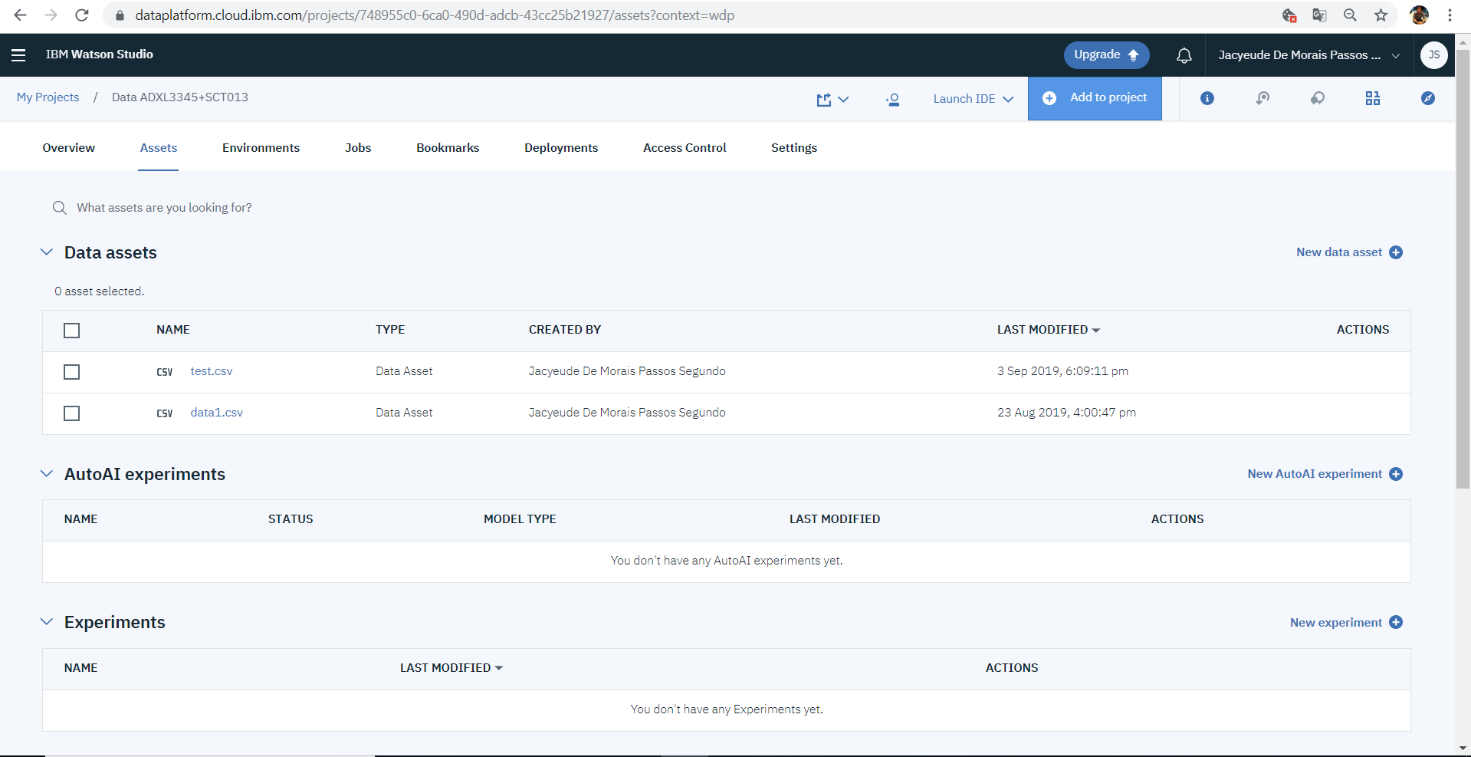
## **5.2 Sistema de Implementação**

Após ter êxito no armazenamento de dados dos testes efetuados no motor de indução em diferentes condições (normal, alimentação de tensão por duas fases, desbalanceamento do rotor e desnível na base). Seguiu-se o fluxo do projeto (fluxo55), iniciando a utilização da infraestrutura fornecida por IBM Cloud: Watson Studio, SPSS Modeler e Watson Machine Learning. A implementação da ferramenta estatística MSV objetiva através dos resultados obter a correlação entre os dados e avaliar a acurácia das análises frente ao modelo preditivo gerado, possibilitando a identificação de características dos dados inerentes a cada teste físico realizado. Na figura o fluxo de implementação.



Figura 3 - legenda

A figura 39 mostra a interface de Watson Studio, que disponibiliza uma aba de criação de um novo projeto e gaps de adição de novos *assets types* onde se encontra o *modeler flow*, que dá acesso à SPSS Modeler, a ferramenta de mineração de dados selecionada para implementar algoritmos preditivos que auxiliam na identificação de características entre os dados dos testes realizados no motor de indução



Dando seguimento às etapas, é necessário que se tenha conhecimento de quais diretrizes devem ser tomadas de acordo com as proposições de estudo, pois cada estudo de caso requer ajustes diferentes em etapas anteriores ao processamento dos dados. Para este trabalho, a parte subsequente ao upload de dados é verificar, através de etapas de preparação de dados, para deixá-los em condição ideal de aplicação do algoritmo de MVS, esta etapa é compreendida como pré-processamento.

### 5.2.1 Pré-processamento

É importante ressaltar que o conjunto de dados do estudo de caso precisou passar por etapas de pré-processamento utilizando a refinaria de dados (retirada de espaços vazios, identificação e seleção da quantidade de dados do arquivo farão parte das análises), bem como foi utilizado o nó *type* do SPSS Modeler para transformar o conjunto de dados, organizando a leitura das colunas e descrevendo as colunas em valores contínuos, figura 84. A etapa de preparação é muito relevante pois interpretação dos valores das leituras dos dados organizados em colunas deve ser compreendida como valores numéricos, para este caso em especifico, para esta aplicação em SVM, faz-se a seleção em *Measure* do set *Continuous*, para a interpretação dos valores das colunas. Desenvolvendo uma organização prévia à aplicação que será realizada.

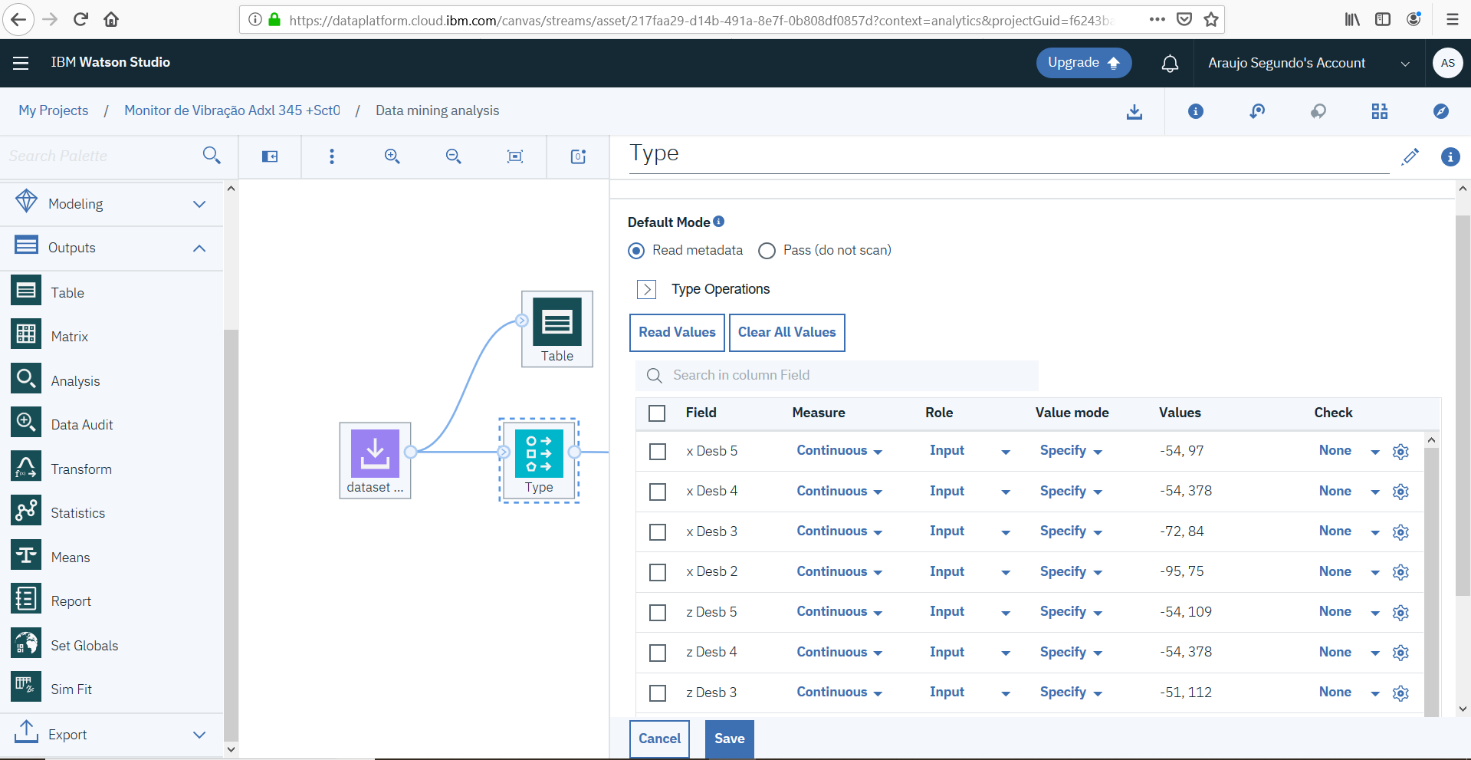
Ainda dentro das configurações de *type*, é necessário na opção *role* selecionar o alvo da predição (*field target*), selecionando a coluna que será confrontada com campos específicos selecionados como entradas (*field inputs*), que serão utilizados na simulação.

Figura 4 - figura 84

O nó *partition* divide o banco de dados em duas partes, configurando a simulação em etapas de treino e teste. Utilizando 80 % do banco de dados para treino e 20% para testar o modelo, viabilizando uma verificação da capacidade de generalização do classificador MVS através do teste dos dados com o modelo gerado a partir do treinamento. A figura x mostra a divisão do banco de dados em *partition*.

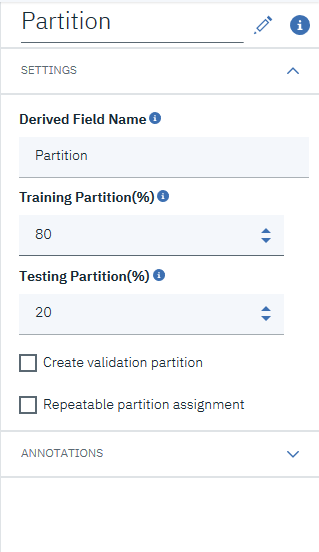


Figura 5- legenda

### 5.2.2 Execução do método

Nesta seção serão descritos os resultados dos experimentos que tem por finalidade validar a metodologia descrita na seção anterior. Para esta finalidade gravação de sinais dos sensores em arquivo CSV e upload na infraestrutura da IBM Cloud em Watson Studio, torna-se necessário desenvolver um fluxo de processamento dentro de SPSS Modeler.

Nesta etapa é onde ocorre a mineração de dados propriamente dita, momento em que se aplica um algoritmo para extração de conhecimento. No caso deste trabalho foram utilizado algoritmos de classificação, dentro de atividades preditivas, selecionando na aba de modelagem, o MVS. A implementação do algoritmo necessita de ajustes nos dados de entrada, como seleção de inputs, target além da seleção do kernel após selecionar o nó MVS na aba de *Modeling* (figura x).

A confiabilidade da análise dados é garantida por uma precisão de acurácia dos testes, que possui tendências que foram revisadas na literatura[][][]. A utilização das funções kernel deve ser avaliada, tornando-se indispensável para visualização das tendências de classificação das condições. Os resultados obtidos através do nó *Alanylsis* indicam as características do modelo gerado, que podem ser verificadas, em termos de correlação e indica o próprio dimensionamento da margem máxima de separação das classes verificadas através dos valores dos erros, sendo possível identificar características como sobre ajuste (overfited) do modelo que é gerado.

Os data sets, em formato CSV utilizados, foram estruturados de forma a organizar as etapas de verificação sobre cada uma das condições ensaiadas no motor de indução. Foram utilizados arquivos contendo informações de dois ensaios distintos, gerando um arquivo para analisar cada condição. Os arquivos foram organizados dessa forma para que possam ser utilizados dados que representem momentos diferentes de condições diferentes, usados como amostras de treinamento e teste para MVS, objetivando verificar a capacidade de identificação das classes através da aplicação da metodologia proposta.

Os arquivos usados contendo a base de dados das simulações estão descritos abaixo:

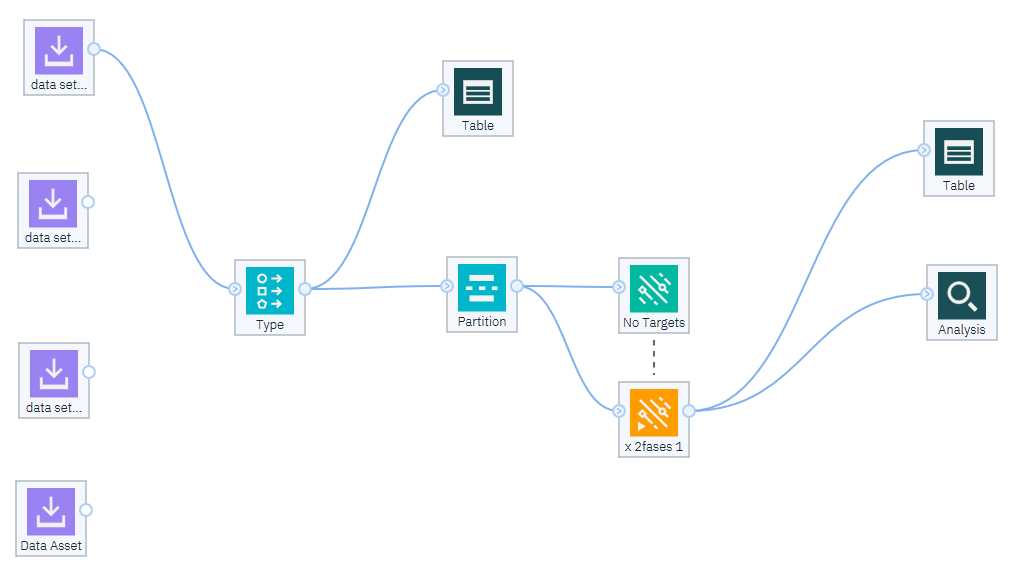
Data set 1 :Condições Normais

Data set 2: Condições de Desbalanceamentos

Data set 3: Condições de ligação de duas fases

Data set 4 :Condições de Desnível na base

## **5.3 Modelos Preditivos Gerados**



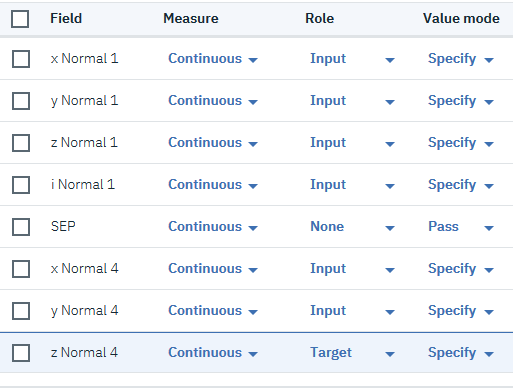
A figura x contém um fluxo da estruturação no SPSS Modeler para a aplicação de MVS realizadas para o base de treinamento e teste dos dados, gerando um novo nó(em amarelo na figura x), que é o modelo da predição gerado através das funções de decisão de MVS, a verificação desse resultado gera uma coluna $filename,(figura x), que deve ser investigada com a finalidade de extrair conhecimento das informações dispostas.

Os bancos de dados abaixo descritos contêm os valores de x y z do acelerômetro e os dados de corrente das condições ensaiadas no sistema físico, de modo que cada arquivo contém dados de dois ensaios de condições distintas para serem estudados individualmente, cada arquivo de *dataset* possui 10000 linhas de dados.

Usando as amostras de cada *dataset* para gerar o modelo e uma amostra separada para testá-lo, sendo possível obter uma boa indicação de quão bem o modelo será generalizado para conjuntos de dados que são semelhantes aos dados usados para as simulações.

### 5.3.1 *Dataset* 1 – Condição Normal

Para avaliar se MVS é capaz de reconhecer uma característica da condição de funcionamento normal do motor de indução foram selecionadas como alvo(*target*) a coluna “y Normal 1” e outras sete colunas de representação da condição do funcionamento normal foram selecionadas como entradas(*inputs)*, para a execução de MVS(figura x).



Após rodar a aplicação de MVS, é possível notar que foi criada uma nova coluna “$-z Normal4”, e através de *Analysis*, foram obtidos os resultados de treino com 80% dos dados, e com 20% utilizado para testar o modelo, neste caso validando as características encontradas ainda na fase de treino. (figura y). Para avaliar a performance de MVS foram realizadas experimentações com os 4 tipos de funções kernel, os resultados seguem:

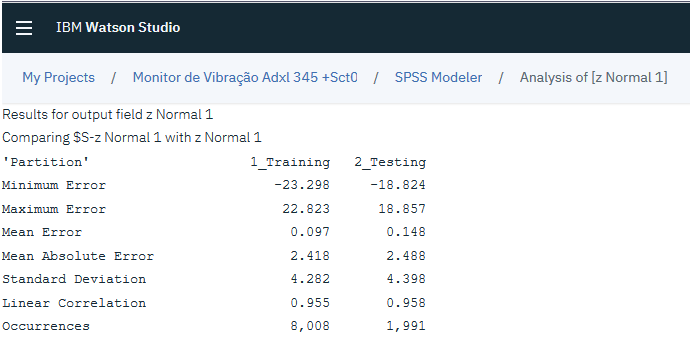


Figura 6´RESULTS KERNEL RBF Figura 7 - KERNEL POLYNOMIAL

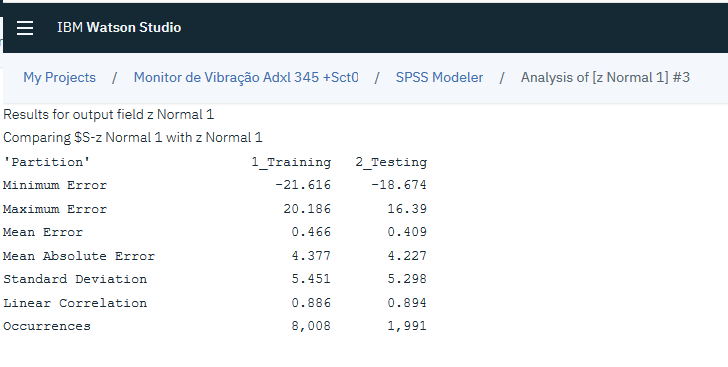
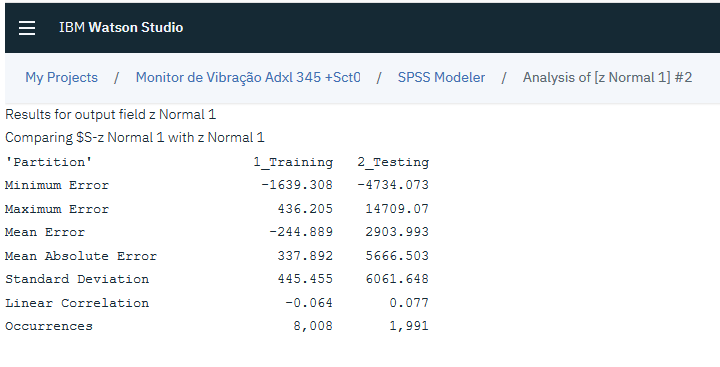
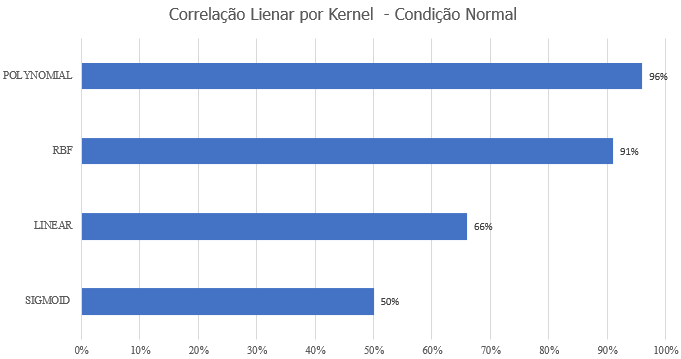


Figura 8 - KERNEL SIGMOID Figura 9- kernel linear

Avaliando que as aplicações de MVS foram feitas utilizando os parâmetros , e em todos os kernels em *default*, observando que o ajuste desses parâmetros pode resultar em diferentes resultados para cada simulação, mostrou que o kernel polynomial foi o que apresentou o melhor desempenho em classificar a situação indicada. A seguir um gráfico de desempenho das simulações, na figura x.



Os resultados obtidos de *Analysis*, mostram que a correlação linear entre os valores dos dados das colunas geradas como modelos e os alvos selecionados alcançou 0,996 de correlação linear, sendo que este valor varia entre -1 e 1, mostrando neste caso excelente índice de correlação de entre resultados previstos e os alvos(*targets*) indicados. Em termos de correlação linear foi obtido nesse caso 99,8% de correlação, utilizando o kernel polynomial. As descrições de valores dos erros representam os limites do dimensionamento do hiperplano de margem máxima de separação das classes.

Sendo desta forma assertivo a confirmação de que o modelo gerado, caracteriza a situação a que o motor o motor foi ensaio, encontrando o padrão da condição de normalidade do motor de indução.

Através da utilização do no *table,* é possível visualizar os valores gerados que compõe o modelo preditivo. (anexo)5

### 5.3.2 *Dataset* 2 – Condição de Desbalanceamento

Para avaliar o desempenho de MVS sobre o banco de dados de desbalanceamento, foi selecionado como alvo(*target*) a coluna “i Desb 4”, caracterizando o fluxo da corrente alternada em uma das fases, e outras sete colunas de representação de desbalanceamento foram selecionadas como *input* para serem comparadas com *target*, através de *Analaysis*, obtendo os seguintes resultados:

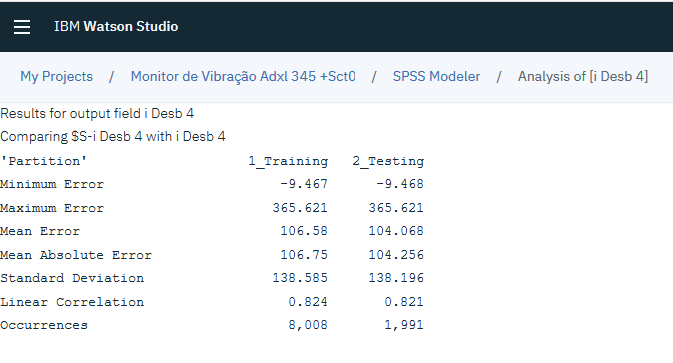
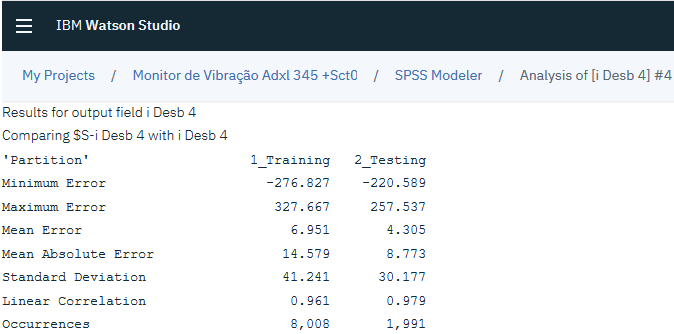
 

Figura 10 - kernel rbf Figura 11 - kernel polynomial

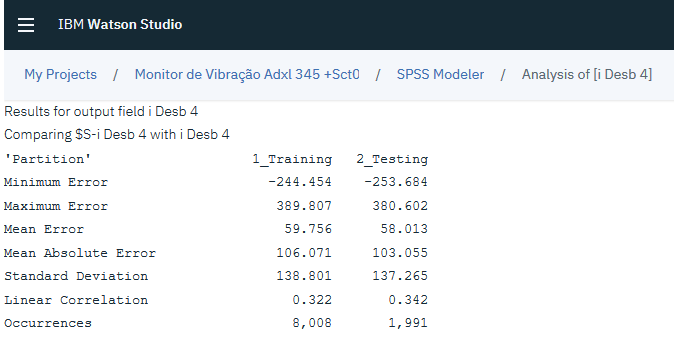
 

Figura 12 kernel sigmoid Figura 13- kernel linear

Os resultados visualizados em *Analysis*, retirados das simulações com cada função kernel, apresentam o desempenho do classificador MVS nos campos de treino e teste do modelo gerado para a classificação da condição especificada do motor de indução. Mostrando as tendências de correlação entre resultados previstos e o alvo(*target*) indicado. Tem-se nesse caso como desempenho mais eficiente de classificação utilizando o kernel polynomial, com 0.xxx. As descrições de valores dos erros representam os limites do dimensionamento do hiperplano de margem máxima de separação das classes.

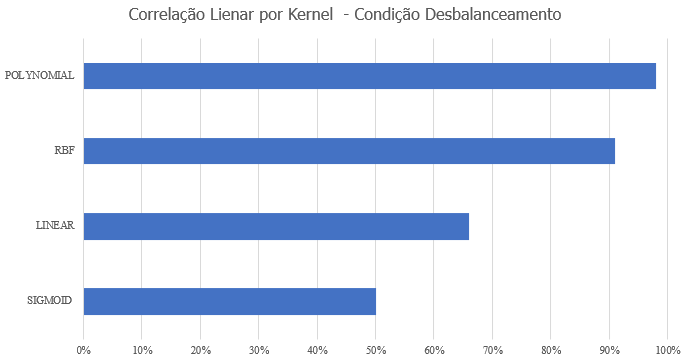
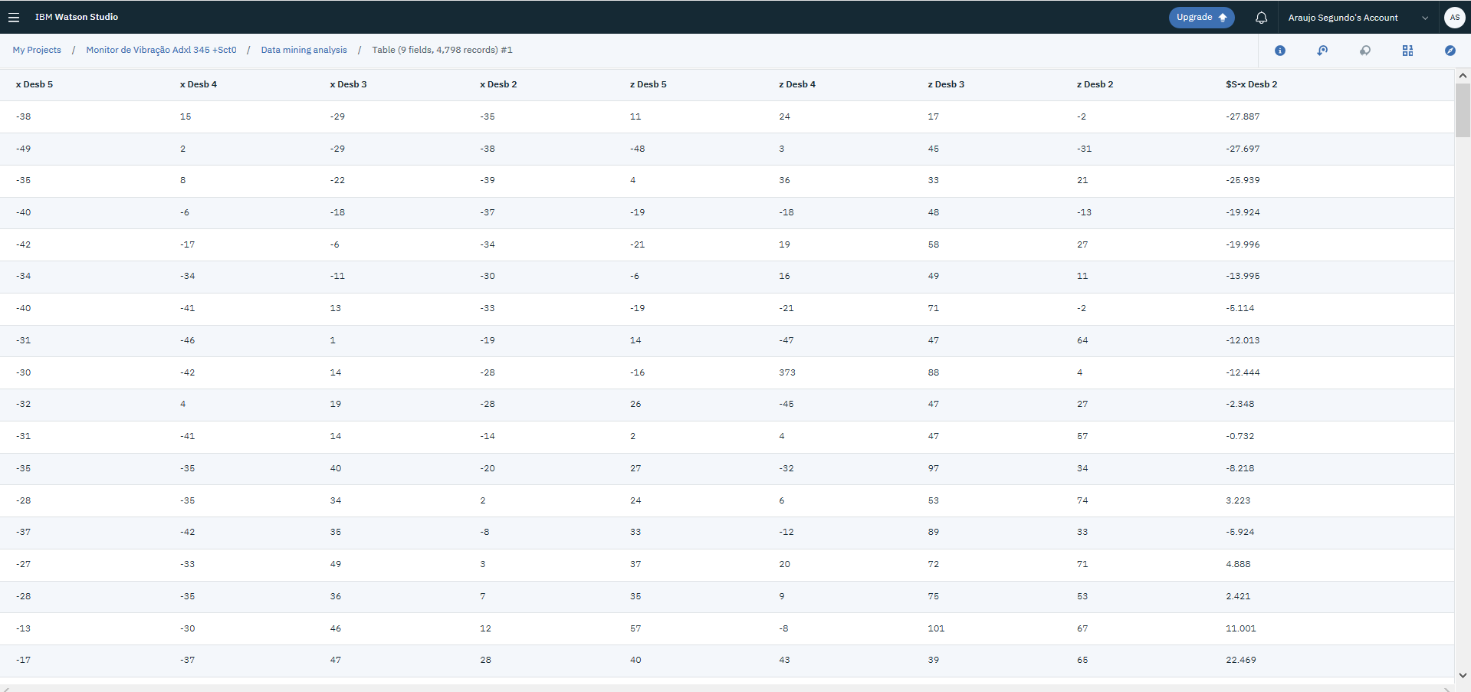


Figura 14 - correlação desbalanceamento verificar polynomial

Sendo satisfatório a acurácia de 93,9% entre os valores do modelo gerado “$ i Desb4” comparado com “i Desb4”, encontrada na utilização do kernel polynomial. A acurácia estima a capacidade de o classificador reconhecer corretamente uma amostra, caracterizando a situação a que o motor o motor foi ensaio, encontrando o padrão da condição de desbalanceamento.

Através da utilização do no *table,* é possível visualizar os valores gerados que compõe o modelo preditivo, na coluna “$ i Desb4”.



Satisfazendo um dos objetivos deste trabalho, classificando os valores de entrada para possa gerar um modelo preditivo de uma característica através dos dados de treinamento fornecidos, e testando o modelo com 20% dos valores do banco de dados fornecido. Sendo possível ratificar a eficácia de MVS para encontrar padrões.

### 5.3.3 *Dataset* 3 - Condições de alimentação por duas fases

Para esse *dataset* foram selecionadas as colunas “y 2fases 5” como alvo e outras sete colunas como inputs da condição de alimentação por somente duas fases para as simulações de MVS. A seguir os resultados obtidos através de *Analysis.*

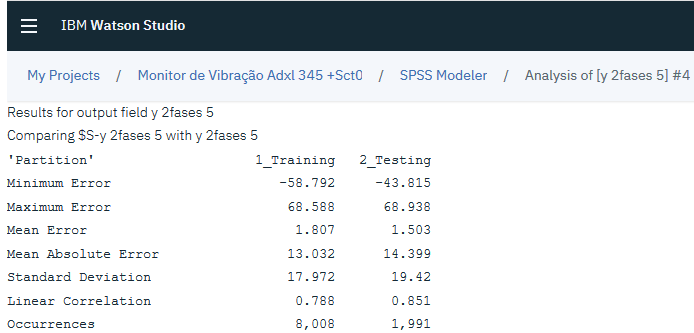
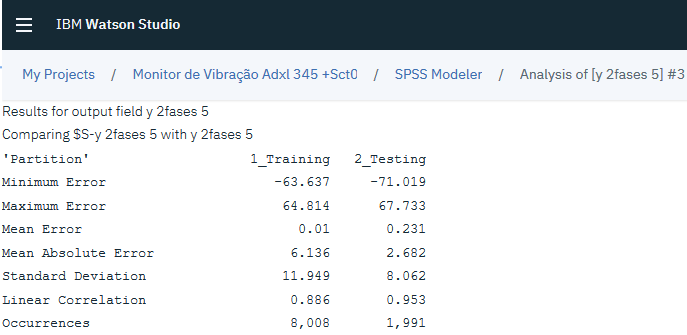
 ****

Figura 15 - kernel rbf Figura 16 - kernel polynomial

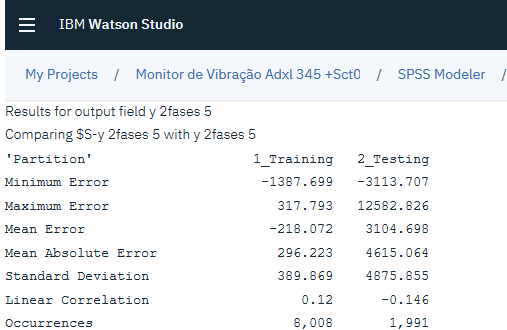
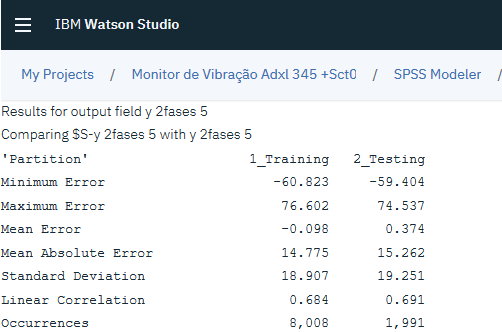
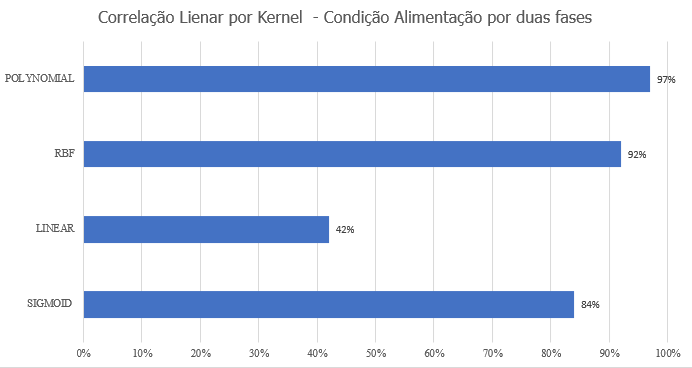
 

Figura 17 - kernel linear Figura 18 - kernel sigmoid

Os resultados obtidos de *Analysis*, mostraram que o kernel polynomial apresentou melhor performance de treino, onde foi obtido 0.953, mostrando a tendência de correlação entre resultados previstos e o alvo(*target*) indicado. As descrições de valores dos erros representam os limites do dimensionamento do hiperplano de margem máxima de separação das classes. A seguir na figura x, o gráfico das representações das correlações obtidas para os kernels utilizados.



Portanto, a capacidade de o classificador reconhecer corretamente a amostra selecionada é satisfatória, sendo polynomial o kernel que obteve melhor performance. Definindo a localização da característica do teste físico de funcionamento do motor com alimentação de somente duas fases.

### 5.3.4 *Dataset* 4 - Condições de Desnível na base

Para avaliar o reconhecimento de uma classe das condições de desnível ensaiadas no motor de indução foram selecionadas como alvo(*target*) a coluna “i Desnível 3” e outras sete colunas de representação da condição de desnível foram selecionadas como entradas(*inputs)*, para a execução de MVS. Através de *Analaysis*, foram obtidos os seguintes resultados. (figura y)

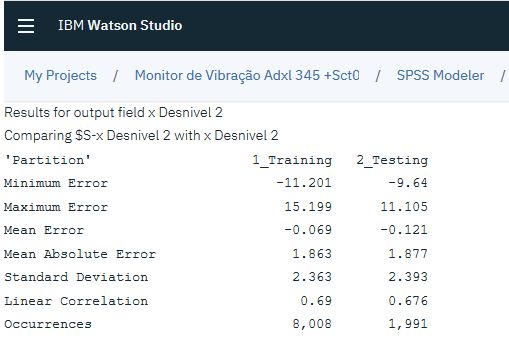
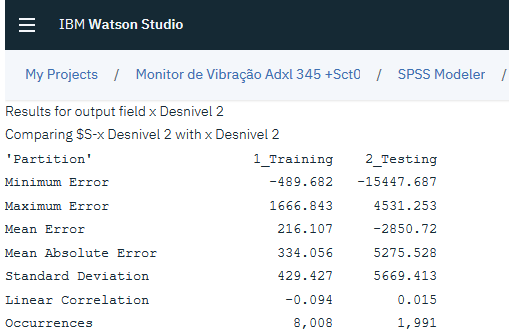
 

Figura 19 - kernel linear Figura 20 - kernel sigmoid

Os resultados obtidos de *Analysis*, mostram que foi obtida correlação linear de 0,894, mostrando a tendência de correlação positiva entre resultados previstos e o alvo(*target*). Indicando que foi obtido, neste caso, 90,85% de correlação. As descrições de valores dos erros representam os limites do dimensionamento do hiperplano de margem máxima de separação das classes.

Notando-se que a aplicação de SVM conseguiu classificar com alta correlação entre os valores reais e os valores previsto para a característica em questão, sendo novamente assertivo em classificar a classe que representa uma condição do motor de indução, neste caso a de desnível na base.

## **5.4** **Discussão dos resultados obtidos no estudo do modelo físico do sistema de monitoramento de vibração e corrente aplicado em IBM® Cloud**

Na sessão anterior foram apresentados os resultados obtidos para análises de desempenho de MVS para auxiliar na decisão de qual conjunto de funções executou de maneira mais assertiva a criação de modelos com base em características de diferentes classes em diferentes bancos de dados simulados.

Nota-se que a oferta da escolha das funções de kernel condiciona os dados a diferentes performances de processamento. Podendo ser possível que sejam necessários vários testes empíricos para que se possa obter a melhor performance das funções, onde a característica de generalização do modelo deve ser mantida, para que desta forma MVS classifique modelos entre as classes avaliadas.

Neste estudo de caso, foram utilizados os valores dos parâmetros de kernel em modo *default*, sendo os seguintes para cada simulação:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | RBF | POLYNOMIAL | SIGMOID | LINEAR |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Desta forma foi possível observar o desempenho de classificação proposto por MSV conseguiu com alto nível de correlação obter modelos preditivos acerca das características selecionadas, mostrando que o kernel polynomial obteve desempenho superior, como mostrado na tabela x.

|  |  |  |
| --- | --- | --- |
| Condição | Correlação Linear | Kernel |
| Normal | 98% | Polynomial |
| Desbalanceamento | 93,9% | Polynomial |
| Duas fases | 94,5% | Polynomial |
| Desnível na base | 90,85% | Polynomial |

# **6 Conclusão e Trabalhos Futuros**

### 6.1 Conclusões

Nessa dissertação foi proposta uma metodologia para classificar quatro condições diferentes de funcionamento de um motor de indução trifásico, onde o objetivo foi cumprido através da utilização de MVS que classificou diferentes padrões do motor do motor de indução.

O desenvolvimento teve início com a rotina implementada para capturar os dados de vibração e corrente do motor, através do ESP8266, gerando bancos de dados que foram importados na infraestrutura da IBM Cloud, para utilização da ferramenta de mineração de dados SPSS Modeler, onde foi aplicado o MVS.

As aplicações mostraram que a seleção distinta dos tipos e parâmetros das funções *kernel* aplicadas aos exemplos produzem diferentes valores de acurácia dos modelos gerados, sendo polynomial o que atende as necessidades de uma classificação com maior grau de confiança entre os kernels testados. Esta metodologia pode ser bastante útil em situações onde o pesquisador não conhece a natureza dos dados a serem classificados, sendo obrigado a empregar métodos empíricos para configurar os o classificador MVS.

Mostrando, junto as fundamentações de [][][] et al, que SVM tem a capacidade de melhorar o desempenho na realização de alguma tarefa por meio da experiência. Onde os algoritmos de MVS mostraram que capacidade de aprendizado a partir de um conjunto de dados de treinamento, onde ele busca por uma hipótese(alvo), no espaço de possíveis hipóteses, sendo capaz de descrever as relações entre os objetos e gerar um modelo que melhor se ajuste aos dados de treinamento fornecidos.

A identificação das condições – Normal, Desbalanceamento, Alimentação por duas fases e Desníveis na base -, formam a base para implementações onde a classificação de MVS ..

Em suma, destaca-se que o desenvolvimento experimental de técnicas de inteligência artificial, podem ser combinadas com as tradicionais técnicas de manutenção ,podendo ser a saída mais viável para uma manutenção industrial eficiente e eficaz no contexto da indústria 4.0.

### 6.2 Trabalhos Futuros

A partir desse ponto de avaliação e confirmação que a ferramenta aplicada pode traz boas acurácia em classificar as condições do motor de indução, torna-se viável estruturar uma proposta de aplicação de avaliação da condição do motor em tempo real, trazendo gatilhos que possam identificar, diagnósticas e disparar um alarme ou desarme do motor, desta forma evitando situações de colapsos que poderiam causar mais danos, tanto inerentes a disponibilidade da máquina quanto à produtividade de um sistema que dependa do motor de indução.

Algumas aplicações futuras podem ser realizadas, dentre elas :

* A
* A
* A
* A

# **7 REFERÊNCIAS**

Vapnik, V. (1995). **The Nature of Statistical Learning Theory**. New York, NY: Springer-Verlag.

[99] Lorena Teixeira Marques**. Análise Computacional Do Motor De Indução Trifásico: Regime Transitório E Permanente.** UNIVERSIDADE DE SÃO PAULO,São Carlos 2009

[88]Vinícius Augusto Diniz Silva. **Detecção de falhas em motores elétricos**

**através das máquinas de vetores de suporte**. UNIVERSIDADE ESTADUAL DE CAMPINAS, Campinas, 2012.

[40] James Surowiecki, **The Wisdom of Crowds, why the many are smarter than the few and how collective wisdom shapes business, economies, societies and nations**, Anchor Books, 2004, pp 66-83.

[47]Daniel Larose and Chantal Larose, **Discovering Knowledge in Data: An Introduction to Data Mining**, John Wiley & Sons, 2014.

[48] Jiawei Han, Micheline Kamber and Jian Pei, **Data Mining Concepts and Techniques 3rd Edition**, Elsevier Inc., 2012, pp 23

[49] Ian H. Witten and Eibe Frank, **Data Mining Practical Machine Learning Tools and Techniques**, 3rd Edition, Elsevier Inc. 2011.

[50] Tom M. Mitchell, **Machine Learning**, McGraw-Hill Science, 1997, pp 2

[51] Kevin P. Murphy, **Machine Learning, A Probabilistic Perspective**, Kevin P. Murphy, The MIT Press, 2012.

[52] Christopher M. Bishop, **Pattern Recognition and Machine Learning**, Springer, 2006.

[53] Noam Nisan and Shimon Schocken, **The Element of Computing Systems Building a Modern Computer from First Principles**, MIT press 2005, pp 57-58.

[54] Xin Liu et al, **Computational Trust Models and Machine Learning**, CRC Press, 2015.

[55] Peter Harrington, **Machine learning in Action**, Manning Publications, 2012.

[56] Sameer Wadkar, Madhu Siddalingaiah, **Pro Apache Hadoop**, 2nd Edition, Apress, 2014.

[74]. Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi, **Mastering Cloud Computing**, Morgan Kaufmann, USA, May 2013.

[76] Caesar Wu, Rajkumar Buyya, **Cloud Data Centers and Cost Modeling**, Morgan Kaufmann, 2015.

[797] Rajkumar Buyya e Kotagiri Ramamohanarao ,**Big Data Analytics = Machine Learning + Cloud Computing** , University of Melbourne January 2016

[9999]<https://introduceti.com.br/blog/iaas-paas-saas-conheca-os-modelos-fundamentais-de-cloud-computing/>

[4] Han, J. and Kamber, M.( 2006). **Data Mining: Concepts and Techniques**. Morgan Kaufmann Publishers, pp. 1-7.

[427]Lima, C. A. M. Comitê de Máquinas: **Uma Abordagem Unificada Empregando Máquinas de Vetores-Suporte**, Tese de Doutorado, FEEC/Unicamp, 2004.

[555].IBM KNOWLEDGE CENTER <https://www.ibm.com/support/knowledgecenter/en>

[565]REDBOOK IBM, **Cloud Object Storage as a Service IBM Cloud Object Storage from Theory to Practice**. March 2017, International Technical Support Organization, Copyright International Business Machines Corporation 2017. All rights reserved

[902]Han, J.; Kamber, M. **Data Mining: Concepts and Techniques**. 2nd ediEon, Morgan Kaufmann, 2006.

[2222] Ana Carolina Lorena e André F. de Carvalho .**Uma Introdução às Support Vector Machines** ufrgs 2007

[878] Eulanda Miranda dos Santos, **Teoria e Aplicação de Support Vector Machines à Aprendizagem e Reconhecimento de Objetos Baseado na Aparência**, Universidade Federal da Paraíba 2002

[449] [Müller et al., 2001] Müller, K.-R., Mika, S., et al. (2001). **An introduction to kernel-based learning algorithms.** IEEE Transactions on Neural Networks, 12(2):181–201.

[24] V. Vapnik, **Statistical Learning Theory**, John Wiley and Sons, **(1998)**.

[25] E. Sami, “**Support Vector Machines for classification and locating faults on transmission lines**”, Applied Soft Computing, vol. 12, **(2012)**, pp. 1650–1658.

[26] T. Joachims, “**Learning to classify text using support vector machines**”, Kluwer Academic Publishers, **(2002)**.

[172] CHRISTOPHER J.C. BURGES, **A Tutorial on Support Vector Machines for Pattern Recognition** 1998

Elmore, W .A . (2004). **Protective Relaying Theory and Applications**. New York, NY: Marcel Dekker, Inc.

IEEE Operation Center (1997). **Advancement in Microprocessor Based Protection and**

**Communication**, IEEE Tutorial Course Text, 97-TP-120-0, Piscataway, NJ.

Kezunovic, M. (1997). **A survey of neural network application to protective relaying and fault analysis. Engineering Intelligent Systems**, vol. 5, No. 4, 185-192.

Kolla, S., & Altman, S. (2007). **Artificial neural network based fault identification scheme implementation for a three-phase induction motor**. ISA Transactions, vol. 46, No.2, 261- 266.

AVELAR, V. S.; BACCARINI, L. M. R.; AMARAL, G. F. V. **Desenvolvimento de um sistema inteligente para diagnostico de falhas nos enrolamentos do estator de motores de indução**. X SABI–Simpósio Brasileiro de Automação Inteligente. São Joao Del Rei–MG–Brasil, 2011.

Widodo, A., & B-S.Yang. (2007). **Support vector machine in machine condition and fault diagnosis. Mechanical System and Signal Processing**, vol.21, 2560-2574.

Brun, Y., & Ernst, M. D. (2004). **Finding latent code errors via machine learning over program executions**. IEEE Computer Society.

Sudha, M., & Anbalagan, P. (2009). **A Protection Scheme for Three-Phase Induction Motor from Incipient Faults Using Embedded Controller**. Asian Journal of Scientific Research , 28-50.

[52] IBM Systems and Technology, **Watson – Um Sistema Projetado para Respostas O futuro do design de sistemas otimizados para carga de trabalho**. 2011

[1] Gilles Blanchard,1 Olivier Bousquet and Pascal Massart. **STATISTICAL PERFORMANCE OF SUPPORT VECTOR MACHINES**. Institute of Mathematical Statistics, 2008

[876] DIEGO BONESSO **Estimação dos Parâmetros do Kernel em um Classificador SVM na Classificação de Imagens Hiperespectrais em uma Abordagem Multiclasse**. UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

[92]SAMADZADEGAN, F.; HASANI, H.; SHENK, T. **Simultaneous feature selection and SVM parameter determination in classification of hyperspectral imagery using Ant Colony Optimization**. , [S.l.], p.139–156, 2012.

# **8 APÊNDICES**

APENDICE 0 PROCESSING

(várias fontes , editar)

Processing é uma biblioteca gráfica de [código aberto](https://en.wikipedia.org/wiki/Open-source_software) e [um ambiente de desenvolvimento integrado](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) criado para comunidades de artes eletrônicas, [novas mídias](https://en.wikipedia.org/wiki/New_media_art) e [design visual](https://en.wikipedia.org/wiki/Visual_design), com o objetivo de ensinar a não programadores os fundamentos da [programação](https://en.wikipedia.org/wiki/Computer_programming) de [computadores](https://en.wikipedia.org/wiki/Computer_programming) em um contexto visual.

O processamento usa a [linguagem Java](https://en.wikipedia.org/wiki/Java_(programming_language)) , com simplificações adicionais, como classes adicionais e funções e operações matemáticas aliadas. Além disso, ele também possui uma interface gráfica do usuário para simplificar o estágio de compilação e execução.

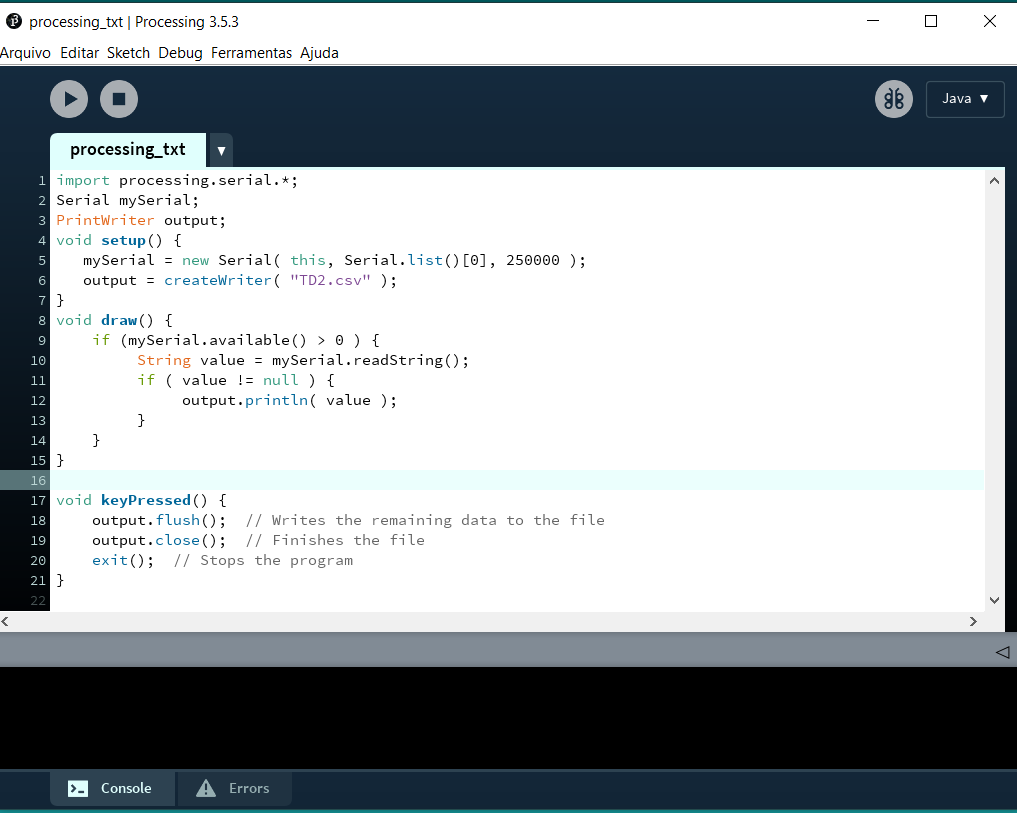
A linguagem Processing e o IDE foram os precursores de outros projetos, incluindo [Arduino](https://en.wikipedia.org/wiki/Arduino) , [Wiring](https://en.wikipedia.org/wiki/Wiring_(development_platform)) e p5.js

Além disso, a ideia da comunicação entre Processing e Arduino é basicamente assim: o Processing pode ser utilizado de duas formas, fazendo o Arduino executar determinada tarefa ou atuando como uma representação gráfica e visual daquilo que está acontecendo no seu protótipo em protoboard.

Quando o Processing atua como um agente que vai “comandar” o Arduino, então temos que fazer com que o Arduino apenas realize a leitura da porta serial (onde é feita a comunicação entre eles).

Caso o Arduino receba algum valor, enviado pelo processing, ele realiza determinada função, ou seja, ele só fará uma ação X, ligar um led, por exemplo, se, o Processing mandar o valor correto, caso contrário isso não ocorrerá.

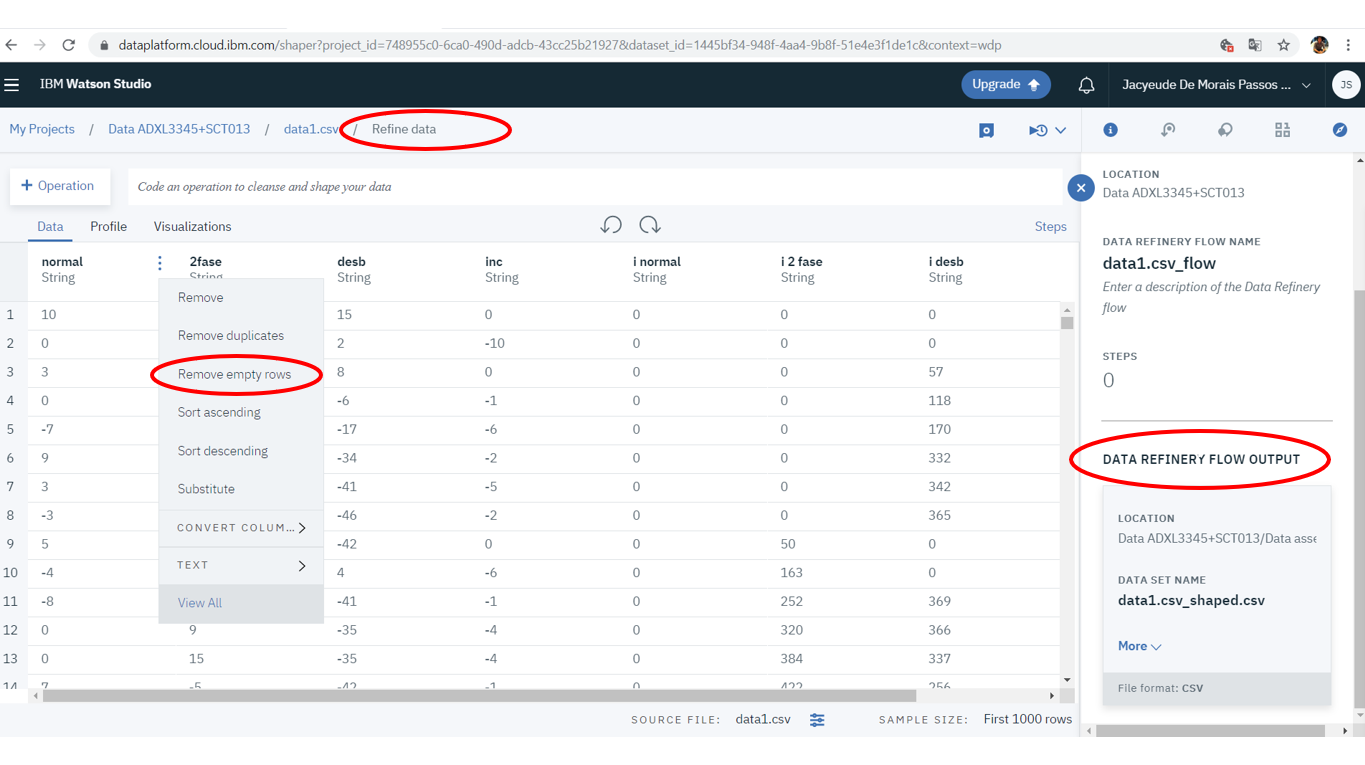
Agora quando o processing atua como uma representação visual daquilo que está ocorrendo na protoboard, temos basicamente o contrário. Agora é o Arduino que vai enviar pela serial o que está acontecendo na protoboard e a partir desse dado enviado, uma imagem (ou uma janela de cores) é mostrada no processing



APÊNDICES 95

**Refinaria de Dados**

Através da Refinaria de Dados(Data Refinery) é possível limpar e modelar dados tabulares com um editor de fluxo gráfico, havendo possibilidade também de usar operações, funções e operadores lógicos da biblioteca *dplyr R(..)*, para executar modificações iniciais inerentes a organização e disposição dos dados importados. Ao limpar os dados*(figura XZ)*, é possível corrigir ou remove dados incorretos, incompletos, formatados ou duplicados incorretamente. Podendo personalizar os dados, filtrando, classificando, combinando ou removendo colunas e executando operações. Na figura XZ é possível notar a seleção da opção “Remove empty rows”, para remover os espaços vazios da coluna selecionada.



É possível criar um fluxo da refinaria de dados como um conjunto de operações ordenadas, para ser realizado sob demanda quando houverem novas importações de dados. A Refinaria inclui uma interface gráfica para criar um perfil de seus dados, para validá-los e possui opção de mais de 20 gráficos personalizáveis que oferecem perspectiva e insights sobre databases importadas.

6.4.1 Especificação do formato de dados

Há de ser notado que há uma limitação quanto a especificidade de formatos que podem ser importados, sendo eles:

- Avro, CSV, JSON, Parquet ou dados de texto

- Tabelas em fontes de dados relacionais

A Refinaria de dados opera em um subconjunto de linhas de amostra no conjunto de dados, realizando importação de arquivos de grande tamanho e volume de dados. No entanto, quando é executada uma tarefa de fluxo da refinaria de dados, todo o conjunto de dados do arquivo é processado. De acordo com IBM Cloud Knowledge Center.

6.4.2 Validação de dados

Após adicionados ao Data Refinery, é possível validar os dados importados. Normalmente, isso deve ser feito em vários pontos do processo de refinamento.

Para validar os dados, o IBM Cloud Knowledge Center, da os seguintes passos:

1. Acesso na guia perfil.
2. Revisão de métricas para cada coluna.
3. Ações apropriadas, frente a frequência e estatística dos dados.
4. Salvar o fluxo gerado na Refinaria de Dados.

Sobre frequência e distribuição estatística, tem-se no IBM Cloud Knowledge que:

* Frequência é o número de vezes que um valor ou um valor em um intervalo especificado, ocorre. Cada distribuição de frequência mostra a contagem de valores exclusivos em uma coluna. Revisar a distribuição de frequência para encontrar anomalias nos dados é preciso.

Nas colunas, é possível personalizar o número de posições (agrupamentos) que é desejado para visualizar.

* Distribuição Estatísticas são uma coleção de dados quantitativos. As estatísticas para cada coluna mostram o mínimo, o máximo, a média e o número de valores exclusivos nessa coluna.

Dependendo do tipo de dados de uma coluna, as estatísticas para cada coluna podem variar ligeiramente. Por exemplo, as estatísticas para uma coluna do tipo *Integer*(inteiro) de dados têm valores mínimo, máximo e médio, enquanto as estatísticas para uma coluna do tipo de dados *String (corda)*possuem valores de comprimento mínimo, comprimento máximo e comprimento médio.