Programa de Pós Graduação

Dissertação de Mestrado

APLICANDO MACHINE LEARNING PARA ANÁLISE DO PERFIL DO COMPORTAMENTO NO APOIO A DECISÃO DO DEPARTAMENTO DE ASSISTÊNCIA ESTUDANTIL EM CURSOS DO ENSINO MÉDIO NO IFRS - CAMPUS SERTÃO

CEDEMIR PEREIRA

Pós-Graduação em Computação Aplicada

Computação Aplicada

Dissertação de Mestrado

APLICANDO MACHINE LEARNING PARA ANÁLISE DO PERFIL DO COMPORTAMENTO NO APOIO A DECISÃO DO DEPARTAMENTO DE ASSISTÊNCIA ESTUDANTIL EM CURSOS DO ENSINO MÉDIO NO IFRS - CAMPUS SERTÃO

CEDEMIR PEREIRA

**UNIVERSIDADE DE PASSO FUNDO**

**INSTITUTO DE CIÊNCIAS EXATAS E GEOCIÊNCIAS**

**PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA**

**APLICANDO MACHINE LEARNING PARA ANÁLISE NO PERFIL DO COMPORTAMENTO NO APOIO A DECISÃO DO DEPARTAMENTO DE ASSISTÊNCIA ESTUDANTIL EM CURSOS DO ENSINO MÉDIO NO IFRS - CAMPUS SERTÃO**

**Cedemir Pereira**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Computação Aplicada na Universidade de Passo Fundo.

**Orientador: Roberto Rabello**

Passo Fundo

2022

**APLICANDO MACHINE LEARNING PARA ANÁLISE DO PERFIL DE COMPORTAMENTO NO APOIO A DECISÃO DO DEPARTAMENTO DE ASSISTÊNCIA ESTUDANTIL EM CURSOS DO ENSINO MÉDIO NO IFRS - CAMPUS SERTÃO**

**RESUMO**

Este trabalho apresenta o desenvolvimento de uma ferramenta de apoio ao estudante chamada SYSDAE para o Departamento de Assistência Estudantil do IFRS - Campus Sertão. O campus dispõe para os alunos o Regime de Residência Estudantil onde os alunos que moram em municípios atendidos(que no total são cerca de 153, abrangendo 13 estados do Brasil) podem permanecer como residentes. No dia-a-dia da Residência acontecem muitas ocorrências que são registradas em Atas com ciência e assinatura dos alunos. Essas Atas, posteriormente, serão julgadas por uma Comissão Disciplinar, e, dependendo do caso, serão tomadas as medidas cabíveis. O sistema a ser desenvolvido além de registrar o controle das ocorrências dos alunos, pretende através do uso de Machine Learning e da análise dos dados armazenados, predizer padrões de comportamento para saná-los antes que ocorra o pior, que é a perda da residência estudantil*.*

Palavras-chave::Assistência Estudantil. Sistemas Preditivos. Machine Learning

**APPLYING MACHINE LEARNING FOR THE ANALYSIS OF THE BEHAVIOR PROFILE IN SUPPORTING THE DECISION SUPPORT OF THE DEPARTMENT OF STUDENT ASSISTANCE IN HIGH SCHOOL COURSES IN IFRS - CAMPUS SERTÃO**

**abstract**

This work presents the development of a student support tool called SYSDAE for the Student Assistance Department of IFRS - Campus Sertão. The campus offers students the Student Residence Regime where students who live in municipalities served (which in total are about 153, covering 13 states in Brazil) can remain as residents. In the day-to-day life of the Residence, there are many occurrences that are recorded in Minutes with the students' acknowledgment and signature. These Minutes will subsequently be judged by a Disciplinary Committee, and, depending on the case, the appropriate measures will be taken. The system to be developed, in addition to recording the control of students' occurrences, intends, through the use of Machine Learning and the analysis of stored data, to predict behavior patterns to remedy them before the worst occurs, which is the loss of the student residence .

Keywords: Student Assistance. Machine Learning. Predictive System

**Lista de Siglas**

COAGRI- COORDENAÇÃO NACIONAL DE ENSINO AGRÍCOLA

DAE - DEPARTAMENTO DE ASSISTÊNCIA ESTUDANTIL

IFRS - INSTITUTO FEDERAL DO RIO GRANDE DO SUL

PNAES - PROGRAMA NACIONAL DE ASSISTÊNCIA ESTUDANTIL

PROEJA - PROGRAMA NACIONAL DE INTEGRAÇÃO DA EDUCAÇÃO PROFISSIONAL COM A EDUCAÇÃO BÁSICA NA MODALIDADE DE EDUCAÇÃO DE JOVENS E ADULTOS

RS - REVISÃO SISTEMÁTICA

TDAH - TRANSTORNO DE DÉFICIT DE ATENÇÃO E HIPERATIVIDADE

**Sumário**

**1 - Introdução**

O Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS) é uma instituição federal de ensino público e gratuito que se propõe a fornecer ensino humanizado, crítico e cidadão. Oferece cursos gratuitos em 16 municípios gaúchos. São cursos de nível médio (técnicos que podem ser cursados de forma integrada, concomitante e subsequente ao Ensino Médio), superiores (de graduação e pós-graduação) e de extensão.

No total, são aproximadamente 22.200 alunos e 217 opções de cursos. Tem aproximadamente 1.192 professores e 918 técnicos-administrativos.

O Campus Sertão foi criado em 19 de julho de 1957, com a denominação de Escola Agrícola de Passo Fundo, o campus iniciou seu efetivo funcionamento no ano de 1963.O Campus Sertão do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul está situado no Distrito de Engenheiro Luiz Englert, município de Sertão, a 25 quilômetros de Passo Fundo, região Norte do Estado do Rio Grande do Sul e integra a Rede Federal de Educação Profissional, Científica e Tecnológica.

O campus tem autonomia para ministrar Curso de Educação Básica em Nível de Ensino Médio(com 495 alunos) e Nível Superior(435 alunos) e pós-graduação(49 alunos). O Campus conta com cerca de 200 servidores, conta com um Corpo docente de Mestres e Doutores; Corpo técnico-administrativo de pós-graduados e mestres; e abrange cerca de 153 municípios abrangidos (13 Estados do Brasil).

No IFRS Campus Sertão há diversos setores, entre eles, o Departamento de Assistência Estudantil(DAE) que atende desde alunos com dificuldades pedagógicas, passando também por alunos com dificuldades psicológicas específicas - como transtornos de ansiedade, TDAH (Transtorno de Déficit de Atenção e Hiperatividade), transtorno depressivo, entre outros. Também trabalha com as habilidades necessárias para convívio coletivo, especialmente na Residência Estudantil, de acordo com o Regulamento de conduta para estudantes residentes e semirresidentes.

Atualmente o IFRS não possui software disponível para os servidores que atuam no Departamento de Assistência Estudantil, de forma que há uma notável omissão do Estado no tocante a esta realidade. Essas dificuldades resultam em obstáculos ao atendimento que é desejável aos alunos, pais e aos próprios servidores.

Neste projeto, propõe-se o desenvolvimento de um software que proporcione corrigir as lacunas existentes na ausência de um sistema informatizado, de modo a auxiliar os servidores do DAE com um sistema no qual seja possível centralizar as informações em um único local, propiciando uma resposta eficiente aos operadores do sistema.

Este trabalho, contemplará desde uma pesquisa preliminar comparativa das características dos principais sistemas informatizados para assistência estudantil e acompanhamento acadêmico em uso nos Institutos Federais (IFs), até a concepção e desenvolvimento de um sistema próprio, o SYSDAE , desenvolvido em arquitetura cliente/servidor para uso em ambiente web e utilização voltada primariamente para estações de trabalho (desktops).

Os Desafios da Administração do DAE para proporcionar um atendimento melhor para os alunos, são a falta de recursos humanos, a logística (conseguir integrar todos os setores que estão em lugares diferentes) problemas estruturais ( os alojamentos estão em prédios antigos) e falta de ferramenta de apoio ao estudante.

**2 - Problema de pesquisa**

O Instituto Federal do Rio Grande do Sul(IFRS) - Campus Sertão oferta residência aos alunos que residem em cidades distantes. A residência é o setor responsável pelo controle desses alunos, que optam ou necessitam se instalar na área interna do campus, promovendo a segurança e o bem-estar, além de tranquilidade aos pais ou responsável. Na Coordenadoria de Residência Estudantil são armazenadas e gerenciadas as informações dos alunos. Atualmente este controle é feito manualmente, e todos os formulários de registros são criados em programas de edição de texto, e a cada novo registro é refeito o trabalho de digitar os documentos, com novas informações.

Quando um interno descumpre qualquer regra estabelecida pelo Regulamento de Conduta, o coordenador pode adverti-lo verbalmente ou, se julgar necessário, formalmente através do registro de um documento chamado “Atas de Ocorrências da Residência Estudantil”. Nesse documento tem-se os dados escolares do interno, o número do seu quarto, a descrição da infração cometida com maiores detalhes possíveis, a referência no Regulamento que dará respaldo a advertência e o nome do servidor que registrou o Ato de Indisciplina, e, ainda, os campos para assinatura do servidor e dos alunos responsáveis pela desobediência.

No Conselho disciplinar são julgadas todas as ocorrências dos alunos e se, o aluno for reincidente, este corre o risco de perder o uso da residência estudantil.

Dessa forma o nosso problema de pesquisa “é possível através do desenvolvimento de um sistema que utilize Machine Learning, prever a partir do padrão de comportamento de alunos o risco de perder a residência estudantil?”

**3. Justificativa**

A equipe do Departamento de Assistência Estudantil (DAE) do Instituto Federal do Rio Grande do Sul (IFRS) – campus Sertão, identificou, através de sua atuação junto aos alunos da instituição, a necessidade de que um sistema informatizado possa acolher seus processos de assistência estudantil. O referido setor solicitou então auxílio para pesquisa, desenvolvimento e implantação de um software que atenda suas necessidades operacionais diárias.

Todos os processos da DAE são operacionalizados com apoio de planilhas eletrônicas ou documentos de texto. Segundo experiência da equipe, o uso de tais ferramentas mostra-se pouco produtivo para controlar, organizar, sistematizar, catalogar, e acessar informações de forma eficaz, levando inevitavelmente a operações repetitivas, informações conflitantes ou inconsistentes. Face a uma equipe reduzida e demanda crescente, fica evidente a necessidade de apoio de um software que atenda processos em três frentes: na área de assistência estudantil, atendimento personalizado aos alunos e nas atividades dos alunos da residência estudantil, além da área do registro de atendimentos psicossociais.

Apesar de oferecer estes e outros serviços para uma comunidade de cerca de 1291 alunos, e estender o atendimento do Restaurante Universitário a aproximadamente 190 servidores, o DAE não contava, até o ano de 2022, com um software para auxiliar os servidores na condução e registro das atividades de ocorrências dos alunos. Toda a gestão documental do setor era feita em formulários de papel, documentos de texto e planilhas eletrônicas, dificultando uma busca rápida dos atendimentos referentes aos apontamentos dos alunos.

Tarefas preditivas são tarefas voltadas para previsão, que é encontrar uma função, modelo ou hipótese que pode ser utilizada para previsão, digamos que eu posso querer prever um valor de um imóvel, ou por exemplo eu quero prever o estado para um novo paciente daqui a 5 meses depois da cirurgia, se ele estará doente ou saudável como nos vimos anteriormente, então na previsão eu tenho uma entrada (geralmente representada por X) e uma saída (geralmente representada por Y)

O SysDAE e Machine Learning serão usados para predizer alguns comportamentos de alunos. Dessa forma saberemos qual percentual de possibilidade um aluno tem, por exemplo de perder a residência estudantil, baseado num comportamento padrão dos alunos da escola.

**4 - Objetivos**

O objetivo geral deste trabalho é desenvolver um software de gestão para controle, integração e otimização dos processos do Departamento de Assistência Estudantil do IFRS - Campus Sertão e usar Machine Learning para predizer alguns resultados. Além disso, tem-se os seguintes objetivos específicos:

* Controlar os dados das Atas de Ocorrências;
* Usar Machine Learning para prever padrões de comportamento nas ocorrências de atendimento aos alunos;
* Avaliar os padrões levantados pelo comportamento dos alunos para gerar recomendações.

**5 - Fundamentação Teórica**

Recentemente, o mundo dos negócios passou por uma transição, de uma economia industrial para uma economia de informação, criando um ambiente no qual a informação passa a ser fonte de riquezas e prosperidade e as empresas habituadas a vencer pelo tamanho começam a perder terreno para concorrentes mais ágeis no uso da informação.

Neste contexto de mudanças, sejam elas culturais, políticas, sociais, econômicas ou tecnológicas, as empresas dotadas de uma visão mais holística compreenderam que é necessário gerenciar adequadamente as informações. O uso correto da informação torna-se necessário não só pelo fato de manter-se atualizado, mas principalmente pela possibilidade de permitir a identificação de oportunidades e ameaças para as organizações.

Apresenta-se a seguir os conceitos fundamentais diretamente relacionados com o tema abordado no presente trabalho, o qual envolve estudos sobre Machine Learning e sua contribuição para a melhoria de processos administrativos na gestão do DAE.

* 1. **5.1 - IFRS e DAE**

O Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS) foi criado em 29 de dezembro de 2008, pela Lei 11.892, que instituiu, no total, 38 Institutos Federais de Educação, Ciência e Tecnologia. Por força de lei, o IFRS é uma autarquia federal vinculada ao Ministério da Educação (MEC). Goza de prerrogativas com autonomia administrativa, patrimonial, financeira, didático-científica e disciplinar. Pertence à Rede Federal de Educação Profissional e Tecnológica.

Em sua criação, o IFRS se estruturou a partir da união de três autarquias federais: o Centro Federal de Educação Tecnológica (Cefet) de Bento Gonçalves, a Escola Agrotécnica Federal de Sertão e a Escola Técnica Federal de Canoas. Logo após, incorporaram-se ao instituto dois estabelecimentos vinculados a Universidades Federais: a Escola Técnica Federal da Universidade Federal do Rio Grande do Sul (Ufrgs) e o Colégio Técnico Industrial Prof. Mário Alquati, de Rio Grande. No decorrer do processo, foram federalizadas unidades de ensino técnico nos municípios de Farroupilha, Feliz e Ibirubá e criados os campi de Caxias do Sul, Erechim, Osório e Restinga. Estas instituições hoje fazem parte do IFRS na condição de campi.

O Campus Sertão do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul está situado no Distrito de Engenheiro Luiz Englert, município de Sertão, a 25 quilômetros de Passo Fundo, região Norte do Estado do Rio Grande do Sul e integra a Rede Federal de Educação Profissional, Científica e Tecnológica.

Criado pela Lei n° 3.215, de 19 de julho de 1957, com a denominação de Escola Agrícola de Passo Fundo, o campus iniciou seu efetivo funcionamento no ano de 1963. Através do Decreto Lei n° 53.558, de 13 de fevereiro de 1964, passou a denominar-se Ginásio Agrícola de Passo Fundo, com localização em Passo Fundo (RS), subordinado à Superintendência do Ensino Agrícola e Veterinária, ligada ao Ministério da Agricultura. Pelo Decreto n° 60.731, de 19 de maio de 1967 a instituição foi transferida, juntamente com outros órgãos de Ensino, para o Ministério da Educação e Cultura.

O Decreto n° 62.178, de 25 de janeiro de 1968, autorizou o Ginásio Agrícola de Passo Fundo a funcionar como Colégio Agrícola. A denominação Colégio Agrícola de Sertão, foi estabelecida pelo Decreto n° 62.519, de 09 de abril de 1968. A partir de então, ficou sob a coordenação da Coordenação Nacional de Ensino Agrícola – COAGRI – durante o período de 1973 até 1986.

Pelo Decreto n° 83.935, de 04 de setembro de 1979, passou a denominar-se Escola Agrotécnica Federal de Sertão, subordinada à Secretaria de Educação de 1° e 2° Graus do Ministério da Educação e Cultura. Obteve declaração da regularidade de estudos pela Portaria nº 081, de 06 de setembro de 1980, da Secretaria do Ensino de 1º e 2º Graus, do Ministério da Educação e Cultura. A Lei Federal n° 8.731, de 16 de novembro de 1993 transformou a Escola Agrotécnica Federal de Sertão em autarquia Federal, com autonomia administrativa e pedagógica.

A Lei nº 11.892, que cria os Institutos Federais de Educação, Ciência e Tecnologia no dia 29 de dezembro de 2008, transformou a antiga Escola Agrotécnica Federal de Sertão em Campus do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul.

Inicialmente, o campus oferecia o curso Ginasial Agrícola e conferia ao concluinte o diploma de Mestre Agrícola, de acordo com o Decreto-Lei nº 9.613, de 20 de agosto de 1946, da Lei Orgânica do Ensino Agrícola. No período de 1970 a 1975, oferecia o curso Técnico Agrícola e conferia ao concluinte o diploma de Técnico em Agricultura, em nível de 2º Grau. A partir do segundo semestre de 1973, a habilitação passou a titular-se Técnico em Agropecuária.

Hoje, o campus tem autonomia para ministrar Curso de Educação Básica em Nível de Ensino Médio e Formação Profissional com cursos de nível técnico a pós-graduação.

Integrado ao Plano de Expansão da educação profissional, desempenha função relevante na cooperação para o desenvolvimento socioeconômico regional, especialmente em regiões em que predominam as pequenas e médias propriedades rurais.

O campus tem marcante atuação junto à comunidade regional e desempenha papel importante no atendimento de demandas específicas na região, por meio dos cursos que desenvolve e das parcerias com municípios da região, empresas, cooperativas e outras instituições de ensino como Universidades e Sindicatos.

Contando com uma área de 237 hectares, além de modernos laboratórios, o campus mantém setores de ensino, pesquisa e produção nas áreas de: Agricultura (culturas anuais, fruticultura, silvicultura, olericultura e hidroponia), Zootecnia (avicultura de postura e de corte, cunicultura, apicultura, piscicultura, suinocultura, ovinocultura, bovinocultura de leite e de corte e equinocultura/equoterapia), Jardinagem, Agroindústria (abatedouro e processamento de carnes, leites e vegetais), Unidade de Beneficiamento de Sementes e Fabricação de Rações (recebimento, classificação, secagem, armazenamento de grãos e processamento em rações) e Mecanização Agrícola, constituindo um laboratório para prática profissional, atividades pedagógicas e produção de matéria-prima para o processo agroindustrial.

O campus funciona em período integral, com aulas teóricas e práticas, nos turnos da manhã, tarde e noite, incluindo, ainda, outras atividades para atendimento da clientela externa, como cursos de curta duração, que visam à atualização, capacitação e treinamento em áreas diversas e cursos de qualificação.

O IFRS tem disponível os seguintes cursos técnicos integrados a nível médio: manutenção e suporte em informática e agropecuária e o Técnico em Comércio do Proeja. Tem também o curso pós-médio em agropecuária.

O Campus tem ainda a nível superior os cursos de Agronomia, Zootecnia, Biologia, Agronegócio, Tecnologia em Análise e Desenvolvimento de Sistemas, Tecnologia em Agronegócio, Tecnologia em Gestão Ambiental e Formação Pedagógica para Graduados não Licenciados.

Na Pós Graduação tem os cursos de Lato Sensu em Teorias e Metodologias da Educação e o Lato Sensu em Desenvolvimento e Inovação

* 1. **5.2 - O Departamento de Assistência Estudantil**

O Departamento de Assistência Estudantil (DAE) é responsável por planejar e desenvolver a política de assistência estudantil do Campus Sertão, de acordo com as diretrizes do Programa Nacional de Assistência Estudantil - PNAES. Na prática, beneficiar os alunos envolve ações que promovam moradia, saúde, alimentação, acompanhamento, acolhimento, transporte e tantos outros itens importantes na vida dos estudantes. A assistência estudantil está dividida basicamente em ações universais que apoiam todos os estudantes matriculados, sem distinção, e o Programa de Benefícios. Iniciativas voltadas à equidade de oportunidades e melhoria de condições socioeconômicas, tendo como público específico estudantes que preencham critérios de vulnerabilidade, envolvendo o auxílio permanência e auxílio-moradia, por exemplo.

O DAE possui atualmente uma coordenadora e está subdividido em atendimento geral, restaurante, atendimento psicossocial e pedagógico, ambulatório, residência estudantil e lavanderia. A equipe de atendimento geral de apoio envolve uma telefonista e três auxiliares administrativos.

Na residência estudantil, voltada aos alunos do ensino técnico integrado ao ensino médio, há um coordenador, dois assistentes de alunos, um vigilante e um zelador. A equipe atende e dá suporte aos estudantes residentes e semi residentes.

Vejamos a definição de residentes e semirresidentes:

I - Estudante Residente: estudante regularmente matriculado nos cursos técnicos integrados ao ensino médio, com frequência às aulas e que esteja morando na área interna da instituição a esse fim destinada, com direito a pernoite e que concorrem à vaga na residência estudantil através de edital específico.

II - Estudante Semirresidente: estudante regularmente matriculado nos cursos técnicos integrados ao ensino médio, com frequência às aulas e que utilize o espaço a este fim destinado, sem direito a pernoite.

De acordo com os dados da Coordenação de Residência Estudantil, tem-se 145 residentes do sexo masculino e 82 residentes do sexo feminino, com um total de 167 alunos semirresidentes.

Há também uma lavanderia onde atua uma operadora de máquinas de lavar que atende os alunos residentes no que diz respeito às roupas das aulas práticas.

O atendimento psicossocial e pedagógico é voltado para todos os estudantes e conta com psicóloga, pedagogo, técnica em assuntos educacionais e assistente social, que realizam atendimentos e o acompanhamento de estudantes ao longo de sua trajetória acadêmica. Os estudantes buscam o atendimento frequentemente por questões relacionadas à sua adaptação escolar, questões de saúde mental, dificuldades de aprendizagem, bullying e conflitos interpessoais com seus pares.

O DAE atende desde alunos com dificuldades pedagógicas, passando também por alunos com dificuldades psicológicas específicas - como transtornos de ansiedade, TDAH (Transtorno de Déficit de Atenção e Hiperatividade), transtorno depressivo, entre outros. Também trabalha com as habilidades necessárias para convívio coletivo, especialmente na Residência Estudantil, de acordo com o Regulamento de conduta para estudantes residentes e semirresidentes.

Com relação aos cuidados com a saúde da comunidade acadêmica, o DAE oferece um ambulatório para atendimentos básicos em saúde e conta em sua equipe com técnica de enfermagem, enfermeira, médica e dentista. São realizados atendimentos de baixa complexidade e encaminhamento de casos de maior complexidade para órgãos de saúde externos.

O Restaurante contempla a questão da alimentação estudantil, atendendo também servidores e comunidade externa (com valores mais acessíveis aos discentes e totalmente gratuitos aos estudantes dos cursos técnicos). São oferecidas refeições de café da manhã No restaurante, especificamente, atuam uma nutricionista, uma cozinheira, com o apoio de uma equipe terceirizada(04 cozinheiras, 10 auxiliares de serviço de alimentação, 01 padeira e 02 higienizadoras).

O DAE desempenha um importante papel institucional, sendo uma rede de apoio multiprofissional no suporte aos estudantes em seu processo de ingresso, permanência e êxito acadêmico. Em especial, além da rede de atendimento, também há o Programa de Benefícios, que colabora imensamente com a permanência de estudantes em situação de vulnerabilidade social através de auxílios estudantis, hoje divididos em duas categorias: o auxílio permanência e o auxílio moradia.

Segundo os dados da Assistente Social, no Auxílio Moradia do Técnico em Agropecuária Integrado tem-se 11 alunos beneficiados e no PROEJA temos 2 alunos. Já no Auxílio Permanência temos 174 alunos beneficiados, com 18 do PROEJA.

* 1. **5.3 - Machine Learning**

Machine learning [6] é um ramo da inteligência artificial (IA) e da ciência da computação que se concentra no uso de dados e algoritmos para imitar a maneira como os humanos aprendem, melhorando gradualmente sua precisão.

Machine learning é um componente importante do crescente campo da ciência de dados. Por meio do uso de métodos estatísticos, os algoritmos são treinados para fazer classificações ou previsões, revelando os principais insights em projetos de mineração de dados. Esses insights subsequentemente conduzem a tomada de decisões em aplicativos e negócios, impactando de forma ideal as principais métricas de crescimento. Conforme o big data continua a se expandir e crescer, a demanda do mercado por cientistas de dados aumentará, exigindo que eles auxiliem na identificação das questões de negócios mais relevantes e, posteriormente, os dados para respondê-las.

Bonnin [11] define Machine Learning assim: *"Diz-se que um programa de computador aprende com a experiência E em relação a alguma classe de tarefas T e medida de desempenho P, se seu desempenho nas tarefas em T, medido por P, melhora com experiência E.”*

Esta definição está completa e restabelece os elementos que desempenham um papel em cada projeto de aprendizagem de máquina: a tarefa a realizar, as sucessivas experiências e uma clara e apropriada medida de desempenho apropriada. Em palavras mais simples, temos um programa que melhora como executa a tarefa baseada sobre experiência e guiada por um certo critério.

De acordo com Netto e Maciel [13] Machine Learning (Aprendizado de Máquina) é a disciplina que faz uso de toda uma série de procedimentos e algoritmos para identificar padrões, agrupamentos ou tendências e, então, extrair informação útil para análise de dados, de maneira totalmente automatizada. Grosso modo, pode-se dizer que são métodos matemáticos usados para treinar algoritmos que identificam padrões.

**5.3.1 Como o machine learning funciona**

UC Berkeley [7] quebra o sistema de aprendizado de um algoritmo de machine learning em três partes principais.

Um processo de decisão: Em geral, algoritmos de machine learning são usados para fazer uma predição ou classificação. Com base em alguns dados de entrada, que podem ser rotulados ou não rotulados, seu algoritmo produzirá uma estimativa sobre um padrão nos dados.

Uma função de erro: Uma função de erro serve para avaliar a predição do modelo. Se houver exemplos conhecidos, uma função de erro poderá fazer uma comparação para avaliar a precisão do modelo.

Um processo de otimização de modelo: Se o modelo pode se ajustar melhor aos pontos de dados no conjunto de treinamento, então os pesos são ajustados para reduzir a discrepância entre o exemplo conhecido e a estimativa do modelo. O algoritmo repetirá este processo de avaliação e otimização, atualizando os pesos de maneira autônoma até que um limite de precisão seja atingido.

Bonnin [11] nos diz que o aprendizado de máquina como a disciplina não é um campo isolado - isso é emoldurado dentro de um domínio mais amplo, **Inteligência Artificial** ( **AI** ). Mas, como você pode imaginar, o aprendizado de máquina não apareceu do nada. Como disciplina tem seus predecessores, e vem evoluindo em estágios de aumentar a complexidade em quatro etapas claramente diferenciadas a seguir:

1. O primeiro modelo de aprendizado de máquina envolvia decisões baseadas em regras e um simples nível de algoritmos baseados em dados que os inclui em si, e como pré-requisito, todos as possíveis ramificações e regras de decisão, implicando que todas as ramificações possíveis serão codificadas no modelo de antemão por um especialista na área. Esse estrutura foi implementada na maioria das aplicações desenvolvidas desde as primeiras linguagens de programação que apareceram em 1950. O principal tipo de dados e função tratada por esse tipo de algoritmo é o Booleano, pois ele tratou exclusivamente de decisões sim ou não.

2. Durante o segundo estágio de desenvolvimento do raciocínio estatístico, começamos a deixar que as características probabilísticas dos dados têm uma palavra a dizer, além das escolhas anteriores previamente definidas. Isso reflete melhor a natureza difusa dos problemas do mundo real, onde pontos fora da curva são comuns e onde é mais importante levar em consideração uma conta as tendências não determinísticas dos dados do que a abordagem rígida de questões fixas. Esse disciplina adiciona ao mix de elementos de ferramentas matemática da **teoria da probabilidade bayesiana** . Os métodos pertencentes a esta categoria incluem ajuste de curva (geralmente linear ou polinomial), os quais tem a propriedade comum de trabalhar com dados numéricos.

3. O estágio de aprendizado de máquina é o domínio no qual vamos trabalhar ao longo deste livro, e envolve tarefas mais complexas do que os mais simples elementos Bayesianos do estágio anterior.

A característica mais notável dos algoritmos de aprendizado de máquina é a de que eles podem generalizar modelos a partir de dados, mas os modelos são capazes de gerar seus próprios seletores de recursos, que não são limitados por uma função de destino rígida, pois são gerados e definidos à medida que o processo de treinamento evolui. Outro diferencial de esse tipo de modelo é que eles podem receber uma grande variedade de tipos de dados como entrada, como fala, imagens, vídeo, texto e outros dados suscetíveis de serem representados como vetores.

4. A IA é o último passo na escala de capacidades de abstração que, de certa forma, incluem todos os tipos de algoritmos anteriores, mas com uma diferença fundamental: os algoritmos de IA são capazes de aplicar o conhecimento aprendido para resolver tarefas que nunca foram consideradas durante o treinamento. Os tipos de dados com os quais este algoritmo trabalha são ainda mais genéricos do que os tipos de dados suportados pelo aprendizado de máquina, e eles devem ser capazes, por definição, de transferir capacidades de resolução de problemas de um tipo de dados para outro, sem um retreinamento completo do modelo. Desta forma, nós poderíamos desenvolver um algoritmo para detecção de objetos em imagens em preto e branco e o modelo abstrato poderia abstrair o conhecimento para aplicar o modelo para colorir imagens.

No diagrama a seguir, representamos esses quatro estágios de desenvolvimento em direção à IA real.

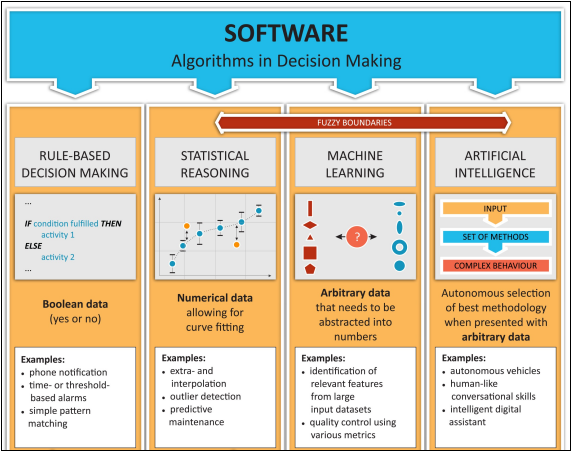


Figura 1 - Os algoritmos na tomada de decisão - [11]

Campesato [12] nos diz que em termos de alto nível, o aprendizado de máquina é um subconjunto de IA que pode resolver tarefas inviáveis ou muito complicadas com linguagens de programação mais tradicionais. Um filtro de spam para e-mail é um dos primeiros exemplos de aprendizado de máquina. O aprendizado de máquina geralmente substitui a precisão dos algoritmos mais antigos.

Campesato [12], continua, dizendo que apesar da variedade de algoritmos de aprendizado de máquina, os dados são indiscutivelmente mais importantes do que o algoritmo selecionado. Muitos problemas podem surgir com os dados, como dados insuficientes, baixa qualidade dos dados, dados incorretos, dados ausentes, dados irrelevantes, valores de dados duplicados e assim por diante.

Bonnin [11] ainda fala sobre os tipos de aprendizado de máquina:

**5.3.2 - Tipos de aprendizado de máquina**

Vamos tentar dissecar os diferentes tipos de projeto de aprendizado de máquina, começando pelo grau de conhecimento prévio do ponto de vista do implementador. O projeto pode ser dos seguintes tipos:

**5.3.2.1 - Aprendizagem supervisionada** : neste tipo de aprendizagem, recebemos um conjunto de amostras de dados, acompanhados do resultado que o modelo deve nos dar depois de aplicá-lo. Em estatístico termos, nós temos o resultado de todos os conjuntos de experimentos do treinamento.

**5.3.2.2 - Aprendizado não supervisionado:** esse tipo de aprendizado fornece apenas os dados de amostra do domínio do problema, além da tarefa de agrupar dados semelhantes e aplicar uma categoria que não tem informação prévia das quais ela pode ser inferida.

**5.3.2.3- Aprendizado por reforço:** esse tipo de aprendizado não possui um conjunto de amostras rotuladas e tem um número diferente de elementos participantes, que incluem um agente, um ambiente, um aprendizado e uma política ótima ou um conjunto de etapas, maximizando uma abordagem orientada a objetivos usando recompensas ou penalidades (o resultado de cada tentativa).

Observe o diagrama a seguir:

|  |  |
| --- | --- |
|  |  |
|  |  |

Figura 2 - Principais áreas do Machine Learning[11]

Campesato assim define os tipos de aprendizado [12]:

Aprendizado supervisionado significa que os pontos de dados em um conjunto de dados possuem um rótulo que identifica seu conteúdo. Por exemplo, o conjunto de dados MNIST contém arquivos PNG 28x28, cada um contendo um único dígito desenhado à mão (ou seja, 0 a 9 inclusive). Toda imagem com o dígito 0 tem o rótulo 0; toda imagem com o dígito 1 tem o rótulo 1; todas as outras imagens são rotuladas de acordo com o dígito exibido nessas imagens.

Como outro exemplo, as colunas no conjunto de dados do Titanic são características dos passageiros, como sexo, classe da cabine, preço da passagem, se o passageiro sobreviveu ou não e assim por diante. Cada linha contém informações sobre um único passageiro, incluindo o valor 1 se o passageiro sobreviveu. O conjunto de dados MNIST e o conjunto de dados Titanic envolvem tarefas de classificação: o objetivo é treinar um modelo com base em um conjunto de dados de treinamento e, em seguida, prever a classe de cada linha em um conjunto de dados de teste.

Em geral, os conjuntos de dados para tarefas de classificação têm um pequeno número de valores possíveis: um dos nove dígitos no intervalo de 0 a 9, um dos quatro animais (cachorro, gato, cavalo, girafa), um dos dois valores (sobreviveu versus pereceu , comprado versus não comprado). Como regra geral, se o número de saídas puder ser exibido em um número relativamente pequeno de valores (que é um número subjetivo) em uma lista drop-down, provavelmente é uma tarefa de classificação.

No caso de um conjunto de dados que contém dados imobiliários, cada linha contém informações sobre uma casa específica, como número de quartos, metros quadrados da casa, número de banheiros, preço da casa e assim por diante. Neste conjunto de dados, o preço da casa é o rótulo de cada linha.

Observe que a faixa de preços possíveis é muito grande para caber razoavelmente bem em uma lista drop-down. Um conjunto de dados imobiliário envolve uma tarefa de regressão : o objetivo é treinar um modelo com base em um conjunto de dados de treinamento e, em seguida, prever o preço de cada casa em um conjunto de dados de teste.

O aprendizado não supervisionado envolve dados não rotulados, o que normalmente é o caso de algoritmos de agrupamento (discutidos posteriormente). Alguns algoritmos importantes de aprendizado não supervisionado que envolvem agrupamento são os seguintes:

* k-means
* análise de cluster hierárquico (HCA)
* maximização da expectativa

Alguns algoritmos importantes de aprendizado não supervisionado que envolvem redução de dimensionalidade são as seguintes:

* análise de componentes principais (PCA)
* kernel PCA
* incorporação linear localmente (LLE)
* incorporação estocástica de vizinhança t-distribuída (t-SNE)

Há mais uma tarefa não supervisionada muito importante chamada detecção de anomalias. Esta tarefa é relevante para detecção de fraudes e detecção de outliers.

O aprendizado semisupervisionado é uma combinação de aprendizado supervisionado e não supervisionado: alguns pontos de dados são rotulados e outros não rotulados. Uma técnica envolve o uso de dados rotulados para classificar (ou seja, rotular) os dados não rotulados, após o que você pode aplicar um algoritmo de classificação.

Segundo Netto e Maciel [13], os algoritmos de aprendizagem supervisionada podem ser de Classificação ou de Regressão. Algoritmos de Classificação tratam de problemas em que os dados têm uma classificação prévia e se deseja prever a qual categoria um dado não classificado pertence. Por exemplo, o problema da classificação dos e-mails legítimos e dos spams.

Nesse caso, treinamos o algoritmo de classificação para identificar quais são as características de um e-mail legítimo e quais as de um spam. Treinado, o algoritmo pode testar se um e-mail (ou um conjunto de palavras) pertence a uma das duas classes.

Regressão é utilizada se um ou mais atributos são variáveis contínuas (reais) e dependentes. Ou seja, a mudança em uma variável implica a mudança da outra. O exemplo claro aqui é o do investimento em pesquisa, administração e propaganda nas cinquenta startups. O conjunto é formado por variáveis numéricas — reais e contínuas (em contraste a numéricas discretas ou nominais).

Pode-se dizer que algoritmos de classificação atribuem rótulos aos dados (em alguns materiais, você encontrará a expressão labels), classificando-os, como o nome indica, em grupos. Diz-se que os algoritmos de classificação produzem como saída um atributo classe ou meta.

Algoritmos de regressão, por sua vez, produzem valores, que tentam prever alguma característica numérica dos dados.

**5.3.2.5 - Algoritmos de regressão**

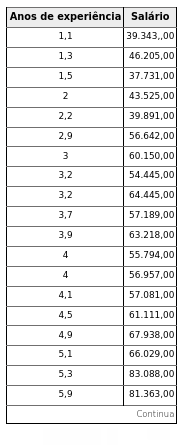
São cinco os tipos de algoritmos de regressão mais comumente estudados em cursos introdutórios de Machine Learning: Linear, Polinomial, Árvore de Decisão (Decision Tree) e Floresta Randômica ou Floresta Aleatória (Random Forest).

Pergunta: Existe uma relação entre antiguidade na empresa e salário recebido?

Esse tipo de algoritmo tenta estabelecer uma relação entre duas variáveis (no exemplo, tempo de serviço em anos e o salário recebido). Ou em outras palavras: é possível prever o salário a partir do tempo de serviço? Ou mesmo se pode inferir quantos anos de serviço um funcionário tem a partir do salário?

Uma maneira visual de entender o conceito por trás da regressão é montar um gráfico com as variáveis em dois eixos e marcar os pontos correspondentes. A regressão linear é a maneira pela qual conseguimos encontrar uma reta (linha) que ligue todos os pontos, ou, mais precisamente, uma reta que tenha a menor distância possível dos pontos.

O quadro a seguir representa parte dos dados utilizados.



Quadro 1

**5.3.3.1 - Algoritmos de classificação**

Como abordado anteriormente, esses algoritmos são utilizados para criar partições de conjuntos, um nome pomposo para subconjuntos próprios. Como visto, são muito úteis em problemas em que uma classe (ou classes) deve ser definida. Pode haver algoritmos de classificação que usem aprendizagem supervisionada ou não supervisionada. As definições de aprendizagem supervisionada e não supervisionada serão apresentadas na próxima seção.

São exemplos de algoritmos de classificação: K-N,N (K-Nearest Neighbors), SVM (Support Vector Machines), Regressão Logística (que, apesar do nome, é um algoritmo de classificação), entre outros.

**5.3.3.2 - Aprendizagem não supervisionada**

Em alguns algoritmos (principalmente nos de aprendizagem supervisionada), é comum a aprendizagem ser dividida em duas etapas: treino e previsão. Usa-se parte dos dados (geralmente entre 70% e 80%) para “treinar” o algoritmo, e a outra parte, para prever os resultados, testar o algoritmo na parte restante e verificar se ele está próximo do resultado correto. Obviamente, quanto maior o tamanho dos dados, melhor será a previsão.

No entanto, o melhor algoritmo utilizado servirá muito bem para o conjunto de testes. No mundo real, onde a quantidade de dados aumenta exponencialmente, é necessário que esse algoritmo sempre seja atualizado e volte a ser testado sobre uma base de dados real.

Aplica-se aqui um conceito de cross-validation, ou validação cruzada, que é separar as partes de treino e teste em subconjuntos distintos que abranjam todo o conjunto de dados.

**5.4 - Análise Preditiva**

De acordo com [14], a análise preditiva se baseia em um conceito simples: prever a probabilidade de eventos futuros com base em dados históricos. Sua história pode remontar a pelo menos 650 aC. Alguns dos primeiros exemplos incluem os babilônios, que tentaram prever mudanças climáticas de curto prazo com base no aparecimento de nuvens e halos.

A medicina também tem uma longa história de necessidade de classificar as doenças. O rei babilônico Adad apla-iddina decretou que os registros médicos fossem coletados para formar o Manual de Diagnóstico. Algumas previsões neste corpus listam tratamentos com base no número de dias em que o paciente esteve doente e em sua pulsação. Uma das primeiras instâncias da bioinformática.

Posteriormente, a análise preditiva especializada foi desenvolvida no início das indústrias de subscrição de seguros. Isso foi usado como uma forma de prever o risco associado ao seguro de embarcações marítimas [22]. Mais ou menos na mesma época, as seguradoras de vida começaram a prever a idade que uma pessoa viveria para definir as taxas de prêmio mais adequadas.

Embora a ideia de previsão sempre parecesse enraizada desde cedo na necessidade humana de entender e classificar, não foi até o século 20 , e com o advento da computação moderna, que ela realmente se consolidou.

Além de ajudar o governo dos Estados Unidos na década de 1940 a quebrar códigos, Alan Turing também trabalhou nos primeiros algoritmos de xadrez de computador que colocavam o homem contra a máquina. Os métodos de simulação de Monte Carlo originaram-se como parte do projeto Manhattan, onde computadores mainframe processavam números durante dias para determinar a probabilidade de ataques nucleares.

Na década de 1950, desenvolveu-se a teoria da Pesquisa Operacional (PO) , na qual se poderia otimizar a menor distância entre dois pontos. Até hoje, essas técnicas são utilizadas na logística por empresas como UPS e Amazon.

Não-matemáticos também entraram em ação. Na década de 1970, o cardiologista Lee Goldman (que trabalhava a bordo de um submarino) passou anos desenvolvendo uma árvore de decisão que fazia isso com eficiência. Isso ajudou a equipe a determinar se o submarino precisava ou não ressurgir para ajudar os que sofriam de dores no peito.

O que muitos desses exemplos tinham em comum era que as pessoas primeiro faziam observações sobre eventos que já haviam ocorrido e depois usavam essas informações para generalizar e tomar decisões sobre o que poderia ocorrer no futuro. Junto com a previsão, veio uma maior compreensão de causa e efeito e como as várias partes do problema estavam inter-relacionadas. A descoberta e o insight surgiram por meio da metodologia e da adesão ao método científico.

Mais importante, eles surgiram para encontrar soluções para problemas importantes e muitas vezes práticos da época. Foi isso que os tornou únicos.

**5.4.1 - A análise preditiva está em muitos setores**

Percorremos um longo caminho desde então, e as soluções analíticas práticas promoveram o crescimento em muitos setores diferentes. A internet teve um efeito profundo nisso; permitiu que cada clique fosse armazenado e analisado.

**5.4.1.1 - Análise preditiva em marketing**

Uma indústria que abraçou a PA por um bom tempo é o marketing. O marketing sempre se preocupou com a aquisição e retenção de clientes e desenvolveu modelos preditivos envolvendo várias ofertas promocionais e pontos de contato com o cliente, todos voltados para manter clientes e conquistar novos. Isso é muito acentuado em determinados segmentos de marcação, como cartões de compras sem fio e online, nos quais os clientes estão sempre em busca do melhor negócio.

O mix de marketing e a modelagem de atribuição se tornem cada vez mais importantes. Os dispositivos baseados em localização permitiram que os aplicativos preditivos de marketing incorporassem dados em tempo real para emitir recomendações ao cliente enquanto ele estava na loja.

**5.4.1.2 - Análise preditiva na área da saúde**

A análise preditiva na área da saúde tem suas raízes em ensaios clínicos, que usam amostras cuidadosamente selecionadas para testar a eficácia de medicamentos e tratamentos. Os dados podem ser incorporados à análise preditiva para monitorar pacientes com doenças críticas e enviar alertas ao paciente quando ele estiver em risco.

As empresas de assistência médica agora podem prever quais pacientes individuais seguirão os cursos de tratamento preconizados pelos provedores de saúde. Isso enviará sinais de alerta precoce a todas as partes, o que evitará complicações futuras, além de reduzir o custo total do tratamento.

**5.4.1.3 - Análise preditiva em outros setores**

Outros exemplos podem ser encontrados em quase todos os outros setores. Aqui estão apenas alguns:

**5.4.1.4 - Finanças**

A detecção de fraudes é uma área enorme. As instituições financeiras são capazes de monitorar as transações internas e externas do cliente em busca de fraude, por meio de reconhecimento de padrões e outros algoritmos de aprendizado de máquina, e alertar o cliente sobre atividades suspeitas. As análises geralmente são realizadas em tempo real.

* 1. **5.4.1.5 - Programa de Negociação em Wall Street**
     + 1. 1. Os algoritmos de negociação preverão altos e baixos intradiários e decidirão quando comprar e vender títulos.
  2. **5.4.1.6 - Gerenciamento de esportes**
     1. A análise de gerenciamento de esportes é capaz de prever quais eventos esportivos renderão o maior público e instituir preços de ingressos variáveis com base no interesse do público.
  3. **5.4.1.7 - Ensino superior**
     + 1. As faculdades podem prever quantos e quais tipos de alunos provavelmente frequentarão o próximo semestre e podem planejar os recursos de acordo.
  4. As avaliações baseadas no tempo dos módulos on-line podem permitir que os professores identifiquem as áreas de problemas potenciais dos alunos e adaptem a instrução individual.
  6. **5.4.1.8 - Governo**
  7. Os governos federal e estadual adotaram o conceito de dados abertos e disponibilizaram mais dados ao público, o que capacitou os Cientistas de Dados dos Cidadãos a ajudar a resolver problemas sociais e governamentais críticos.

O uso potencial de dados para fins de serviços de emergência, segurança no trânsito e uso de saúde é extremamente positivo.

Embora esses setores possam ser bastante diferentes, os objetivos da análise preditiva são normalmente implementados para aumentar a receita, diminuir os custos ou alterar os resultados para melhor.

Para [15] , para entender a análise preditiva, vamos nos referir a uma famosa definição de aprendizado de máquina de Tom Mitchell, onde ele explicou o que realmente significa aprender do ponto de vista da ciência da computação:

“Diz-se que um programa de computador aprende com a experiência E em relação a alguma classe de tarefas T e medida de desempenho P, se seu desempenho em tarefas em T, conforme medido por P, melhora com a experiência E”

Com base nessa definição, podemos concluir que um programa ou máquina de computador pode:

• Aprenda com dados e históricos

• Pode ser melhorado com a experiência

• Aprimorar interativamente um modelo que pode ser usado para prever um resultado

Tarefas típicas de aprendizado de máquina são aprendizado de conceito, modelagem preditiva, agrupamento e localização de padrões úteis. O objetivo final é melhorar a aprendizagem de forma que ela se torne automática: para que não sejam mais necessárias interações humanas ou reduza o nível de interação humana tanto quanto possível.

A análise preditiva, por outro lado, é o processo de extrair informações úteis de fatos históricos e transmitir dados (consistindo em objetos de dados ao vivo) para determinar padrões ocultos e prever resultados e tendências futuras.

**5.4.2 - O que a análise preditiva não faz**

A análise preditiva não informa o que acontecerá no futuro, mas sim a criação de modelos preditivos que colocam um valor numérico, ou pontuação, na probabilidade de um determinado evento acontecer no futuro com um nível aceitável de confiabilidade e inclui cenários hipotéticos e avaliação de riscos.

**5.4.2.1 - Por que análise preditiva?**

Na área de inteligência de negócios, com a plataforma correta de gerenciamento de operações, os tomadores de decisão são capazes de gerenciar todas as entradas, eventos e dados relacionados aos negócios que fornecem informações em tempo real para o nível corporativo. Posteriormente, modelos preditivos podem ser usados para identificar padrões úteis de dados históricos, transacionais e recentes para identificar riscos e oportunidades potenciais. Portanto, está ganhando muita atenção e ampla aceitação. Além disso, usando as ferramentas tradicionais de relatórios e monitoramento, você pode passar de operações reativas para operações proativas. A PA ajuda a ir além disso para planejar o futuro e identificar novas áreas de negócios para lucro e produtividade.

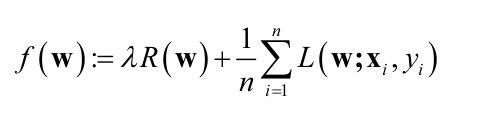
**5.4.2.2 - Princípios de funcionamento de um modelo preditivo**

Estando no centro da análise preditiva, muitas funções de aprendizado de máquina podem ser formuladas como um problema de otimização convexa para encontrar um minimizador de uma função convexa f que depende de um vetor variável w (pesos), que possui d registros.

Formalmente, podemos escrever isso como problema de otimização , onde



a função objetivo é da forma:





Aqui os vetores são os pontos de dados de treinamento

para são seus rótulos correspondentes que queremos prever eventualmente. Chamamos o método de linear se L(w;x,y) puder ser expresso como uma função de wTx e y.

A função objetivo f possui dois componentes: i) um regularizador que controla a complexidade do modelo, e ii) a perda que mede o erro do modelo nos dados de treinamento. A função de perda L(w;) é tipicamente uma função convexa em w. O parâmetro de regularização fixa  define o trade-off entre os dois objetivos de minimizar a perda no erro de treinamento e minimizar a complexidade do modelo para evitar o overfitting.

Uma compreensão mais simplificada pode ser obtida na figura 3: você tem os dados ou observações atuais. Agora é sua chance de usar a caixa preta para prever o resultado futuro com base nos dados atuais e fatos históricos. Nesse contexto, todos os valores indecisos são chamados de parâmetros, e a descrição, ou seja, a caixa preta, é um modelo PA:

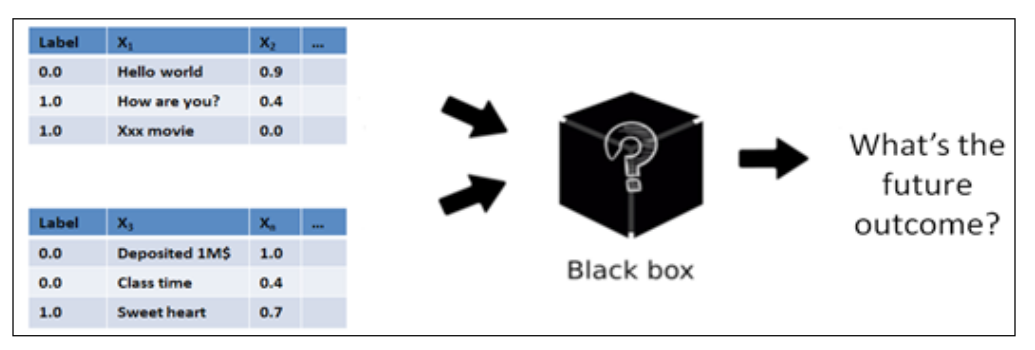


Figura 3: a principal tarefa na análise preditiva é a modelagem preditiva, ou seja, usar a caixa preta [15]

Como engenheiro ou desenvolvedor, você deve escrever um algoritmo que observe os parâmetros/ dados/amostras/exemplos existentes para treinar a caixa preta e descobrir como ajustar os parâmetros para obter o melhor modelo para fazer previsões antes da implantação.

No aprendizado de máquina, observamos o desempenho de um algoritmo em duas etapas: aprendizado e inferência. O objetivo final da etapa de aprendizado é preparar e descrever os dados disponíveis, também chamados de vetores de recursos, que são usados para treinar o modelo.

A fase de aprendizagem é uma das etapas mais importantes, mas também é verdadeiramente demorada. Envolve a preparação de uma lista de vetores também chamados de vetores de recursos (na maioria das vezes) a partir dos dados de treinamento após a transformação, para que possamos alimentá-los com os algoritmos de aprendizado. Por outro lado, os dados de treinamento às vezes também contêm informações impuras que precisam de algum pré-processamento, como limpeza.

Uma vez que temos os vetores de características, o próximo passo nesta etapa é preparar (ou escrever/reutilizar) o algoritmo de aprendizado. O próximo passo importante é treinar o algoritmo para preparar o modelo preditivo. Normalmente (e, claro, com base no tamanho dos dados), a execução de um algoritmo pode levar horas (ou até dias) para que os recursos convirjam em um modelo útil, conforme mostrado na figura a seguir:

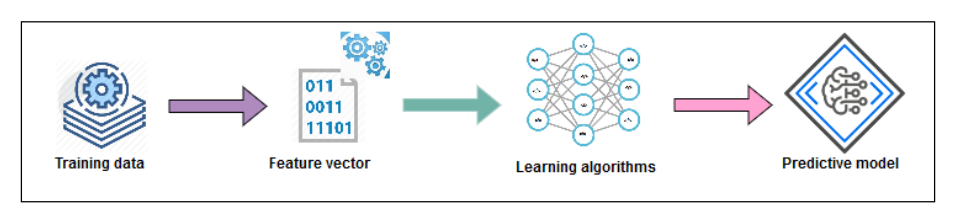


Figura 4: Aprendendo e treinando um modelo preditivo – mostra como gerar os vetores de features a partir dos dados de treinamento para treinar o algoritmo de aprendizado que produz um modelo preditivo [15]

**5.4.2.3 - Métodos comuns de análise preditiva**

Os métodos comuns de análise preditiva incluem análise de regressão, classificação, previsão de séries temporais, mineração de regra de associação, agrupamento, sistemas de recomendação e mineração de texto, análise de sentimento e muito mais. Agora, para preparar os vetores de características, precisamos saber um pouco sobre matemática, estatística e assim por diante.

O segundo estágio mais importante é a inferência que é usada para fazer um uso inteligente do modelo, como prever a partir de dados nunca antes vistos, fazer recomendações, deduzir regras futuras e assim por diante. Normalmente, leva menos tempo em comparação com a etapa de aprendizado e, às vezes, até em tempo real, como mostra a figura a seguir:

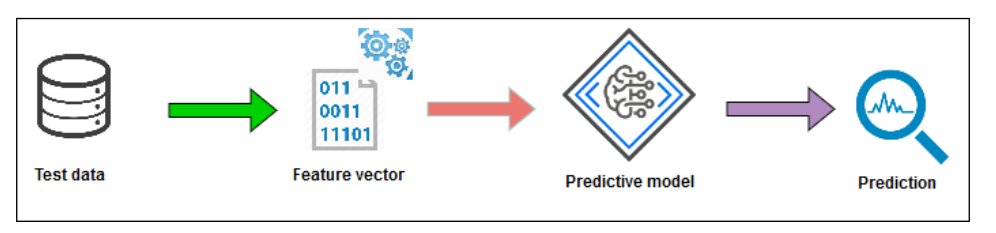


Figura 5: Inferência de um modelo existente para análise preditiva (vetores de features são gerados a partir de dados desconhecidos para fazer previsões) [15]

Assim, a inferência (consulte a figura 4 para obter mais informações) consiste em testar o modelo em relação a dados novos (ou seja, não observados) e avaliar o desempenho do próprio modelo.

No entanto, em todo o processo e para tornar o modelo preditivo bem-sucedido, os dados atuam como cidadãos de primeira classe em todas as tarefas de aprendizado de máquina.

Na realidade, os dados que alimentamos nossos sistemas de aprendizado de máquina devem ser objetos matemáticos, como vetores, matrizes ou gráficos (nos referiremos a eles como tensores para deixar mais claro) para que possam consumir esses dados :

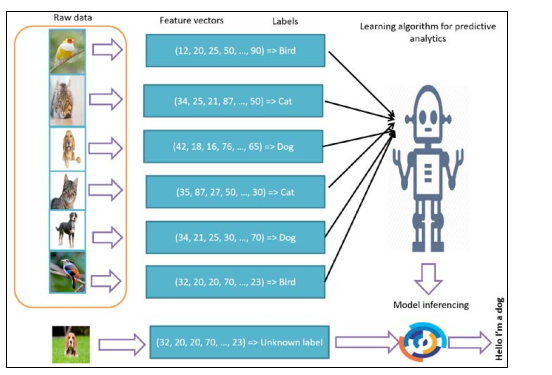


Figura 6: Os vetores de recursos estão em toda parte - eles são usados nos estágios de aprendizado e inferência na análise preditiva [15]

Dependendo dos dados disponíveis e dos tipos de recursos, o desempenho do seu modelo preditivo pode variar drasticamente. Portanto, selecionar os recursos corretos é uma das etapas mais importantes antes que a inferência ocorra. Isso é chamado de engenharia de recursos, que pode ser definido da seguinte forma:

**5.5 Engenharia de recursos**

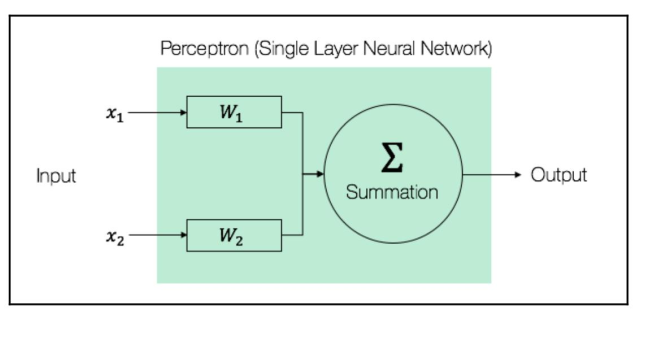
Nesse processo, o conhecimento do domínio sobre os dados é usado para criar apenas recursos seletivos ou úteis que ajudam a preparar os vetores de recursos a serem usados para que um algoritmo de aprendizado de máquina funcione.

Por exemplo, comprar um carro; você costuma ver features como nome do modelo, cor, potência, preço e número de assentos. Assim, considerando essas características, comprar um carro não é um problema trivial. A regra geral de aprendizado de máquina é que quanto mais dados houver, melhor será o modelo preditivo. No entanto, ter mais recursos geralmente cria uma confusão, de modo que o desempenho diminui drasticamente: especialmente se o conjunto de dados for de alta dimensão e esse fenômeno for chamado de maldição da dimensionalidade.

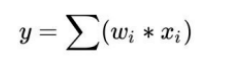
**5.6 Redes Neurais**

As redes neurais são uma classe de algoritmos de aprendizado de máquina vagamente inspirados por neurônios no cérebro humano. No entanto, sem me aprofundar muito nas analogias cerebrais, acho mais fácil simplesmente descrever as redes neurais como uma função matemática que mapeia uma determinada entrada para a saída desejada. Para entender o que isso significa, vamos dar uma olhada em uma rede neural de camada única (conhecida como perceptron)[23]

Um Perceptron pode ser ilustrado com o seguinte diagrama:



Em sua essência, o Perceptron é simplesmente uma função matemática que recebe um conjunto de entradas, realiza alguns cálculos matemáticos e gera o resultado do cálculo. Nesse caso, essa função matemática é simplesmente isso:



Wi refere-se aos pesos do Perceptron. Vamos explicar o que os pesos em uma rede neural se refere posteriomente. Por enquanto, precisamos apenas ter em mente que as redes neurais são simplesmente funções matemáticas que mapeiam uma determinada entrada para uma saída desejada.

Por que redes neurais?

Antes de mergulharmos na criação de nossa própria rede neural, vale a pena entender por que as redes neurais ganharam uma posição tão importante no aprendizado de máquina e na IA.

A primeira razão é que as redes neurais são aproximadores de funções universais. O que isso significa é que, dada qualquer função arbitrária que estamos tentando modelar, não importa quão complexa seja, as redes neurais sempre são capazes de representar essa função. Isso tem uma implicação profunda nas redes neurais e na IA em geral. Assumindo que qualquer problema no mundo pode ser descrito por uma função matemática (não importa quão complexa), podemos usar redes neurais para representar essa função, efetivamente modelando qualquer coisa no mundo. Uma ressalva a isso é que, embora os cientistas tenham provado a universalidade das redes neurais, uma rede neural grande e complexa pode nunca ser treinada e generalizada corretamente.

A segunda razão é que a arquitetura das redes neurais é altamente escalável e flexível. Como veremos a seguir, podemos facilmente empilhar camadas em cada rede neural, aumentando a complexidade da rede neural. Talvez mais interessante, as capacidades das redes neurais são limitadas apenas por nossa própria imaginação. Por meio do design criativo da arquitetura de rede neural, os engenheiros de aprendizado de máquina aprenderam como usar redes neurais para prever dados de séries temporais (conhecidas como redes neurais recorrentes (RNNs)), que são usadas em áreas como reconhecimento de fala. Nos últimos anos, os cientistas também mostraram que, ao colocar duas redes neurais uma contra a outra em uma competição (conhecida como rede adversária generativa (GAN)), podemos gerar imagens fotorrealistas indistinguíveis ao olho humano.

Aqui veremos a arquitetura básica das redes neurais, os blocos de construção nos quais todas as redes neurais complexas são baseadas.

As redes neurais consistem nos seguintes componentes:

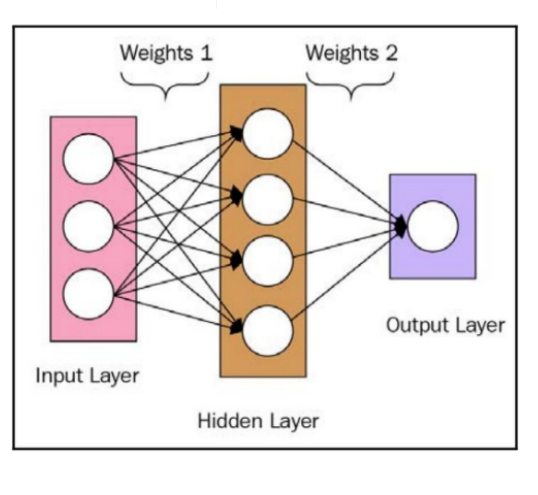
Uma camada de entrada, x

Uma quantidade arbitrária de camadas ocultas Uma camada de saída, ÿ

Um conjunto de pesos e bias entre cada camada, W e b

Uma escolha de função de ativação para cada camada oculta, ÿ

O diagrama a seguir mostra a arquitetura de uma rede neural de duas camadas (observe que a camada de entrada normalmente é excluída ao contar o número de camadas em uma rede neural):



## Apresentando a não linearidade

Este não é um mau momento para dar um passo atrás e refletir sobre o que estamos tentando alcançar, pois isso nos ajudará a entender a ideia por trás do motivo pelo qual queremos empregar algo como uma função de ativação[24].

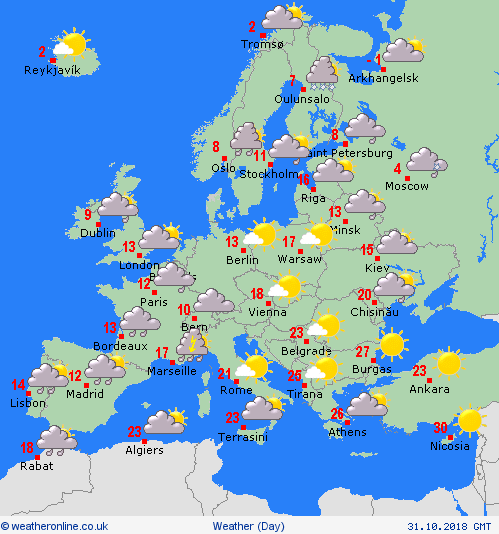
Bem, você vê, os dados do mundo real geralmente não são lineares. O que queremos dizer com isso é que sempre que tentamos modelar uma observação como uma função de diferentes entradas, esta função em si não pode ser representado linearmente, ou em uma linha reta.

Se todos os padrões nos dados constituíssem apenas linhas retas, provavelmente não estaríamos discutindo redes neurais em tudo. Técnicas como **Support Vector Machines** ( **SVMs** ) ou mesmo a regressão linear já se destaca nessa tarefa:

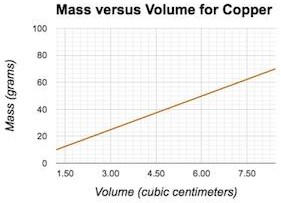


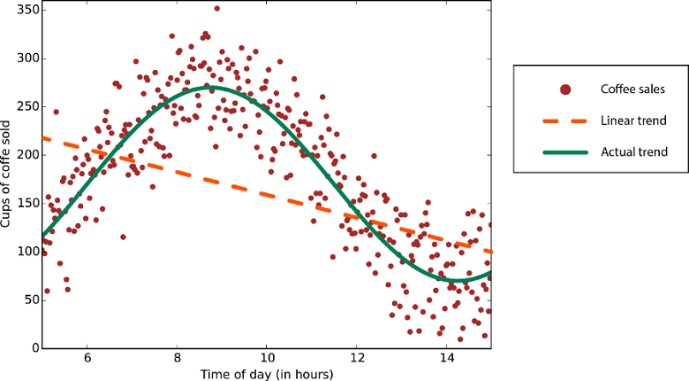
A modelagem de dias ensolarados e chuvosos com temperatura, por exemplo, produzirá uma curva. Com efeito, isso significa apenas que não podemos separar nosso limite de decisão usando uma linha reta. Ou seja, em alguns dias pode chover apesar das altas temperaturas, e nos outros dias é pode permanecer ensolarado apesar Baixas temperaturas.

Isso ocorre porque a temperatura não está linearmente relacionada ao clima. O resultado do tempo em qualquer dia é muito provável que seja uma função complexa, envolvendo variáveis interativas como como velocidade do vento, pressão do ar e muito mais. Assim, em qualquer dia, uma temperatura de 13 graus poderia significa um dia ensolarado em Berlim, Alemanha, mas um dia chuvoso em Londres, REINO UNIDO:



Existem alguns casos, é claro, em que um fenômeno pode ser representado linearmente. Em física, por exemplo, a relação entre a massa de um objeto e seu volume pode ser linearmente definido, como mostrado na seguite captura de tela:



Esse é um exemplo de uma função não linear :

|  |  |
| --- | --- |
| **Uma função linear** | **Uma função não linear** |
| *Y = m x + b* | *Y = m x 2 + b* |

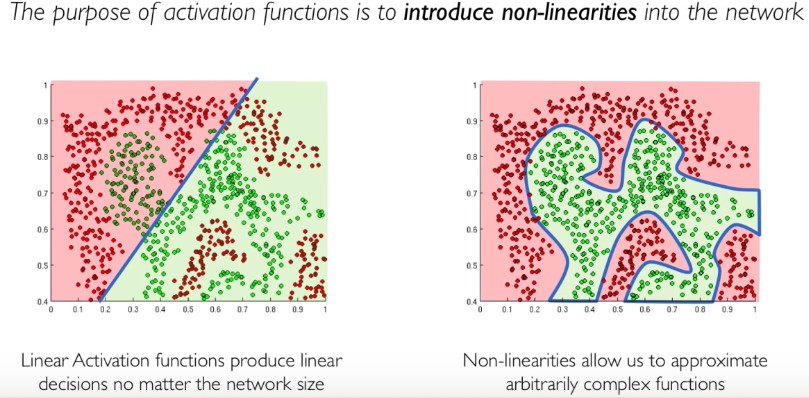
Aqui, *m* é a inclinação da linha, *x* é qualquer ponto (uma entrada ou um valor *x* ) na linha e *b* é onde o linha cruza o eixo *y* .

Infelizmente, a linearidade geralmente não é garantida com dados do mundo real, pois modelamos observações usando múltiplos recursos, cada um dos quais pode ter uma variedade e contribuição desproporcional para determinar nossas classes de produção. Na verdade, nosso mundo é extremamente não linear e, portanto, para capturar essa não linearidade em nosso modelo de perceptron, precisamos incorporar funções não lineares que sejam capazes de representar tais fenômenos. Ao fazer isso, aumentamos a capacidade do nosso neurônio para modelar padrões que na verdade existem no mundo real, e ‘desenham’ decisão limites que seria não possível, se usássemos apenas funções lineares. Esses tipos de funções, usados para modelar relacionamentos não lineares em nossos dados, são conhecidos como  **função de ativação** .

**função de Ativação**

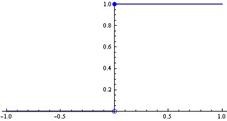
Bem, basicamente o que fizemos até agora foi representar nossos diferentes recursos de entrada e seus pesos em uma dimensão inferior, como uma representação escalar. Podemos usar esta redução de representação e passá-la por uma função não linear simples que nos diz se nossa representação está acima ou abaixo de um determinado valor limite. Semelhante aos pesos que inicializamos antes, esse valor limite pode ser pensado como um parâmetro apreensível de nosso modelo perceptron.

Em outras palavras, queremos que nosso perceptron descubra as combinações ideais de pesos e um limite, permitindo que ele corresponda de forma confiável nossas entradas à classe de saída correta. Portanto, nós comparamos nossa representação de recurso reduzida com um valor limite e, em seguida, ativamos nossa unidade perceptron se estivermos acima desse valor limite ou não fizermos nada caso contrário. Esta função muito que compara nosso valor de recurso reduzido com um limite, é conhecida como **função de ativação** :

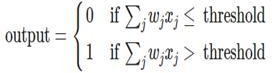


Essas funções não lineares vêm em diferentes formas e serão exploradas com mais detalhes posteriormente. Por enquanto, apresentamos duas funções de ativação diferentes; o passo do **conjunto pesado** e a função de ativação **logística sigmoide** . A unidade perceptron que anteriormente mostrada foi originalmente implementado com uma função de etapa tão pesada, levando a saídas binário de 1 (ativo) ou 0 (inativo). Usando a função step em nossa unidade perceptron, observe que um valor acima da curva levará à ativação (1), enquanto um valor abaixo ou na curva não levará ao disparo da unidade de acionamento (0). Este processo pode ser resumido em também de maneira algébrica.

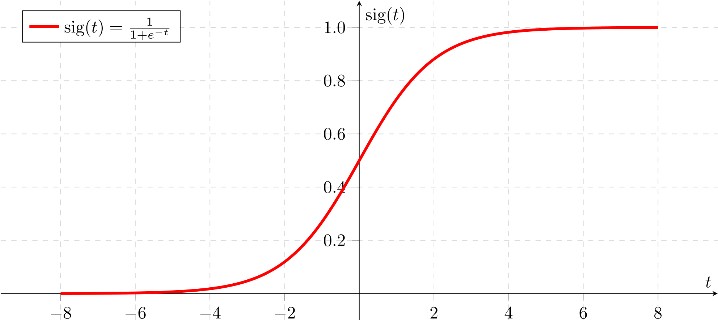
O seguindo captura de tela shows o pesado etapa função:



A saída limite da Fórmula é como segue:

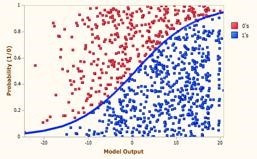


Em essência, uma função step não é realmente uma função não linear, pois pode ser reescrita como duas combinações lineares finitas. Assim, esta função constante por partes não é muito flexível em modelagem de dados do mundo real, que geralmente são mais probabilísticos do que binários. A logística sigmoide, por outro lado, é de fato uma função não linear e pode modelar dados com mais flexibilidade. Esta função é conhecida por **reduzir** sua entrada a um valor de saída entre 0 e 1, o que a torna uma função popular para representar probabilidades, e é comumente empregado função de ativação para neurônios em rede neural moderna:



Cada tipo de função de ativação vem com seu próprio conjunto de vantagens e desvantagens que também abordaremos em capítulos posteriores. Por enquanto, você pode pensar intuitivamente sobre a escolha de diferentes funções de ativação como uma consideração com base no seu tipo específico de dados. Em outras palavras, idealmente tentamos experimentar e escolher uma função que melhor capture o sublinhado tendências que talvez estejam presente nos seus dados.

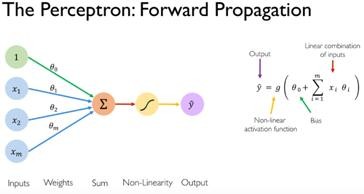
Portanto, empregaremos tais funções de ativação para limitar as entradas de um neurônio. As entradas são consequentemente transformadas e calibradas contra esta ativação limite, por sua vez, fazendo com que um neurônio dispare ou se abstenha disso. Na seguinte ilustração, nós podemos visualizar a decisão limite produzida por uma função de ativação.



## Entetendendo o papel do termo viés

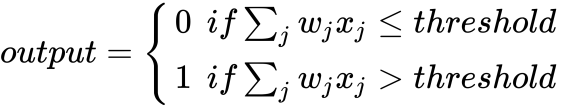
Então, agora temos uma boa ideia de como os dados entram em nosso perceptron; está emparelhado com pesos e reduzidos através de um produto escalar, apenas para serem comparados a uma ativação limite. Podemos nos perguntar neste ponto, *e se quiséssemos que nosso limite se adaptasse a padrões diferentes nos dados?* Em outras palavras, e se os limites da função de ativação não eram ideais para identificar separadamente os padrões específicos que queremos que nosso modelo aprenda? Nós precisamos poder jogar com a forma da nossa curva de ativação, de forma a garantir algum flexibilidade no organizar de padrões cada neurônio pode capturar localmente.

E como exatamente vamos moldar nossa função de ativação? Bem, uma maneira de fazer isso é por introduzindo um **termo de viés** em nosso modelo. Isso é representado pela seta saindo da primeira entrada o nó (marcado com o número '1') no diagrama a seguir:

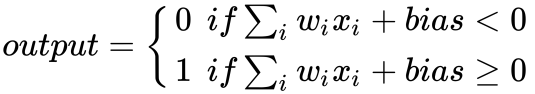


Representativamente, podemos pensar nesse viés como o peso de uma entrada **ficcional** *.* Este fictício diz-se que a entrada está sempre presente, permitindo que nossa unidade de ativação dispare à vontade, sem exigir que quaisquer recursos de entrada estejam explicitamente presentes (como mostrado no círculo verde anteriormente). A motivação por trás desse termo é ser capaz de manipular a forma da nossa função de ativação, que por sua vez afeta o aprendizado de nosso modelo. Queremos que nossa forma de ajustar de forma flexível diferentes padrões em nossos dados. O peso do termo de viés é atualizado na mesma maneira como todos os outros pesos são. O que o torna diferente é que não é perturbado por sua entrada de neurônio, a qual simplesmente sempre detém o valor constante (como mostrando anteriormente).

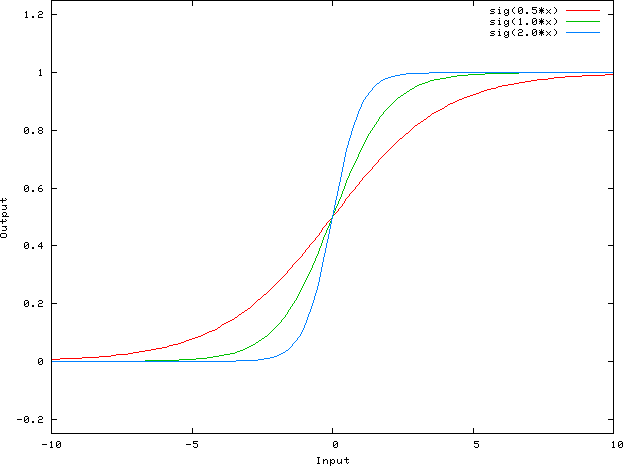
Então, como realmente influenciamos nosso limite de ativação usando esse termo de viés? Bem, vamos considerar um exemplo simplificado. Suponha que temos algumas saídas geradas por um função de ativação escalonada, que produz qualquer um '0' ou um '1' para cada saída, assim:



Nós podemos então reescrever essa Fórmula para incluir o termo viés, como segue:

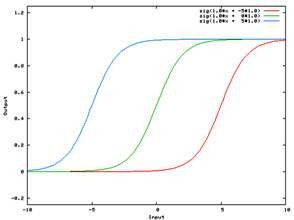
Em outras palavras, estamos usando mais um truque matemático e redefinindo o valor limite como o negativo do nosso termo viés ( *Threshold = -(bias)* ). Este termo de viés é aleatoriamente inicializado no início de nossa sessão de treinamento e é atualizado iterativamente conforme o modelo vê mais exemplos e aprende com esses exemplos. Por isso, é importante entender que, embora inicializemos aleatoriamente os parâmetros do modelo, como pesos e vieses, a esperança é realmente mostrar ao modelo exemplos de entrada suficientes e sua saída correspondente classes. Ao fazer isso, queremos que nosso modelo aprenda com seus erros, buscando as combinações paramétrico ideais de pesos e viés correspondente para as corretas classes de saídas. Observe que quando inicializamos pesos diferentes, o que realmente estamos fazendo é modificar a inclinação de nossa função de ativação.

O gráfico a seguir mostra como diferentes pesos afetam a inclinação de um sigmóide de ativação de função:

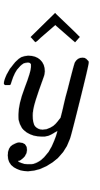
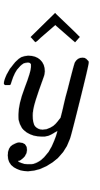
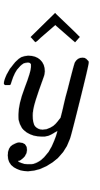


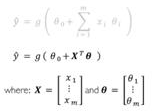
Esperamos essencialmente que, ao mexer com a inclinação de nossa função de ativação, sejamos capaz de capturar idealmente um determinado padrão subjacente em nossos dados. Da mesma forma, quando nós inicializarmos diferentes termos de viés, o que estamos realmente tentando fazer é mudar a função de ativação de maneira ideal (para a esquerda ou para a direita), de modo a acionar a ativação correspondente para configurações específicas de recursos de entrada e saída.

O gráfico a seguir mostra como diferentes termos de viés afetam a posição de um sigmóide de função de ativação:

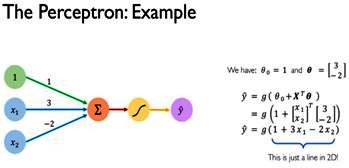


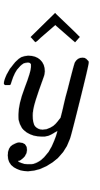
## Saída

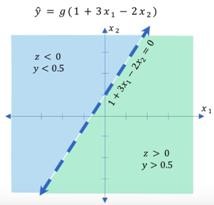
Em nosso modelo de perceptron simples, denotamos a classe de saída atual como *y* e a classe prevista classes de saída como . As classes de saída simplesmente se referem às diferentes classes em nossos dados que estamos tentando prever. Para elaborar, usamos os recursos de entrada ( *x n* ), como temperatura ( *x 1* ) e pressão do ar ( *x 2* ) em um determinado dia, para prever se esse dia específico é ensolarado ou chuvoso um ( ). Podemos então comparar as previsões de nosso modelo com a classe de saída real de naquele dia, denotando se aquele dia estava realmente chuvoso ou ensolarado. Podemos denotar esta simples comparação como ( - *y* ), o que nos permite observar o quanto nosso perceptron perdeu a marca, em média. Mas falaremos mais sobre isso mais tarde. Por enquanto, podemos representar todo a nosso modelo de predição usando tudo que nós temos aprendido até agora, em uma maneira matemática:



O seguindo diagrama exibe um exemplo da Fórmula precedente:

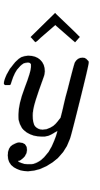


Se nós graficamente traçarmos nossa linha de predição ( ) mostrada anteriormente, conseguiremos visualizar o limite de decisão separando todo o nosso espaço de recursos em dois subespaços. Em essência, traçar uma linha de predição simplesmente nos dá uma ideia do que o modelo aprendeu, ou como o modelo escolhe separar o hiperplano contendo todos os nossos pontos de dados nos vários classes de saída que nos interessam. Na verdade, traçando esta linha, podemos visualizar quão bem nosso modelo se sai simplesmente colocando observações de dias ensolarados e chuvosos neste espaço de recursos e, em seguida, verificando se nosso limite de decisão separa idealmente o classes de saída, como segue:

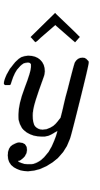


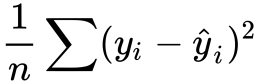
# Aprendizado através erros

Basicamente, tudo o que fazemos com nossos dados de entrada é calcular um produto escalar, adicionar um termo de viés, passá-lo através de uma equação não linear e, em seguida, compare nossa previsão com o valor de saída real, dando um passo na direção da saída real. Esta é a arquitetura geral de um neurônio artificial. Você logo verá como essa estrutura, configurada repetidamente, dá origem a algumas dass mais complexas redes neurais à volta.

Exatamente como convergimos para valores paramétricos ideais dando um passo na direção certa é através de um método conhecido como **propagação reversa de erros** , ou **retropropagação** para abreviar. Mas para propagar erros para trás, precisamos de uma métrica para avaliar o quão bem estamos fazendo em relação ao nosso objetivo. Definimos essa métrica como uma perda e a calculamos usando uma função de perda. Esta função tenta incorporar a diferença residual entre o que o nosso modelo pensa que vê e a realidade real do terreno. Matematicamente falando, isso é mostrado como ( *y* - ). É importante entender aqui que os valores de perda podem realmente ser definidos como uma função dos parâmetros do nosso modelo. Assim, ajustar esses parâmetros nos permite reduzir nossa perda e aproximar nossas previsões dos valores de saída reais. Você verá exatamente o que nós queremos dizer com isso quando nós revermos o completo treinamento processo de nosso perceptron.

## O significar quadrado erro perda função

Uma função de perda usada com destaque é a função **de erro quadrático médio** ( **MSE** ), representada algebricamente na seguinte fórmula. Como você notará, esta função, em sua essência, simplesmente compara o atual modelo de saída ( *y* ) com o modelo de saída previsto( ). Esta função é particularmente útil para avaliarmos nosso poder preditivo, pois essa função modela a perda quadraticamente. Ou seja, se nosso modelo tiver um desempenho ruim e nossos valores de saídas preditas e atuais, tornam-se cada vez mais divergentes, a perda aumenta em um expoente de dois, permitindo nós penalizar os mais alto erros mais severamente:



A média MSE entre valores de saída y e os valores preditos .

Revisitaremos essa noção para entender como reduzimos a diferença entre o que nosso modelo prevê versus a saída real usando vários tipos de funções de perda. Por enquanto, basta saber que a perda do nosso modelo pode ser minimizada através de um processo conhecido como **gradiente descedente**. Como veremos em breve, o gradiente descedente é simplesmente baseado em cálculo e implementado através de algoritmos baseados em retropropagação. O processo de matematicamente reduzir a diferença entre a saída prevista e real, ajustando os parâmetros de uma rede, é na verdade o que faz a rede aprender. Essa sintonia ocorre à medida que treinamos nosso modelo, por mostrá-lo a novos exemplos de entradas e saídas associadas.

# Treinando um perceptron

Até agora, temos uma compreensão clara de como os dados realmente se propagam através do nosso perceptron. Nós também vimos brevemente como os erros do nosso modelo podem ser ropagados para trás. Nós usamos uma função de perda para calcular um valor de perda em cada iteração de treinamento. Este valor de perda nos diz até que ponto nosso do modelo predição mente da verdade fundamental real. Mas e depois?

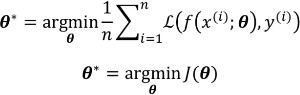
## Quantificando a perda

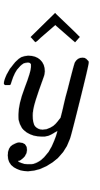
Uma vez que o valor da perda nos dá uma indicação da diferença entre nossa previsão e saídas atuais, é lógico que, se nosso valor de perda for alto, haverá uma grande diferença entre as previsões do nosso modelo e a saída real. Por outro lado, um baixo valor de perda indica que nosso modelo está aproximando a distância entre o previsto e saída atual. Idealmente, queremos que nossa perda converja para zero, o que significa que não há muito diferença entre o que nosso modelo pensa que vê e o que realmente é mostrado. Nós fazemos nossa perda convergir para zero simplesmente usando outro truque matemático, baseado em cálculo.

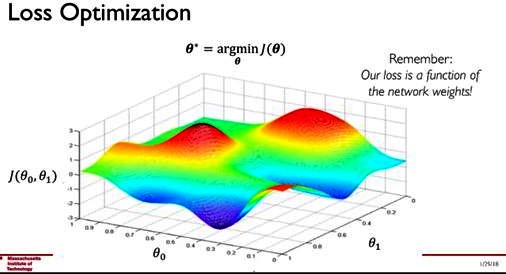
## Perda como a função de modelo pesos

Bem, lembre-se de quando dissemos que também podemos pensar em nosso valor de perda em *função* dos parâmetros do modelo? Considere isto. Nosso valor de perda nos diz a que distância nosso modelo está da previsão real. Essa mesma perda também pode ser redefinida em função do peso do nosso modelo ( θ ). Lembre-se de que esses pesos são o que realmente leva à previsão do nosso modelo em cada iteração de treinamento. Pensando nisso intuitivamente, queremos poder mudar nosso modelo pesos com a respectiva para a perda, então como reduzir nossos erros de predição tanto quanto possível.

Em termos mais matemáticos, queremos minimizar nossa função de perda para atualizar iterativamente os pesos para o nosso modelo e idealmente convergir para os melhores pesos possíveis. estes serão os melhores pesos no sentido de que eles serão capazes de representar melhor os recursos que são preditivos de nossas classes de saída. Este processo é conhecido como **otimização de perdas** e pode ser ilustrado matematicamente da seguinte forma:

Gradiente descendente

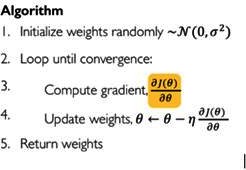
Observe que representamos nossos pesos de modelo ideais ( θ *\** ) como o mínimo de nossa função de perda sobre todo o conjunto de treinamento. Em outras palavras, para todas as entradas de recursos e saídas rotuladas, mostrar nosso modelo, queremos que ele encontre um lugar em nosso espaço de recursos onde a diferença geral entre o real ( *y* ) e previsto ( ) os valores são os menores. O espaço de features que nos referimos são todas as diferentes combinações possíveis de pesos que o modelo pode inicializar. Para o caso de ter uma representação simplificada de nossa função de perda, nós a denotamos como *J* ( θ ). Nós podemos agora resolver iterativamente o mínimo de nossa função de perda, *J* ( θ ), e desçer o hiperplano para convergir a um mínimo global. Este processo é o que nós chamamos de **gradiente descendente** :



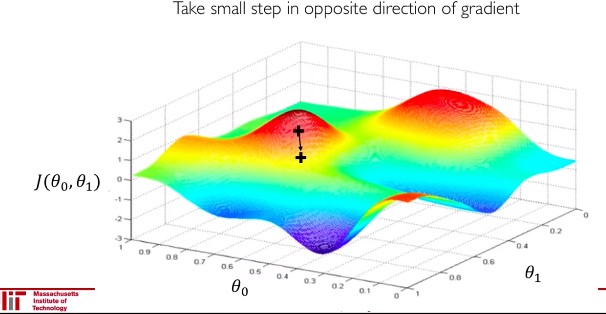
# Retropropagação

Para os mais orientados matematicamente, você deve estar se perguntando como exatamente descemos nosso gradiente iterativamente. Bem, como você saber, nós começamos por inicializar pesos aleatórios para nosso modelos, alimentamos alguns dados, calculamos produtos escalares e os passamos por meio de nossa função de ativação junto com nosso viés para obter uma saída prevista. Usamos essa saída prevista e a saída real para estimar os erros nas representações do nosso modelo, usando a perda função. Agora vem o cálculo. O que podemos fazer agora é diferenciar nossa função perda, *J* ( θ ), em relação aos pesos do nosso modelo ( θ ). Este processo permite-nos essencialmente compare como as mudanças nos pesos do nosso modelo afetam as mudanças na perda do nosso modelo. O resultado dessa diferenciação nos dá o gradiente de nossa função J( θ ) no modelo atual peso ( θ ) junto com a direção da subida mais alta. Por subida mais alta, queremos dizer a direção em qual o diferença entre predição e saída valores parecem mais alto.

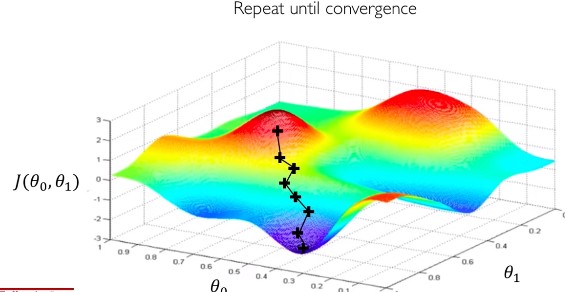
Portanto, simplesmente damos um passo na direção oposta e descemos o gradiente de nossa função perda, *J* ( θ ), com respeito para nosso modelo de pesos ( θ ). Nós apresentamos uma representação algorítmica de esse conceito, em forma de pseudo-código, como segue:



O seguindo gráfico é a visualização do algoritmo gradiente decendente:



Como vemos, o algoritmo do gradiente descendente nos permite descer o hiperplano de perda, até que nosso modelo convirja para alguns parâmetros ótimos. Neste ponto, a diferença entre as previsões do nosso modelo e a realidade será bastante insignificante, e podemos considerar nosso modelo treinado.

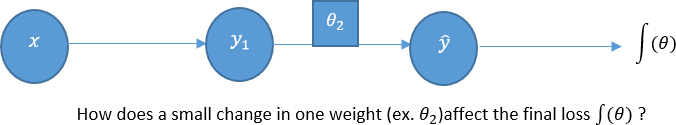


Assim, calculamos as mudanças em nossas redes de pesos em relação às mudanças dos valores gerados pela função de perda (ou seja, os gradientes da rede de pesos). Então nós atualizamos proporcionalmente os pesos da rede, na direção oposta dos gradientes calculados, assim como ajustamos para o erros.

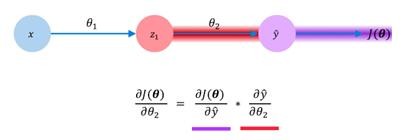
**Computando o gradiente**

Agora que estamos familiarizados com o algoritmo de retropropagação, bem como com a noção de descida do gradiente, podemos abordar questões mais técnicas. Perguntas como, *como nós realmente calculamos este gradiente?* Como você sabe, nosso modelo não tem a liberdade de visualizar o cenário das perdas e escolher um bom caminho de descida. Na verdade, nosso modelo não pode dizer o que está em cima ou o que está em baixo. Tudo o que sabe, e sempre saberá, são números. No entanto, como se verifica, números podem dizer muito.

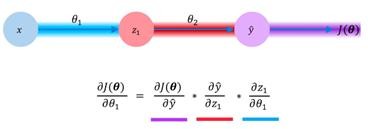
Vamos reconsiderar nosso modelo perceptron simples para ver como podemos retropropagar seus erros por calcular o gradiente da nossa função de perda, *J* ( θ ), iterativamente:



E se quiséssemos ver como as mudanças nos pesos da segunda camada afetam as mudanças em nossa perda? Obedecendo às regras do cálculo, podemos simplesmente diferenciar nossa perda função, *J* ( θ ), em relação aos pesos da segunda camada ( θ *2* ). Matematicamente, podemos realmente representam isso numa maneira diferente também. Usando a regra da cadeia, podemos mostrar como mudanças em nossa perda em relação aos pesos da segunda camada são, na verdade, um produto de dois próprios gradientes diferentes. Um representa as mudanças em nossas perdas em relação a previsão do modelo, e o outro mostra as mudanças na previsão do nosso modelo em relação para o pesos na segunda camada. Isso pode ser representado a seguir:



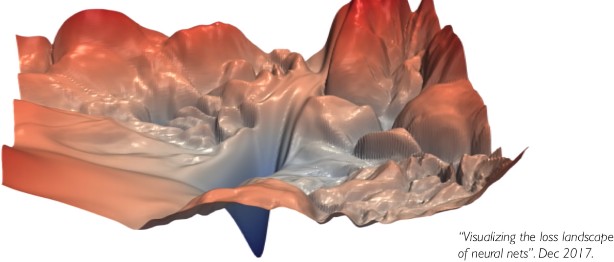
Podemos levar essa recursão ainda mais longe. Digamos que em vez de modelar o impacto dos pesos variáveis da segunda camada ( θ *2* ), queríamos propagar todo o caminho de volta e ver como nossa perda muda em relação aos pesos de nossa primeira camada. Em seguida, simplesmente redefinimos essa equação usando a regra da cadeia, como fizemos anteriormente. De novo, nós estamos interessados em trocar na perda de nosso modelo em respeito as pesos do modelo para nossa primeira camada, ( θ1 ) *.* Nós definir isso usando os produtos de três diferentes gradientes; as mudanças em nossa perda com respeito a saída, mudanças em nossa saída com respeito a nosso valor da camada oculta e, finalmente, as mudanças no valor de nossa camada oculta com respeito as nossas primeiras camadas de pesos. Podemos resumir isso do seguinte modo:



E então, é assim que usamos a função de perda e retropropagamos os erros calculando o gradiente de nossa função de perda em relação a cada peso em nosso modelo. Fazendo isso, nós somos capazes de ajustar o curso de nosso modelo na direção certa, sendo a direção da descida mais alta, como vimos antes. Fazemos isso para todo o nosso conjunto de dados, denotado como uma época. E o que acontece com o tamanho de nosso step? Bem, isso que é determinado pelo taxa de aprendizado que nós montamos.

## A taxa de aprendizado

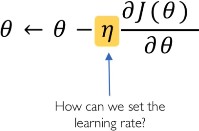
Embora um tanto intuitivo, a taxa de aprendizado de um modelo simplesmente determina o quão rápido ele pode aprender. Colocado matematicamente, a taxa de aprendizado determina o tamanho exato do passo que damos em cada iteração, à medida que descemos ao nosso cenário de perda para convergir aos pesos ideais. Configurar a taxa de aprendizado certa para o seu problema pode ser desafiadora, especialmente quando o cenário de perdas é complexo e cheio de surpresas, como pode ser visto no ilustração aqui:



Esta é uma noção bastante importante. Se definirmos uma taxa de aprendizado muito pequena, então, naturalmente, nosso modelo aprende menos do que potencialmente poderia por qualquer iteração do treinamento. Ainda pior com baixas taxas de aprendizado é quando nosso modelo fica preso em um mínimo local, pensando que atingiu um mínimo global. Por outro lado, uma alta taxa de aprendizagem pode, por outro lado, dissuadir nosso modelo de captura de padrões de valor preditivo.

Se nossos passos forem muito grandes, podemos simplesmente continuar ultrapassando qualquer mínimo global presentes em nosso espaço de pesos característicos e, portanto, nunca convergir para nosso modelo ideal de pesos.

Uma solução para este problema é definir uma taxa de aprendizagem adaptativa, responsiva ao cenário de perda que pode encontrar durante o treinamento.



* 3. **5.7 - Trabalhos Relacionados**

Como trabalho relacionado temos [1], o qual prevê o desempenho acadêmico de alunos do ensino fundamental e médio usando **algoritmos de aprendizado de máquina (MLAs)** com base em vários dados sociodemográficos (como idade, sexo, obesidade, renda familiar média, tamanho da família e estado civil dos pais), variáveis relacionadas à escola (tipo de educação e nível acadêmico) e relacionadas ao aluno (estresse e estilo de vida). O Grade Point Average (GPA), que é um reflexo do desempenho acadêmico, é considerado o resultado do modelo. Cinco MLAs diferentes são considerados para identificar e classificar os parâmetros que afetam o desempenho acadêmico: multinomial logistic regression, artificial neural network, random forest, gradient boosting e métodos de empilhamento. Para avaliar o desempenho dos MLAs, três métricas são utilizadas: precisão, recall e F1-score. Observa-se que o método gradient boosting superou as demais técnicas gerando resultados superiores, seguido pela random forest. A partir da análise do modelo, concluiu-se que um estilo de vida consciente saudável se correlaciona positivamente com o desempenho acadêmico, enquanto a existência de estresse tem um impacto negativo. No entanto, o gênero não é considerado um preditor significativo do desempenho acadêmico de um aluno.

Outro trabalho relacionado [2], no qual os alunos que enfrentaram problemas que podem atrapalhar sua busca acadêmica pelo sucesso, problemas que vão desde questões triviais, como condição de classe, sentimentos do alunos, até questões graves, como desagregação familiar, razões econômicas e muito mais. Este é um grande problema porque os alunos moldam o futuro de uma nação – o que afetará muitas coisas no futuro. Os professores estão procurando uma maneira eficaz de encontrar o que geralmente pode ser a melhor solução para resolver determinados problemas, pois cada aluno pode enfrentar problemas diferentes, resolver um de cada vez não é possível com o número de alunos a cada ano. Neste artigo, tentaremos encontrar fatores que possam prejudicar ou melhorar o desempenho do aluno usando a correlação de Pearson entre cada recurso em relação ao resultado G3 dos alunos. Com base no resultado, as reprovações passadas impactarão negativamente as notas dos alunos com correlação de -0,360415 e, então, a Educação da Mãe impactará positivamente as notas dos alunos com 0,217147. Depois de descobrir qual fator afeta a nota do aluno, tentamos prever a nota do aluno usando modelos de ML para provar se esse fator realmente afeta a nota do aluno. Nosso modelo MLP de 12 neurônios apresenta o melhor desempenho com valor RMSE de 4,32, seguido por **Random Forest** com valor RMSE de 4,52 e, finalmente, Árvore de Decisão com valor RMSE de 5,69.

Ainda outro trabalho relacionado seria [3] no qual encontrar uma solução para o problema da retenção de alunos é uma tarefa frequentemente solicitada em todo o Aprendizado de máquina no Ensino Superior. Na maioria das vezes, gerentes e acadêmicos confiam na intuição e na experiência para identificar a análise de aprendizagem, os alunos e fatores de risco em potencial. Este trabalho examina a literatura em torno dos métodos e medidas atuais em uso no Learning Analytics. Descobrimos que, embora as ferramentas estejam disponíveis, elas não se concentram na identificação mais precoce possível de alunos com dificuldades. Nosso trabalho define uma nova estatística descritiva para a frequência dos alunos e aplica ferramentas e técnicas modernas de aprendizado de máquina para criar um **modelo preditivo**. Demonstra-se como os alunos podem ser identificados já na semana 3 (do semestre de outono) com aproximadamente 97% de precisão. Além disso, situa-se este resultado num contexto pedagógico adequado para apoiar a sua utilização como parte de um mecanismo de apoio ao aluno mais abrangente.

Por fim tem-se [4] onde a pandemia do COVID-19 trouxe mudanças para os indivíduos, principalmente no comportamento do consumidor. Como o governo de diferentes países implementou protocolos de segurança para mitigar a propagação do vírus, as pessoas ficaram apreensivas em viajar e sair. Isso abriu caminho para o surgimento da logística de terceiros (3PL). As estatísticas provaram a rápida escalada em relação ao uso de 3PL em vários países. Este estudo utilizou **Rede Neural Artificial** e Classificador **Randon Forest** para validar e justificar os fatores que afetam a intenção do consumidor em selecionar um provedor de serviços 3PL durante a pandemia de COVID-19 integrando as Dimensões da Qualidade do Serviço e a Teoria Pró-Ambiental do Comportamento Planejado. Os resultados deste estudo revelaram que a atitude é o fator mais significativo que afeta a intenção comportamental dos consumidores. Outros fatores, como satisfação do cliente, valor percebido pelo cliente, preocupação ambiental percebida, garantia, capacidade de resposta, empatia, confiabilidade, tangibilidade, controle comportamental percebido, norma subjetiva e suporte de autoridade percebido, são todos fatores que contribuem para afetar a intenção comportamental. Os algoritmos de aprendizado de máquina, especificamente **ANN e RFC,** mostraram-se confiáveis na **previsão de fatores**, pois obtiveram taxas de precisão de 98,56% e 93%. Os resultados mostraram que a atitude dos consumidores, a satisfação, o valor percebido, a garantia do 3PL e as preocupações ambientais percebidas foram altamente influentes na escolha de uma transportadora de pacotes 3PL. Verificou-se que as pessoas seriam incentivadas a usar os prestadores de serviços 3PL se demonstrassem disponibilidade e preocupação ambiental em atender às necessidades dos clientes. Posteriormente, os provedores de 3PL devem garantir segurança e conveniência antes, durante e depois da prestação do serviço para garantir o patrocínio contínuo dos consumidores. Este é considerado o primeiro estudo que utilizou um ensemble de aprendizado de máquina para medir a intenção comportamental para o setor de logística. A estrutura, as ferramentas de análise e os resultados deste estudo podem ser estendidos e aplicados entre outras intenções comportamentais em relação ao transporte em todo o mundo. Insights gerenciais entre prestadores de serviços são discutidos ali.

Outro trabalho relacionado foi o desenvolvido por Pereira[16], no qual estabelece uma abordagem para **previsão** de precipitação e irradiação solar utilizando modelos de deep learning e na escolha de um local adequado para instalação de um sistema de ultrafiltração de água de chuva utilizando energia fotovoltaica. Esses modelos foram empregados com a finalidade de prever precipitação e irradiação solar para os próximos seis meses, de acordo com o ultimo mês de dados históricos coletado. Modelos lineares e de machine learning foram testados e confrontados com os modelos de deep learning a fim de traçar uma linha de testes e determinar o RMSE de cada modelo. O modelo skillfull escolhido e empregado neste trabalho foi o long short-term memory (LSTM). Esse modelo teve um RMSE de 42,53 para precipitação e 0,45 para irradiação solar. Sendo superior a outros modelos de deep learning, como a MLP e ConvNet. Cálculos de normalização e combinação média, utilizando três pesos (0,7; 1,0 e 1,3) foram utilizados para corroborar com a viabilidade do modelo. Uma aplicação web foi desenvolvida para a apresentação dos resultados. Segundo o experimento realizado, a abordagem apresentou-se adequada, e pode servir como tomada de decisão em cálculos de dimensão de um sistema de energia fotovoltaica, quantidade de água de chuva predita, bem como para o sistema de ultrafiltração de água de chuva.

Um sistema preditivo foi o de Coussement et a [17] no qual segundo ele, o aprendizado on-line foi adotado rapidamente por instituições e organizações educacionais. Apesar de suas muitas vantagens, incluindo acesso 24 horas por dia, 7 dias por semana, alta flexibilidade, conteúdo rico e baixo custo, o aprendizado on-line sofre com altas taxas de abandono que prejudicam os resultados pedagógicos e econômicos dos objetivos. Ferramentas aprimoradas baseadas em assinaturas de **previsão** de abandono de alunos ajudariam os provedores a detectar proativamente os alunos em risco de abandono e identificar fatores de que eles podem abordar para ajudar os alunos a continuar sua experiência de aprendizado. Portanto, este estudo busca melhorar as previsões de evasão escolar, com três contribuições principais. Primeiro, compara um algoritmo de modelo logit leaf (LLM) recentemente proposto com outros oito algoritmos, usando um conjunto de dados da vida real de 10.554 alunos de um provedor global de aprendizado on-line baseado em assinatura. O LLM supera todos os outros métodos ao encontrar um equilíbrio entre desempenho preditivo e compreensibilidade. Em segundo lugar, uma nova visualização informativa multinível do LLM adiciona novos benefícios em relação a uma visualização LLM padrão. Em terceiro lugar, esta pesquisa especifica os impactos dos gráficos de demonstração dos alunos; características da sala de aula; e variáveis de engajamento acadêmico, cognitivo e comportamental na evasão escolar. Ao revisar os segmentos LLM, esses resultados mostram que diferentes percepções surgem para vários segmentos de alunos com diferentes padrões de aprendizagem. Esse resultado notável pode ser usado para personalizar campanhas de retenção de alunos.

Outro foi o [18], onde o projeto consiste, na concepção e implementação de um sistema de orientação dos alunos do núcleo comum para um dos ramos técnicos ou científicos do primeiro bacharelado. O objetivo deste sistema é ter uma boa orientação que permita ao aluno obter uma boa nota de acordo com um modelo já existente que contém o conjunto de alunos que já concluíram o bacharelado na região de Guelmim Ouad Noun. Para resolver o problema discutido antes, nossa solução deve implementar um sistema inteligente que atenda às necessidades dos alunos. Isso exigirá um sistema que preveja a média de bacharelado do aluno por meio de sua ou suas notas principais usando Machine Learning e técnicas de mineração de dados.

A Fig. 8 ilustra nossa abordagem para prever as médias dos alunos, na verdade, os dados de entrada são lidos do MYSQL DBMS

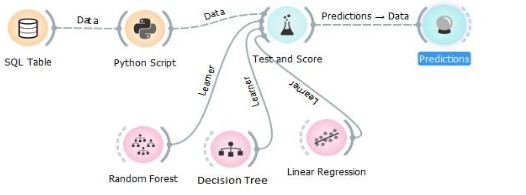


Figura 8 - Abordagem do Projeto[18]

e depois transferidos para um script python que divide os dados em duas partes de treinamento e teste e executa os três algoritmos: floresta aleatória, árvore de decisão e regressão linear e dá ao final a pontuação de cada algoritmo.

**6 - MATERIAIS E MÉTODOS**

Como a pesquisa aborda uma arquitetura que utiliza dados quantitativos da planilha de Registro Disciplinar dos alunos. Posteriormente, objetiva a análise dessas informações. Pode-se classificar o trabalho descrito nesta proposta como uma pesquisa do tipo quantitativa. Já com relação a sua natureza, como neste trabalho a arquitetura proposta objetiva a resolução de um problema prático que é a previsão da possibilidade de perder a residência estudantil usando sistemas preditivos.

Quanto aos objetivos da presente pesquisa, pode-se afirmar que se encaixam entre o tipo de pesquisa descritiva - já que envolve a coleta de dados dos alunos- e o tipo de pesquisa explicativa, que visa a caracterização das causas dos fenômenos encontrados através da pesquisa descritiva. Com relação aos seus procedimentos, a presente pesquisa pode ser classificada como experimental, pois passa por uma etapa de validação dos dados, cálculos em relação ao sistema preditivo. Finalmente, a classificação final da pesquisa pode ser nominada como pesquisa quantitativa-aplicada-explicativa-experimental.

A metodologia utilizada na previsão do risco de perder a residência estudantil faz uso de técnicas de machine learning, utilizando modelos de árvores de decisão.

O KNN (K-nearest neighbors, ou “K-vizinhos mais próximos”) costuma ser um dos primeiros algoritmos aprendidos por iniciantes no mundo do aprendizado de máquina.

Em resumo, o KNN tenta classificar cada amostra de um conjunto de dados avaliando sua distância em relação aos vizinhos mais próximos. Se os vizinhos mais próximos forem majoritariamente de uma classe, a amostra em questão será classificada nesta categoria.

O Algoritmo Naive Bayes funciona como classificador e baseia-se na probabilidade de cada evento ocorrer, desconsiderando a correlação entre features. Por ter uma parte matemática relativamente simples, possui um bom desempenho e precisa de poucas observações para ter uma boa acurácia. Uma aplicação bastante comum é para identificar se um determinado e-mail é um spam ou não.

Uma árvore de decisão é um algoritmo de aprendizado de máquina supervisionado que é utilizado para classificação e para regressão. Isto é, pode ser usado para prever categorias discretas (sim ou não, por exemplo) e para prever valores numéricos (o valor do lucro em reais).

A árvore de decisão estabelece nós(decision nodes) que se relacionam entre si por uma hierarquia. Existe o nó-raiz(root node), que é o mais importante, e os nós-folha(leaf nodes), que são os resultados finais. No contexto de machine learning, o raiz é um dos atributos da base de dados e o nó-folha é a classe ou o valor que será gerado como resposta.

A figura 7 apresenta o fluxo metodológico no qual descreve os passos a serem

seguidos.

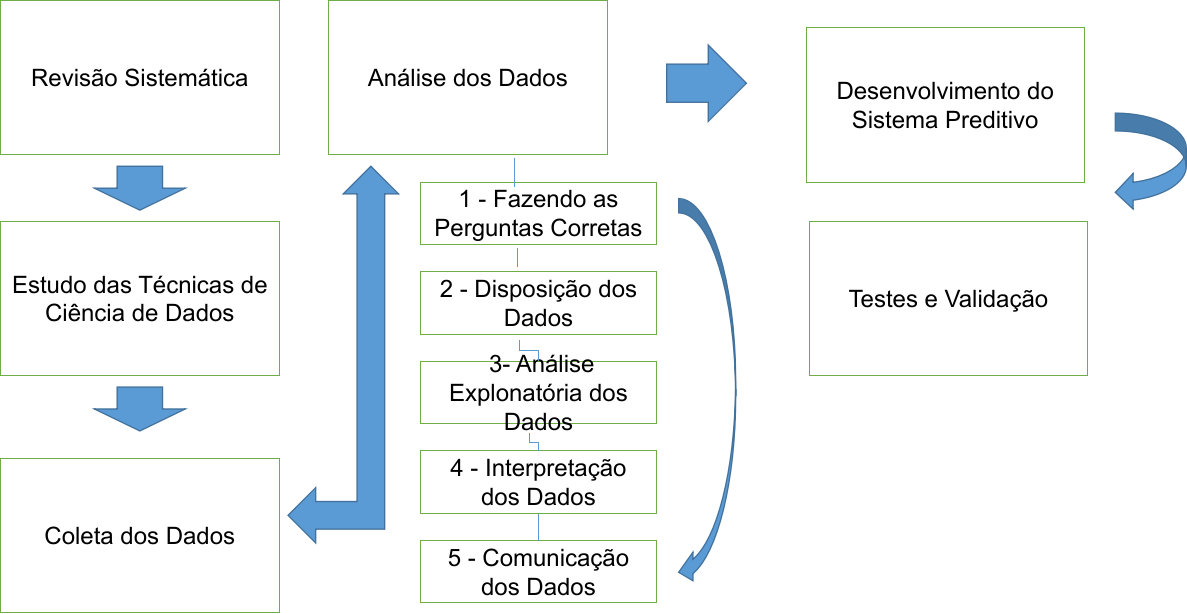


Figura 7 - Fluxo Metodológico(Autoria Própria)

**6.1 - Estudo e Escolha das Técnicas**

O estudo deu início com uma revisão sistemática, onde, cumprido todas as etapas, foram escolhidos 15 trabalhos com critérios pré-definidos na RS. Também teve como base a busca por estudos que fizessem uso de técnicas para ciência de dados como base para esse projeto.

As técnicas e ou modelos mais utilizados nestes estudos foram divididas em 3, em machine learning, a escolha de árvores de decisão se destaca pela sua estrutura visual (decision tree).

Isso se dá porque a composição desse algoritmo é similar a de um fluxograma, com etapas muito fáceis de visualizar e entender. Assim, além de uma boa técnica com bons resultados, é também ótima abordagem para uma compreensão melhor de como funciona o aprendizado de máquina.

Talvez ele seja o método mais conhecido de ML, se não considerarmos as redes neurais. Mas mesmo que redes neurais sejam conhecidas, poucas pessoas realmente sabem como elas funcionam. Uma árvore pega emprestada uma noção de tomada de decisões que está muito presente em outras áreas, por isso, é mais acessível, o que pode ser observado em [1] e [18].

**6.2 Coleta a análise dos Dados**

A coleta de dados consiste em seguir uma série de etapas capaz de adquirir os dados em planilhas e posteriormente serem analisados utilizando dos processos que fazem parte das boas práticas da análise de dados. A análise de dados, também conhecida como análise estatística e mineração de dados, busca o entendimento do processo de análise, pela capacidade de diferenciar entre essas diferentes atividades [19] [20].

Os arquivos estão no formato .xlsx do Excel e podem ser convertidos no formato CSV, se necessário.

O conhecimento do conjunto de dados é necessário antes de realizar qualquer análise, pois, primeiro, isto fornece uma breve compreensão e, segundo, impedirá interpretações errôneas ao decorrer da análise.

**6.2.1 - Limpeza dos Dados**

A limpeza dos dados significa o processo de identificar a parte incorreta, incompleta, inexata, irrelevante ou ausente dos dados e depois modificá-los, substituí-los ou excluí-los de acordo com a necessidade. A limpeza de dados é considerada um elemento fundamental da ciência básica dos dados.

Para esse processo foi utilizado alguns scripts na linguagem python, o que torna o processo mais rápido e seguro. Toda a análise e limpeza dos dados é de extrema importância para o processo como um todo. Um pequeno erro no conjunto de dados pode lhe causar muitos problemas e todos os esforços podem ser desperdiçados.

**6.3 - Desenvolvimento da Aplicação**

A aplicação será desenvolvida em python com uso das bibliotecas pandas, scikit-learn e django usados no Python. Testamos três algoritmos: KNN, Naive-Bayes e de árvore de decisão .No final, avaliamos qual algoritmo foi mais eficiente.

**6.3.1 - Criação dos Alunos:**

Para a criação de algum aluno deve ser clicado no botão de criação, localizado no menu lateral esquerdo, com os seguintes dados: Matrícula; Nome; Gênero; Data de nascimento; Curso, os valores de KNN; Naive Bayes e Árvore de Decisão devem permanecer o para poder ocorrer o cálculo correto dos valores.

**6.3.2 - Criação das Normas no sistema:**

Para a criação de alguma norma deve ser clicado no botão de criação, localizado no menu lateral esquerdo, com os seguintes dados: Regulamento e Descrição, sendo o campo “Regulamento” sendo tratado como um título da Norma, a qual irá aparecer quando for citar ela nas Violações.

**6.3.3 - Criação das Medidas Disciplinares no Sistema:**

Para a criação de alguma Medida Disciplinar deve ser clicado no botão de criação, localizado no menu lateral esquerdo, com os seguintes dados: Descrição e Classificação, sendo essa podendo considerar ser Leve, Média e Grave.

Por conta do Modelo padrão, foram atribuídos os valores 1, 2 e 3 respectivamente.

**6.3.4 - Criação de Violações do sistema:**

Para a criação de alguma Violação deve ser clicado no botão de criação, localizado no menu lateral esquerdo, com os seguintes dados: Estudante; Infração; Descrição da Ocorrência; Medida Disciplinar; Complemento da Medida e Data do Evento.

No campo de “Infração”, aparecerá as opções criadas na “Normas”.

No campo “Complemento da Medida” pode ser utilizado para justificar melhor a escolha da Medida disciplinar do Aluno.

Após criar uma violação, ou um conjunto de violações, é necessário clicar no botão “Calcular Riscos” para poder calcular os riscos de todos os alunos. O cálculo é feito de forma automática, e será levada a uma página de confirmação, na qual apenas basta voltar para a página anterior (Normalmente utilizando os botões de “retroceder a página” do próprio navegador.

Todos os botões de criação estão presentes nas páginas de Observação das listas criadas, sejam elas “Aluno”, “Normas”, “Medidas Disciplinares” e “Violações”.

**6.3.5 - Como é feito o cálculo de riscos:**

Tanto para o KNN quanto para Naive Bayes utiliza uma função de Aprendizado de Máquina que faz o computador “aprender” com a tabela de violações bases fornecidas, e com isso gera a previsão pro aluno caso ele for perder ou não a residência estudantil.

* 1. **6.4 - Ferramentas** 
     1. **6.4.1 - Django**

Django é um framework web Python de alto nível que permite o rápido desenvolvimento de sites seguros e de fácil manutenção. Construído por desenvolvedores experientes, o Django cuida de grande parte do trabalho de desenvolvimento web, para que você possa se concentrar em escrever seu aplicativo sem precisar reinventar a roda. É gratuito e de código aberto, tem uma comunidade próspera e ativa, ótima documentação e muitas opções de suporte gratuito e pago.[21]

* + 1. **6.4.2 - Linguagem de programação Python**

Python é uma linguagem interpretada, que permite uma programação rápida além da integração com inúmeras bibliotecas e ferramentas. Ela possibilita um fácil aprendizado para novos ou experientes programadores. A linguagem possui estruturas de dados de alto nível e uma abordagem simples, mas eficaz, para programação orientada a objetos

Por ser uma linguagem com código aberto, ela possibilita a utilização e distribuição de aplicações, além de permitir uma vasta portabilidade, podendo ser usada em diversas plataformas de programação. Além disso, a vasta comunidade de programadores Python possibilita a existência de muitos tutoriais e materiais de apoio disponíveis gratuitamente.

* + 2. **6.4.3 - Banco de Dados MariaDB**

De acordo com MARIADB PLATFORM[5], MariaDB é um projeto desenvolvido pela comunidade open source. O Programa Monty Ab é o principal administrador do projeto, e a maioria de seus funcionários fizeram parte do núcleo de desenvolvedores originais do MySQL. No entanto, qualquer um pode participar de seu desenvolvimento.

MariaDB Platform é a solução corporativa completa de banco de dados de código aberto. Ele tem versatilidade para suportar cargas de trabalho transacionais, analíticas e híbridas, bem como modelos de dados relacionais, JSON e híbridos. E tem a escalabilidade para crescer de bancos de dados independentes e data warehouses para SQL totalmente distribuído para executar milhões de transações por segundo e realizar análises ad hoc interativas em bilhões de linhas. Considerando tais características, optou-se por utilizar o MariaDB como solução de persistência do projeto SYSDAE.

**7 - Referências**

[1] Rajendran, Suchithra; Chamundeswari, S ; Sinha, Akhouri Amitanand.

Predicting the academic performance of middle- and high-school students using machine learning algorithms. 2022. Disponível em: <<https://doi.org/10.1016/j.ssaho.2022.100357>>. Acesso em 30/03/2023.

[2] Beckham, Nicholas Robert; Akeh, Limas Jaya ; Mitaart , Giodio Nathanael Pratama ; Moniaga, Jurike V. Determining factors that affect student performance using various machine learning methods. 2022.

[3] Gray, Cameron C. ; Perkins, Dave . Utilizing early engagement and machine learning to predict student outcomes. 2018. Diponível em: <<https://doi.org/10.1016/j.compedu.2018.12.006>>. Acesso em 30/03/2023.

[4] German, Josephine D.; Ong, Ardvin Kester S.; Redi, Anak Agung Ngurah Perwira; Robas, Kirstien Paola E. . Predicting factors affecting the intention to use a 3PL during the COVID-19 pandemic: A machine learning ensemble approach. 2022. Disponível em

<<https://doi.org/10.1016/j.heliyon.2022.e11382>>. Acesso em 30/03/2023.

[5] Mariadb. 2022. [www.Mariadb.com](http://www.mariadb.com). Acesso em 30/03/2023.

[6] Machine Learning - IBM. 2023. <https://www.ibm.com/br-pt/cloud/learn/machine-learning>. Acesso em 30/03/2023.

[7] What is Machine Learning - Berkeley. 2023. <https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/>. Acesso em 30/03/2023.

[8] Amazon scraps secret AI recruiting tool that showed bias against women. 2023. <https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G>. Acesso em 30/03/2023.

[9] The Legal and Ethical Implications of Using AI in Hiring. 2023. <https://hbr.org/2019/04/the-legal-and-ethical-implications-of-using-ai-in-hiring>. Acesso em 30/03/2023.

[10] The Ethics of AI Ethics - An Evaluation of Guidelines. Disponível em: <<https://arxiv.org/ftp/arxiv/papers/1903/1903.03425.pdf>>. Acesso em 30/03/2023.

[11] Bonnin, Rodolfo. Machine Learning for Developers : Your One-stop Guide to Becoming a Machine Learning Expert. 2017. Packt Publishing.

[12] Campesato, Oswald. Python 3 For Machine Learning. 2020. Packt Publishing.

[13] Netto, Amílcar; Maciel, Francisco. Python para Data Science e Machine Learning Descomplicado. 2021. Alta Books.

[14] Winters, Ralph. Practical Predictive Analytics. 2018. 2ª. Edition. Packt Publishing.

[15] Rezaul, Karim, Predictive Analytics with Tensor Flow. 2018. 2ª. Edition. Packt Publishing.

[16] Pereira, Elias Morais. Uma abordagem para Identificar Viabilidade de um Local para implantação de Ultrafiltração de Água de Chuva com Utilização de Deep Learning. 2021.

[17] Coussement, Kristof; Phana, Minh; De Caigny, Arno; F. Benoit , Dries; Raes, Annelies. Predicting student dropout in subscription-based online learning environments: The beneficial impact of the logit leaf model. 2020.

[18] Tarik, Ahajjam; Aissa,Haidar; Yousef, Farhaoui. Artificial Intelligence and Machine Learning to Predict Student - Performance during the COVID-19. 2021.

[19] Pickell, D. 5 Steps of the Data Analysis Process. 2019. https://learn.g2.com/

data-analysis-process. Acesso em: 08/05/2023

[20] Gohrani, K. The 5 Steps of the Data Analysis Process - Medium. 2019. https://

medium.com/@kunal\_gohrani/the-5-steps-of-the-data-analysis-process-2512ba6ac31e.

Acesso em: 08/05/2023

1. Introdução ao Django . Disponível em <https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Django/Introduction>. Acesso em 26/06/2023.
2. <https://www.lloyds.com/lloyds/about-us/history/corporate-history.> Acessso em 26/04/2023.
3. Loy, James. Neural Network Projects with Python. 2019. Packt Publishing Ltd.
4. Purkait, Niloy. Hands on Neural Networks with Keras. 2019. Packt Publishing Ltd.