Classificador para predição de aluno evasor de graduação em universidades particulares

Crhistopher Lenhard[1], **Mirkos Ortiz Martins**[1]

[1]Ciência da Computação – Universidade Franciscana (UFN)
Caixa Postal 151 – 97.010-032 – Santa Maria – RS – Brazil

crhistopher.lenhard, mirkos@ufn.edu.br

Abstract. *Higher education dropout is a problem that is difficult to identify, because it involves a large set of characteristics that define it. This paper aimed to build a computational system, based on machine learning, for the identification of the most relevant characteristics and the consequent classification of students between evaders and non-evaders. For this, the Python language was used, together with the Pandas, Numpy and Scikit-learn libraries, resulting in an implementation that compared the decision tree, random forest and extra trees architectures, demonstrating that the latter obtains better results (92.84 %) in accuracy of the identification of the evasive student, in a database with information from the 10-year period at UFN.*

Resumo. *A evasão no ensino superior é um problema de difícil identificação, pois envolve um grande conjunto de características que a define. Esse trabalho teve como objetivo construir um sistema computacional, baseado em machine learning, para a identificação das características mais relevantes e consequente classificação dos alunos entre evasores e não evasores. Para isso foi utilizada a linguagem Python, juntamente com as bibliotecas Pandas, Numpy e Scikit-learn, resultando em uma implementação que comparou as arquiteturas árvore de decisão, floresta aleatória e árvores extra, demonstrando que a última obtém melhores resultados (92.84%) na acurácia da identificação do aluno evasor, em uma base de dados com informações no período de 10 anos na UFN.*

1. Introdução

As instituições de Ensino Superior (IES), particulares e públicas enfrentam um problema comum, a evasão escolar. Estudos e análises são elaborados tendo o objetivo de diminuir a desistência de alunos evasores, e esses estudos se dividem em duas frentes: O desenvolvimento de campanhas para manter os alunos que possuem a probabilidade de evadir; E encontrar os padrões de perfil que alunos evasores possuem para melhorar foco das campanhas [Tinto 1999].

Na avaliação da provável desistência de um aluno, é preciso considerar e encarar de forma mais individualizada o sucesso acadêmico, medido na forma de andamento do curso, objetivando atingir métricas mais concretas na interação IES - aluno e assim definir um formato mais dinâmico para o acompanhamento de sua vida acadêmica possibilitando a preempção de provável desistência do respectivo curso [Silva Filho et al. 2007]. Também é um complicador desse contexto de desistência, o perfil de cada aluno confrontado com as características particulares de cada curso, onde o casamento entre os diferentes atores nem sempre é possível de um sucesso [Dekker et al. 2009].

No contexto de solução desse problema, surge a implementação de uma ferramenta de *data science* e *machine learning* [Grus 2016] para prever um provável evasor, antes que ele realmente desista do curso, assim auxiliando a gestão da IES o desenvolvimento de estratégias para retensão desses alunos. Um passo importante para a implantação de um sistema indicador de evasão é a identificação das regras que regem o comportamento de desistência do aluno, analisados sob a ótica de dados armazenados em informações acadêmicas.

1.1. Justificativa

A evasão discente é uma mazela presente em todas as IES, sendo ela privado ou público. Essas IES buscam estratégias de manutenção do número de alunos matriculados, diminuição do fechamento de turmas ou mesmo turmas com baixo número de alunos. Com isso, um sistema de preempção (previsão) de evasão discente possui extrema importância para a manutenção da estrutura funcional e econômica da IES.

Nesse contexto, esse trabalho pretende levantar uma série de regras de identificação de dados para serem utilizados como entrada em uma ferramenta de *data science*, estado da arte na computação científica, para a descoberta de padrões nos diversos dados em uma base de informações acadêmicas.

1.2. Objetivo

O objetivo geral desse trabalho é a construção de uma ferramenta de classificação no âmbito da identificação de evasão discente, baseada na arquitetura de *machine learning* com a extração primária de informações da base de dados se utilizando de técnicas de *data science*.

Os objetivos específicos desse trabalho são:

- Normalizar uma base de dados acadêmica utilizando *Pandas* e *Numpy*;
- Identificar padrões de dados em discentes evasores;
- Escolher e comparar classificadores de dados dos discentes;
- Implementar os classificadores utilizando a biblioteca *Scikit-learn*;
- Salvar os modelos dos classificadores em arquivos binários a partir da biblioteca *Pickle*;
- Descrever as estatísticas dos resultados alcançados;
- Criar gráficos para apresentação dos resultados fazendo uso das bibliotecas *Plotly* e *Seaborn*;
- Construir uma interface para visualização dos resultados utilizando *Django*.

2. Referencial teórico

Nesta seção serão abordados conceitos e tecnologias relacionadas ao desenvolvimento deste trabalho.

2.1. Evasão Escolar

Um problema presente em todas as instituições de ensino é a da evasão escolar, que afeta desde o ensino fundamental até o superior e constroem um desperdício social, acadêmico e econômico. Nas IES a evasão é evidente internacionalmente, tanto no setor público

quanto no privado. No setor público se têm gastos públicos sem obter a resposta esperada, já no setor privado se apresenta como uma perda de lucro [Silva Filho et al. 2007].

No Brasil, o índice de evasão nas IES privadas aproximou-se de 53% e de 33% nas IES públicas de acordo com um estudo realizado pelo Instituto Nacional de Estudos e Pesquisas Educacionais (INEP) no ano de 2006 [Savian et al. 2018]. No ano de 2010 segundo dados do Censo da Educação Superior, mostra que 53% dos alunos que ingressaram em IES privadas desistiram no decorrer do curso, nas IES públicas a evasão chegou a 47% nas municipais, 38% nas estaduais e 43% nas federais [Oliveira et al. 2019].

A procura das causas da evasão é o tema recorrente em muitos trabalhos e pesquisas educacionais, e de acordo com [Tinto 1999], as IES que adotam um programa para obter uma redução na evasão escolar utilizam de dados ordinários da vida institucional dos alunos. Ainda para [Tinto 1999], a evasão é algo a ser levado a sério, onde que a retenção dos alunos devem ser unânime e o primeiro ano de faculdade deve ser um ano de inclusão que "promova o ideal importante de que todas as pessoas possam e devem ter voz na construção do conhecimento".

2.2. Inteligência Artificial

A Inteligência Artificial (IA) é um campo de estudo da computação que abrange uma enorme variedade de subcampos, onde tenta não somente compreender entidades inteligentes, mas também contruí-las. Suas definições referem-se à processos de pensamento e raciocínio ou de comportamento que buscam pensar e agir como humanos ou racionalmente [Peter Norvig 2013].

O aprendizado de máquina (ML, *Machine Learning*) é uma subárea da IA que estuda o desenvolvimento de técnicas para construção e aprendizado automático. Os algoritmos de ML possuem uma classificação de acordo com à linguagem de descrição, modo, paradigma e forma de aprendizado utilizado [Monard and Baranauskas 2003].

O aprendizado indutivo é o topo da hierarquia, sendo ela a forma de inferência lógica um dos principais métodos para derivar conhecimento. O aprendizado indutivo se divide em *supervisionado* e *não-supervisionado*, no primeiro é dado ao algoritmo uma base de conhecimento onde se conhece o estado final e treina para determinar o estado final dos quais não se tem o conhecimento, separados posteriormente em *classificação* para os de saídas categóricas e como *regressão* os de saída numéricas. Já no aprendizado *não-supervisionado* o algoritmo tenta formar agrupamentos a partir de seu próprio conhecimento, essa determinação de agrupamentos devem ser validadas [Monard and Baranauskas 2003].

A ciência de dados (*Data Science*) é uma área fortemente ligada ao ML pois é a arte de extrair conhecimento de dados desorganizados para detecção de padrões e tomada de decisões [Grus 2016].

2.2.1. Árvore de Decisão

Árvore de decisão é um classificador estruturado no formato de uma árvore binária e apresenta possíveis caminhos de decisão e resultado para cada caminho. São muito recomendadas pois são fáceis de entender, interpretar e acompanhar a trajetória para uma

previsão, além de poder trabalhar com atributos numéricos e categóricos [Grus 2016].

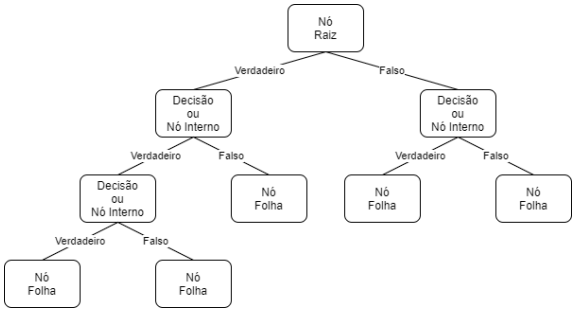

Figura 1. Modelo genérico de uma árvore de decisão. Fonte: Autor

A Figura 1 apresenta um modelo genérico do funcionamento de uma árvore de decisão. Podemos observar que sua estrutura consiste em três tipos de nós. O *Nó Raiz*, presente no início da árvore, representa a aplicação de um teste que pode resultar em *Verdadeiro* ou *Falso*. O caso em teste passa pelos *Nós Internos*, que também os classificam através de testes até chegar em um *Nó Folha*, onde está a classificação encontrada pela árvore no final de seu ramo. [Peter Norvig 2013].

Para [Peter Norvig 2013] o formato da árvore de decisão atinge uma resultado agradável e conciso, mas podem ser ruins para alguns tipos de funções, como por exemplo, a função da maioria, que exige uma árvore extremamente grande para tomar uma decisão de verdadeiro, pois se e somente se as entradas possuírem mais da metade convergindo.

2.2.2. Floresta aleatória

Quando uma árvore de decisão é construída ela é diferente de outra árvore que utilizou dados diferentes ou uma ordem diferente dos dados para ser criada, pois se ajustam com seus dados em seu treinamento. Uma floresta aleatória (*Random Forest*) permite com que possamos construir várias árvores de decisão e deixar com que decidam como classificar sua entrada, tornando um dos modelos mais versáteis disponíveis e assim diminuindo sua variância. [Grus 2016].

A estrutura de uma floresta aleatória é formada por uma coleção de árvores de decisão, como podemos observar na Figura 2, onde são distribuídas cópias de um mesmo vetor para cada árvore da estrutura, e cada árvore retorna uma decisão que é passada para uma classificação final por votação da maioria. O crescimento do conjunto de árvores e a acurácia de suas decisões permitem melhorias significativas na precisão da floresta, fazendo com que seu erro se aproxime de um limite [Breiman 2001].

2.2.3. Árvores Extra

Árvores Extra ou Árvores Extremamente Randomizadas (*Extra Trees*), assim como a floresta aleatória, cria uma coleção de árvores de decisão e também toma a decisão com uma votação majoritária. Porém, diferencia-se da floresta aleatória ao não utilizar uma amostra

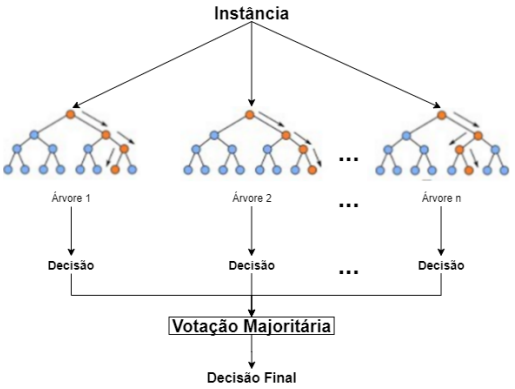

Figura 2. Modelo simplificado da lógica da Floresta Aleatória. Fonte: Adaptado de [Lorenzett and Telöcken 2016]

inicial igual para todas as árvores, e a divisão nos cortes para os nós é realizado de forma aleatória, enquanto na floresta aleatória é realizado a divisão ideal. Essas diferenças fazem com que a árvores extra possuem uma redução no viés e uma menor variância em relação a floresta aleatória [Geurts et al. 2006].

2.3. Validação dos classificadores

Para avaliar o desempenho de um classificador binário é necessário separar uma porção dos dados para testes e validações. Métricas de validação buscam avaliar seu classificador, e existem alguns modelos que fazem isso.

Matriz de Confusão — É utilizada para alcançar a quantidade de predições certas e erradas do modelo. A Figura 3 representa o modelo da matriz, onde que *verdadeiro positivo* (VP) para casos previstos corretamente, *falso positivo* (FP) para casos onde foi classificado como algo que não é. O *falso negativo* (FN) é uma classificação oposta ao FP e *verdadeiro negativo* (VN) para casos classificados certos também, porém com a classificação oposta de VP [Amidi and Amidi 2020].

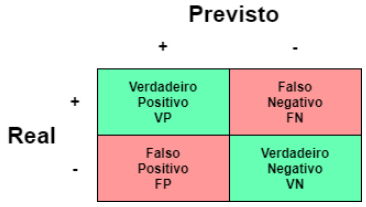

Figura 3. Representação do modelo conceitual de uma Matriz de Confusão. Fonte: Adaptado de [Amidi and Amidi 2020]

É possível, a partir da matriz de confusão, obter as seguintes métricas: Acurácia, Precisão, Sensibilidade, Especificidade e F1 score, descritos na Tabela 1.

ROC e AUC — Chamado de *Receiver Operating Characteristic* e *Area Under the Curve*, são métricas usadas para medição de performance de classificadores binários, onde é plotada a Sensibilidade pela Especificidade em um gráfico para visualizar a curva

Tabela 1. Métricas para avaliar desempenho de modelos [Amidi and Amidi 2020]

Métrica	Descrição	Fórmula
Acurácia	Refere-se ao desempenho geral do modelo	$\frac{VP+VN}{VP+VN+FP+FN}$
Precisão	Refere-se à precisão das predições positivas	$\frac{VP}{VP+FP}$
Sensibilidade	Refere-se à amostra positiva real	$\frac{VP}{VP+FN}$
Especificidade	Refere-se à amostra negativa real	$\frac{VN}{VN+FP}$
F1 score	Refere-se à métrica híbrida de precisão e sensibilidade para classes desequilibradas	$\frac{2VP}{2VP+FP+FN}$

ROC. O AUC representa grau ou medida de separabilidade da curva, descrito de 0 à 1, onde 1 para um modelo que separa o conjunto de dados positivos dos negativos sem erro, e 0 o menor valor, no qual o modelo erra todas as classificações dos dados [Amidi and Amidi 2020].

K-Fold cross validation — É um método de validação cruzada que busca dividir o conjunto de dados em k subconjuntos, onde uma das partes é usada para a validação e é repetido k vezes. Cada vez que um subconjunto é selecionado para ser a parte de validação, os outros $k-1$ subconjuntos são usados para o treinamento. O Treino é feito até todas as k partes terem sido utilizadas como validadores, como pode ser visto na Figura 4, e após isso, a média do erro ou o desvio padrão de todas as k tentativas é calculado [Amidi and Amidi 2020].

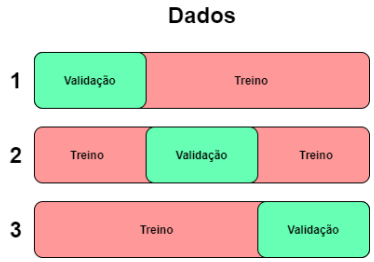

Figura 4. Representação da divisão feita na validação cruzada. Fonte: Adaptado de [Amidi and Amidi 2020]

2.4. Tecnologias utilizadas

Nesta seção serão abordadas tecnologias utilizadas para elaboração deste trabalho.

- **Python:** Python é uma linguagem orientada a objetos de alto nível, que apresenta tipagem dinâmica e forte, além de interpretada e interativa. É presente na área de *data science* devido a simplicidade da escrita, pela quantidade de bibliotecas disponíveis para o tratamento de dados, além de bibliotecas para criação de classificadores [Grus 2016].
- **Scikit-learn:** *Scikit-learn* é uma biblioteca do Python que é utilizada para o aprendizado de máquina, pela eficiência para análise preditiva de dados, onde possui inúmeros algoritmos para o aprendizado supervisionado e não supervisionado [Pedregosa et al. 2011].

- **Pandas:** *Pandas* é uma biblioteca do Python, que é utilizada para a análise e manipulação de dados em alta performance, pois contém o objeto *DataFrame* que possui indexação integrada rápida e eficiente [Grus 2016].
- **NumPy:** *NumPy* é uma biblioteca do Python, que é utilizada principalmente para realizar operações em *Arrays* e matrizes multidimensionais, pois apresenta funções pré-compiladas, com grande capacidade de processamento numérico [Grus 2016].
- **Django:** *Django* é um *framework* para desenvolvimento web de código aberto em Python, que utiliza o padrão *model-template-view* em seu funcionamento, focado para o desenvolvimento rápido, ágil e limpo do projeto [Forcier et al. 2008].
- **Plotly:** *Plotly* é uma plataforma colaborativa de gráficos e análises baseada na web. Permite a criação de gráficos iterativos com fácil implementação [Inc. 2015].
- **Seaborn:** *Seaborn* é uma biblioteca do Python, de visualização de dados baseada na biblioteca *matplotlib*. Fornece um alto nível para desenhar gráficos estatísticos [Waskom et al. 2017].
- **Pickle:** *Pickle* é um módulo da biblioteca padrão do Python, que realiza a escrita e leitura de arquivos binários que possuem a estrutura de um objeto Python. O que este módulo faz é serializar a estrutura em um fluxo de *bytes*, e o reconstrói a partir do arquivo binário dele [Python Software Foundation 2020].
- **PEP 8:** O PEP 8 é uma proposta de aprimoramento do Python com a intenção de manter a consistência nos códigos. O PEP 8 possui propostas de guia de estilo para: Formatação do código; Comentários; *Docstrings*; Controle de versão; Nomes e identificadores; e Recomendações ao programar [Van Rossum et al. 2001].

3. Trabalhos correlatos

Nesta seção serão abordados trabalhos correlatos de problemas semelhantes de evasão que possuíram em suas soluções a utilização de técnicas computacionais.

3.1. Aprendizado de Máquina Aplicado à Análise de Evasão no Ensino Superior

Pinheiro et al (2018) aborda o problema da evasão discente em uma universidade de ensino superior, onde sua proposta de solução utiliza aprendizado de máquina juntamente com os algoritmos *Naive Bayes*, Árvore de Decisão e *Support Vector Machines*, implementados em linguagem de programação R para identificar qual perfil um aluno evasor possui. Os dados utilizados neste trabalho são referentes à 21 anos, do período de 1992 a 2013, oriundos do Sistema Acadêmico do Instituto Federal de Educação Ciência e Tecnologia do Maranhão. Ao aplicar os algoritmos na base de dados obteve-se o resultado de que o *Support Vector Machines* atingiu 98,76% de acurácia, 99,65% de especificidade, 98,38% de sensibilidade e 99,85% de precisão, sendo estes valores maiores dos que os obtidos nos algoritmos de *Naive Bayes* e Árvore de Decisão. Concluiu-se que a utilização de aprendizado de máquina é adequado para prever casos de evasão.

3.2. *Predicting Students Drop Out: A Case Study*

Dekker et al (2009) traz em seu trabalho a mineração de dados educacionais como solução para prever a evasão no curso de Engenharia Elétrica da Universidade de Tecnologia de Eindhoven. Sua pesquisa considerou dados pré-universitários e do primeiro ano de faculdade de 648 alunos do período de 2000 a 2009. Os algoritmos CART (*SimpleCart*),

C4.5 (J48), *BayesNet*, *SimpleLogistic*, *JRip*, *RandomForest* e *OneR* foram usados para realizar uma classificação de aprendizado supervisionado. A acurácia dos algoritmos *SimpleLogistic*, *RandomForest* e *SimpleCart* chegou a 79% e o JRip com 77% . Os piores resultados foram obtidos pelos algoritmos *OneR* e *BayesNet* com 75%, já o J48 teve o melhor resultado com 80% de acerto. Conclui-se que classificadores simples e intuitivos alcançam um resultado útil com precisão de 75% a 80%, e que o aprendizado e análise dos dados são ferramentas importantes para previsão de evasão escolar.

3.3. Identificando o perfil de evasão de alunos de graduação através da Mineração de dados Educacionais: um estudo de caso de uma Universidade Comunitária

Paz e Cazella (2017) apresenta em seu trabalho a busca de perfis de alunos que possuem potencial de evasão, e utilizam para isso a mineração de dados aplicada em um banco de dados de uma universidade comunitária. Sua pesquisa utilizou dados de 12 tabelas, contendo 322 atributos referentes à 4697 alunos. No pré-processamento os dados foram convertidos para arquivo CSV e operados pela ferramenta WEKA (*Waikato Environment for Knowledge Analysis*). Os dados foram classificados através do algoritmo J48, e obtiveram acurácia de 91,25% em seu experimento A (Alegrete e Bagé), 91,42% de acertos em seu experimento B (Bagé) e 92,01% utilizando o experimento B (Alegrete). Conclui-se que o incentivo e o currículo dos alunos estão ligados à propensão de evasão.

3.4. Conclusão dos trabalhos correlatos

Os trabalhos correlatos apresentados assemelham-se a este trabalho, pois buscam identificar o perfil de evasão nos alunos do ensino superior, utilizando aprendizado de máquina e dados históricos dos estudantes. Em contrapartida, o presente trabalho distingue-se destes, pois utiliza dados de alunos de todos os semestres, de todos os cursos, de uma IES privada.

4. Metodologia

Para desenvolver o classificador, o projeto segue boas práticas da metodologia ágil XP presentes na Tabela 3 nos Apêndices, são elas: Pequenas Versões; Projeto Simples; Metáforas; Integração Contínua; Refatoração; e Padrões de Codificação com o uso do PEP 8. Porém, são necessárias algumas modificações para desenvolvimento individual descrito como *Personal Extreme Programming* (PXP) em [Agarwal and Umphress 2008]. Possui também uma análise qualitativa e quantitativa dos resultados estatísticos, validações e testes obtidos. O planejamento do projeto segue o fluxo de atividades apresentado na Figura 15 nos Apêndices.

Este trabalho utiliza os classificadores árvore de decisão, floresta aleatória e árvores extra implementados com a biblioteca *Scikit-learn* e salvos com a biblioteca *Pickle*. A normalização é feita através da utilização das bibliotecas *Pandas* e *NumPy*. A validação é dada pelo *K-Fold*, ROC e AUC e matriz de confusão, com cálculo de sua acurácia, precisão, sensibilidade, especificidade e F1 score. A interface faz uso do *framework Django*, com gráficos criados pelas bibliotecas *Plotly* e *Seaborn*.

4.1. Programação Extrema

Programação Extrema (XP) é uma metodologia ágil de desenvolvimento de software criada por Kent Beck que emprega valores de *Feedback*, comunicação, simplicidade e co-

ragem. Sua organização é baseada em torno de um conjunto de práticas apresentadas na Tabela 3 nos Apêndices [Prikladnicki et al. 2014].

A prática de Pequenas Versões foi utilizada no decorrer do projeto para a implementação dos códigos. Já a de Integração Contínua foi utilizada na otimização dos códigos, enquanto a de Refatoração para melhorias encontradas no decorrer do desenvolvimento do projeto e a prática de Padrões de Codificação para manter uma consistência da codificação durante o desenvolvimento do projeto.

4.2. Diagrama de Atividade

O diagrama de atividade apresenta o funcionamento principal do sistema.

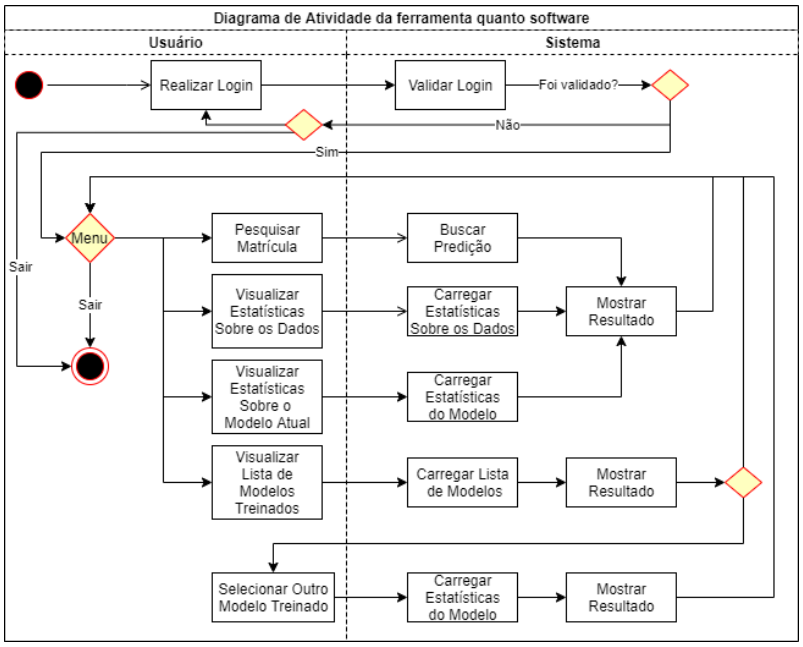

Figura 5. Diagrama de Atividade. Fonte: Autor

A Figura 5 apresenta um diagrama que possui o principal fluxo de atividades realizadas pelo *software*, onde o usuário primeiramente realiza o *login* no sistema. Após o *login*, o usuário pode realizar as ações de pesquisar um aluno pela matrícula, visualizar as estatísticas sobre os dados do banco, visualizar as estatísticas sobre o modelo em funcionamento e visualizar uma lista dos modelos treinados, onde pode selecionar um modelo para usar na classificação.

4.3. Identificação das regras

O primeiro passo para criação de um classificador é a obtenção de dados para seu treinamento, para isso, é preciso primeiro identificar quais são os possíveis motivos que levam um aluno à evadir.

Como metodologia para identificar as regras que podem influenciar na evasão, foi realizada uma pesquisa por trabalhos acadêmicos relacionados que tratam sobre o tema da evasão escolar e apresentam de alguma forma os possíveis motivos para ocorrerem. Com a leitura destes trabalhos, foi obtida uma série de possíveis motivos apresentados na Tabela 4 nos Apêndices.

4.4. Análise dos dados e montagem das regras

A próxima etapa para a construção deste trabalho foi, a partir das regras identificadas para classificar um aluno evasor, fazer a análise dos dados da base acadêmica fornecida e preparar (normalizar) o conteúdo para o uso de um classificador.

O banco de dados utilizado neste projeto foi cedido pela Universidade Franciscana (UFN), contendo 413 tabelas somando 48 Gb (*Gigabytes*) de informações referentes ao período de 2008 à 2018 de 93.5 mil alunos. Foram selecionadas 10 tabelas de interesse para a aplicação das regras. Os dados pessoais irrelevantes para essa pesquisa foram removidos para preservar a privacidade dos mesmos.

Após a identificação dos motivos para evasão e durante a análise dos dados, foram definidas quais regras poderiam ser utilizadas através da disponibilidade de dados no banco, que são:

- **Distância:** Distância entre a casa do aluno e o campus onde estuda;
- **Média de faltas:** Média de faltas por disciplina cursada;
- **Nº de parcelas pagas:** Número de parcelas pagas pelo aluno dentro do prazo de validade + 5 dias de atraso;
- **Média de dias em parcelas pagas atrasado:** Média de dias de atraso do pagamento da parcela + 5 dias;
- **Média das notas abaixo da média da turma:** Média das notas do aluno que ficaram abaixo de 0.5 da média da turma;
- **Média das notas na média da turma**: Média das notas do aluno que ficaram $\pm$ 0.5 da média da turma;
- **Média das notas acima da média da turma:** Média das notas do aluno que ficaram acima da média da turma + 0.5;
- **Razão das notas abaixo:** Dado pelo Nº de disciplinas com a nota abaixo de 0.5 da média da turma dividido pelo Nº de disciplinas cursadas;
- **Razão das notas na média:** Dado pelo Nº de disciplinas com a nota $\pm$ 0.5 da média da turma dividido pelo Nº de disciplinas cursadas;
- **Razão das notas acima:** Dado pelo Nº de disciplinas com a nota acima da média da turma + 0.5 dividido pelo Nº de disciplinas cursadas;
- **Nº de reprovações:** Número de reprovações do aluno;
- **Reprovações vs Semestre:** Refere-se a regra apresentada em [Lenhard and Martins 2019] que atribui um peso para a quantidade de reprovações de acordo com o semestre do aluno;
- **Transferência interna anteriormente:** Número de transferências internas já realizadas pelo aluno;
- **Idade:** Refere-se a idade do aluno;
- **Gênero:** Corresponde ao gênero do aluno;
- **Forma de ingresso:** Tipo de ingresso do aluno;
- **Nº de cancelamentos:** Número de cancelamentos já realizados pelo aluno;
- **Estado Civil:** Estado civil do aluno;
- **Turno do curso:** Turno do curso do aluno;
- **Semestre:** Semestre do aluno;
- **Tempo cursado:** Tempo cursado pelo aluno em anos.

A regra *Reprovações Vs Semestre* apresentada em Lenhard e Martins (2019), que atribui um peso para a quantidade de reprovações do aluno de acordo com o semestre dele, é dada por:

$$\alpha = \sum_{i=1}^{8} (9 - i) * disciplina\_reprovada_i \tag{1}$$

e foi adaptada para:

$$\alpha = \sum_{i=1}^{n\_semestres} (n\_semestres + 1 - i) * disciplina\_reprovada_i \tag{2}$$

onde $n\_semestres$ refere-se ao número de semestres do curso do aluno. Essa adaptação foi necessária, pois nem todos os cursos possuem a mesma quantidade de semestres.

4.5. Obtenção dos dados e implementação dos classificadores

Na primeira etapa da ciência de dados foi realizada a análise e aquisição dos dados do banco, onde foram extraídas as tabelas úteis para arquivos CSV e carregadas como *dataframes* do *Pandas* em Python. Na sequência foi realizada a normalização dos dados, removendo as colunas com dados desnecessários, corrigindo dados faltantes e a tipagem dos dados. Ainda nesta etapa, foi realizada a criação do arquivo CSV que possui as regras descritas na Subseção 4.5, onde foram carregados os arquivos CSV das tabelas já exploradas como *dataframes* do *Pandas*. Para a criação do arquivo CSV com as regras, o código de um curso foi passado para um *script*, em que foi montado, a partir de todos os arquivos CSV das tabelas, os dados de cada aluno do curso.

Foram implementados três classificadores para a predição de um aluno evasor para cada área separadamente e três classificadores considerando todas as áreas. Estes classificadores são: árvore de decisão, floresta aleatória e árvores extra. O treinamento e teste foram realizados através do método de validação *K-Fold*, onde o número de $splits$ (n_splits) foi determinado como 5. O $split$ que obteve os melhores resultados no treinamento do modelo foi separado e salvo em arquivos binários utilizando a biblioteca *Pickle*.

A Figura 6 apresenta a função onde é realizado o treinamento dos modelos. A função recebe os dados/regras (X), a predição (y) e o nome da área do treinamento. Nas linhas 18, 24 e 30 da Figura, o '*tcm*' corresponde a uma classe criada para centralizar os dados dos modelos salvos em um objeto somente, e assim o salvando facilmente com o *Pickle*. A função retorna uma lista de modelos treinados para a área passada.

5. Resultados e discussões

Das regras idealizadas para identificação de perfis evasores pelos classificadores, as regras **Semestre**; **Tempo cursado**; e **Número de parcelas pagas** foram desconsideradas, pois enviesavam como provável evasor todos os alunos que possuíam carga horária menor que a total para conclusão do curso, como por exemplo, qualquer aluno matriculado que está no meio de um curso seria um provável evasor.

```python
# função que treina os modelos
def kfoldTrain(X, y, area):
    list_models = []
    kf = KFold(n_splits=5)
    kf.split(X)

    for train_index, test_index in kf.split(X):
        # lista dos modelos treinados com essa esplitagem
        row = []

        # Separação de treino e teste
        X_train, X_test = X.iloc[train_index], X.iloc[test_index]
        y_train, y_test = y.iloc[train_index], y.iloc[test_index]

        # ExtraTreesClassifier
        clf_etc = ExtraTreesClassifier(n_estimators=100, random_state=0).fit(X_train, y_train)
        y_pred = clf_etc.predict(X_test)
        model_obj = tcm(clf_etc, X_train, X_test, y_train, y_test, y_pred, 'ExtraTreesClassifier', area)
        row.append(model_obj)

        # DecisionTreeClassifier
        clf_tree = DecisionTreeClassifier(random_state=0).fit(X_train, y_train)
        y_pred = clf_tree.predict(X_test)
        model_obj = tcm(clf_tree, X_train, X_test, y_train, y_test, y_pred, 'DecisionTreeClassifier', area)
        row.append(model_obj)

        # RandomForestClassifier
        clf_rf = RandomForestClassifier(random_state=0).fit(X_train, y_train)
        y_pred = clf_rf.predict(X_test)
        model_obj = tcm(clf_rf, X_train, X_test, y_train, y_test, y_pred, 'RandomForestClassifier', area)
        row.append(model_obj)

        list_models.append(row)
    return list_models
```

Figura 6. Função que cria os classificadores. Fonte: Autor

A Tabela 2 mostra os resultados obtidos pelos classificadores para as quatro áreas do conhecimento individualmente e para todas as áreas em conjunto. É visível que a floresta aleatória e árvores extra possuem resultados melhores em todas as áreas que a árvore de decisão devido a sua modelagem. Para a área das Ciências Humanas e para a área das Ciências Sociais o modelo que obteve os melhores resultados foi a árvores extra com uma acurácia de 90.66% e 93.11% respectivamente. Para a área das Ciências Tecnológicas e para área das Ciências da Saúde o modelo que obteve os melhores resultados foi a floresta aleatória com uma acurácia de 96.45% e 95.69% respectivamente.

Tabela 2. Resultados obtidos pelos classificadores

Área	**Classificador**	**Métricas**				
		Acurácia	Precisão	Sensibilidade	Especificidade	F1 score
Todas as Áreas	Árvore de Decisão	88.7%	86.85%	93.37%	83.12%	90.0%
	Floresta Aleatória	92.05%	96.35%	90.7%	94.28%	93.44%
	Árvores Extra	92.84%	94.21%	92.54%	93.2%	93.2%
Ciências Humanas	Árvore de Decisão	86.13%	88.22%	81.61%	90.19%	84.78%
	Floresta Aleatória	90.38%	91.58%	94.62%	81.21%	93.08%
	Árvores Extra	90.66%	91.78%	94.82%	81.66%	93.28%
Ciências Sociais	Árvore de Decisão	88.04%	89.78%	93.49%	75.48%	91.6%
	Floresta Aleatória	92.7%	94.48%	95.09%	87.19%	94.78%
	Árvores Extra	93.11%	95.63%	94.44%	90.05%	95.03%
Ciências Tecnológicas	Árvore de Decisão	94.8%	97.33%	96.44%	86.22%	96.88%
	Floresta Aleatória	96.45%	98.99%	96.75%	94.88%	97.86%
	Árvores Extra	95.81%	99.06%	95.92%	95.28%	97.46%
Ciências da Saúde	Árvore de Decisão	89.68%	95.54%	87.09%	93.7%	91.12%
	Floresta Aleatória	95.69%	97.16%	95.48%	95.98%	96.31%
	Árvores Extra	91.67%	90.24%	93.98%	89.21%	92.07%

5.1. Interface

As Figuras 7 e 8 apresentam uma das principais telas do sistema. A Figura 7 mostra dados sobre um aluno pesquisado para predizer se será evasor ou não. Esta tela exibe os dados das regras daquele aluno e um gráfico dando a probabilidade de ser ou não um aluno evasor.

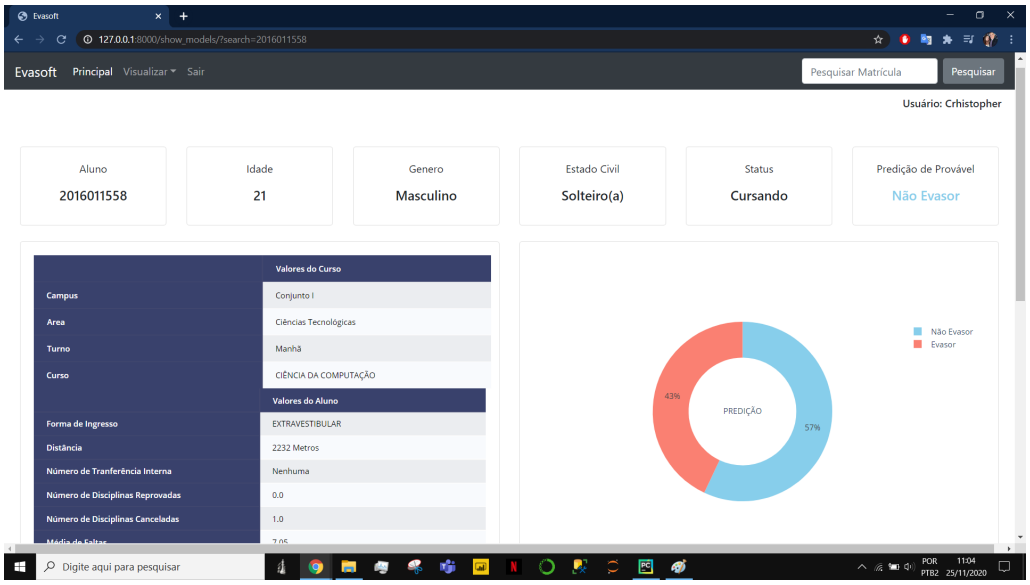

Figura 7. A interface mostra dados sobre a predição de um aluno - Primeira Parte. Fonte: Autor

E em sua continuação na Figura 8, é apresentado gráficos contendo as médias e razões das notas alcançadas por aquele aluno.

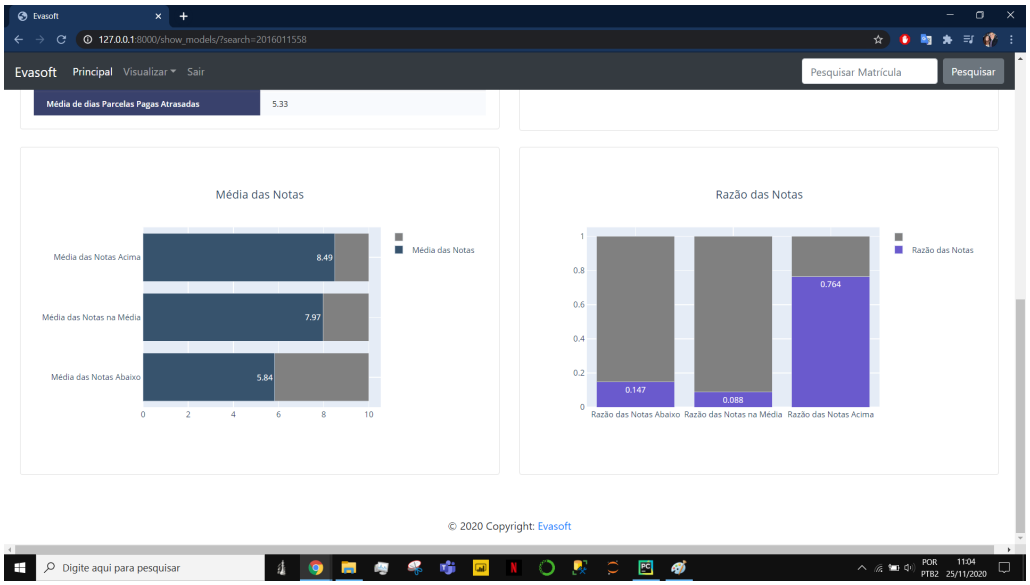

Figura 8. A interface mostra dados sobre a predição de um aluno - Segunda Parte. Fonte: Autor

5.2. Perfil do aluno evasor

Ao analisar as estruturas dos classificadores juntamente com valores estatísticos da maioria dos evasores, é possível determinar que o perfil mais aproximado a de um aluno evasor para todas as áreas é aquele que possui:

- Uma *Razão_Notas_Abaixo* maior que 0.7
- Uma *Razão_Notas_Média* menor que 0.1
- Uma *Razão_Notas_Acima* menor que 0.2
- Uma *Média_Das_Notas_Abaixo* menor que 1.27
- Uma *Média_Das_Notas_Na_Média* igual a 0 ou entre 7.0 e 7.8
- Uma *Média_Das_Notas_Acima* igual a 0 ou entre 7.5 e 8.5
- Uma *Média_de_Faltas* igual a 0
- Um *Número_de_Disciplinas_Reprovadas* igual a 0 ou igual a 3
- Está na faixa etária de 19 a 22 anos

6. Conclusão

Os resultados obtidos mostram que a identificação de perfis evasores e seus padrões, utilizando os classificadores árvore de decisão, floresta aleatória e árvores extra, implementados pela biblioteca *Scikit-learn*, com acurácias superiores à 88% para todas as quatro áreas, foram satisfatórios. Esses resultados apontam também que a utilização das bibliotecas *Pandas* e *Numpy* se mostraram eficientes para normalizar e modelar as regras a partir da base de dados acadêmica da UFN. É visível que os classificadores de floresta aleatória e árvores extra apresentam melhores resultados que os classificadores de árvore de decisão pelo fato de diminuírem a variância dos dados.

A interface construída para visualização dos resultados atingiu os resultados esperados utilizando *Django* e gráficos com *Plotly* e *Seaborn*, além de carregar os modelos já treinados a partir da biblioteca *Pickle*. Entretanto, alguns requisitos funcionais descritos nos Apêndices não foram implementados. O requisito 'Mostrar Estrutura' mostra-se inviável sua implementação, pois os classificadores árvores extra e floresta aleatória são treinados com 100 árvores de decisão. Os requisitos 'Escolher Variáveis', 'Redefinir Modelo' e 'Carregar novos dados', que tratavam do treinamento de novos modelos não foram implementados, pois não era o foco principal deste trabalho a generalização dos classificadores.

Com isso, pode-se concluir que a utilização de aprendizado de máquina supervisionado juntamente com técnicas de ciência de dados e a utilização do banco de dados acadêmico da UFN, possui uma capacidade de predição média de 91.9% da evasão de alunos evasores, e que o classificador de árvores extra obteve os melhores resultados com 92.84%, considerando sua acurácia para as todas as áreas.

Existem diversos motivos que levam um aluno à evadir de um curso, porém nota-se que muitos destes motivos estão fora do alcance do banco de dados utilizado para o treinamento, e que possíveis pesquisas como questionários aplicados nos alunos poderiam auxiliar na predição destes motivos. Ao analisar os resultados, é possível concluir que a utilização das regras de forma correlacionada permite uma melhor predição do perfil evasor e que as regras Semestre; Tempo cursado; e Número de parcelas pagas, que foram desconsideradas ao enviesar os resultados podem ser reconsideradas se modelar os alunos não evasores de modo a retirar o sentido de tempo.

Ficam como sugestão para futuros trabalhos a implementação dos requisitos funcionais 'Escolher Variáveis', 'Redefinir Modelo' e 'Carregar Novos Dados' para a generalização dos classificadores, e da busca de novos dados. E também, a análise dos dados dos anos de 2019 e 2020 para verificação assertiva dos classificadores, além da análise do impacto causado pela pandemia do Covid-19 na evasão das instituições.

Referências

Agarwal, R. and Umphress, D. (2008). Extreme programming for a single person team. In *Proceedings of the 46th Annual Southeast Regional Conference on XX*, pages 82–87.

Amidi, A. and Amidi, S. (2020). Machine learning tips and tricks cheatsheet. https://stanford.edu/ shervine/teaching/cs-229/cheatsheet-machine-learning-tips-and-tricks. Visted on: 21 mai. 2020.

Beck, K. (2003). *Test-driven development: by example*. Addison-Wesley Professional.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Dekker, G., Pechenizkiy, M., and Vleeshouwers, J. (2009). Predicting students drop out: A case study. In *Computers, Environment and Urban Systems*, pages 41–50.

Forcier, J., Bissex, P., and Chun, W. J. (2008). *Python web development with Django*. Addison-Wesley Professional.

Fritsch, R., da Rocha, C. S., and Vitelli, R. F. (2015). A evasão nos cursos de graduação em uma instituição de ensino superior privada. *Revista Educação em Questão*, 52(38):81–108.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63(1):3–42.

Grus, J. (2016). *Data Science do zero*. Alta Books, 1 edition.

Inc., P. T. (2015). Collaborative data science.

Lenhard, C. and Martins, M. O. (2019). Ia: Descrição e aplicação de regras de evasão no curso de ciência da computação em ies. *Disciplinarum Scientia— Naturais e Tecnológicas*, 20(2):199–209.

Lima Júnior, A. A. d. (2019). Mineração de dados para identificação do perfil de evasão de alunos da ufc-campus quixadá. *IX Computer on the Beach*.

Lorenzett, C. and Telöcken, A. (2016). Estudo comparativo entre os algoritmos de mineração de dados random forest e j48 na tomada de decisão. *Simpósio de Pesquisa e Desenvolvimento em Computação (SPDC)*, 2(1).

Monard, M. C. and Baranauskas, J. A. (2003). Conceitos sobre aprendizado de máquina. *Sistemas inteligentes-Fundamentos e aplicações*, 1(1):32.

Oliveira, C. H. M., Santos, F. R. T., Leitinho, J. L., and Farias, L. G. A. T. (2019). Busca dos fatores associados à evasão. *Revista Internacional de Educação Superior*, 5:e019006–e019006.

Paz, F. and Cazella, S. (2017). Identificando o perfil de evasão de alunos de graduação através da mineração de dados educacionais: um estudo de caso de uma universi-

dade comunitária. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 6, page 624.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Peter Norvig, S. R. (2013). *Inteligência Artificial*. Elsevier, 3 edition.

Pinheiro, M. A. L., da Silva, J. C., and de Souza, B. F. (2018). Aprendizado de máquina aplicado à análise de evasão no ensino superior. *Anais do Computer on the Beach*, pages 512–521.

Prado Anjos, A. P. S., da Silva Martins, N., and Pignata, E. K. d. A. A. (2019). A evasão nos cursos de licenciatura da uneb e os impactos na formação docente no oeste da bahia. *Momento-Diálogos em Educação*, 28(1):367–380.

Prikladnicki, R., Willi, R., and Milani, F. (2014). *Métodos ágeis para desenvolvimento de software*. Bookman Editora.

Prim, A. L. and Fávero, J. D. (2013). Motivos da evasão escolar nos cursos de ensino superior de uma faculdade na cidade de blumenau. *Revista E-Tech: Tecnologias para Competitividade Industrial-ISSN-1983-1838*, pages 53–72.

Python Software Foundation (2020). pickle — python object serialization. https://docs.python.org/3/library/pickle.html. Visted on: 21 mai. 2020.

Rigo, S. J., Cazella, S. C., and Cambruzzi, W. (2012). Minerando dados educacionais com foco na evasão escolar: oportunidades, desafios e necessidades. In *Anais do Workshop de Desafios da Computação Aplicada à Educação*, pages 168–177.

Santos, G. A. L., Galli, L. C. D. L. A., Neto, M. S., Giuliani, A. C., et al. (2011). A evasão no ensino superior privado: um estudo de caso em uma instituição de ensino brasileira. *Revista Ciências Administrativas*, 17(1):180–194.

Savian, M. C. B. et al. (2018). Estudo dos fatores de risco associados à evasão de alunos de graduação da universidade federal de santa maria.

Silva Filho, R. L. L., Motejunas, P. R., Hipólito, O., and Lobo, M. B. d. C. M. (2007). A evasão no ensino superior brasileiro. *Cadernos de pesquisa*, 37(132):641–659.

Tinto, V. (1999). Taking retention seriously: Rethinking the first year of college. *NACADA Journal*, 19(2):5–9.

Van Rossum, G., Warsaw, B., and Coghlan, N. (2001). Pep 8: style guide for python code. *Python. org*, 1565.

Waskom, M., Botvinnik, O., O'Kane, D., Hobson, P., Lukauskas, S., Gemperline, D. C., Augspurger, T., Halchenko, Y., Cole, J. B., Warmenhoven, J., de Ruiter, J., Pye, C., Hoyer, S., Vanderplas, J., Villalba, S., Kunter, G., Quintero, E., Bachant, P., Martin, M., Meyer, K., Miles, A., Ram, Y., Yarkoni, T., Williams, M. L., Evans, C., Fitzgerald, C., Brian, Fonnesbeck, C., Lee, A., and Qalieh, A. (2017). mwaskom/seaborn: v0.8.1 (september 2017).

7. Apêndices

7.1. Apêndice A – Requisitos Funcionais

- **RF01 —** Pesquisar Aluno:
  - O usuário insere uma matrícula e o sistema retorna a predição e as estatísticas do aluno.
- **RF02 —** Mostrar Estatísticas do classificador:
  - O sistema deve mostrar estatísticas do classificador atual.
- **RF03 —** Escolher Classificador:
  - O usuário pode escolher qual classificador ele quer utilizar na predição.
- **RF04 —** Escolher Variáveis:
  - O usuário pode escolher quais variáveis ele quer utilizar para redefinir um novo modelo.
- **RF05 —** Mostrar Estrutura:
  - O sistema deve mostrar qual a estrutura do classificador.
- **RF06 —** Redefinir Modelo:
  - O usuário pode criar um novo modelo de classificação.
- **RF07 —** Carregar novos dados:
  - O usuário pode carregar novos dados para a predição do classificador.
- **RF08 —** Salvar Modelo:
  - O sistema deverá salvar os modelos gerados.

7.2. Apêndice B – Requisitos não Funcionais

- **RNF01 —** Linguagem de Programação Python:
  - O sistema deverá ser desenvolvido utilizando a linguagem de programação Python.
- **RNF02 —** *Django*:
  - O sistema fará uso do *framework Django* do Python para desenvolvimento da interface gráfica.
- **RNF03 —** *Pandas* e *NumPy*:
  - O sistema fará uso das bibliotecas *Pandas* e *NumPy* do Python para manipulação de dados
- **RNF04 —** *Plotly* e *Seaborn*:
  - O sistema fará uso das bibliotecas *Plotly* e *Seaborn* do Python para criação de gráficos
- **RNF05 —** *Pickle*:
  - O sistema fará uso da biblioteca *Pickle* do Python para o salvamento de arquivos binários que contém os modelos de classificadores.
- **RNF06 —** *Scikit-lear*n:
  - O sistema fará uso da biblioteca *Scikit-learn* do Python para a implementação dos classificadores e validação dos mesmos.
- **RNF07 —** PEP 8:
  - A código do sistema deverá seguir os padrões de codificação apresentados no PEP 8.

7.3. Apêndice C - Interface

A primeira tela da interface, apresenta um formulário de *login* com usuário e senha.

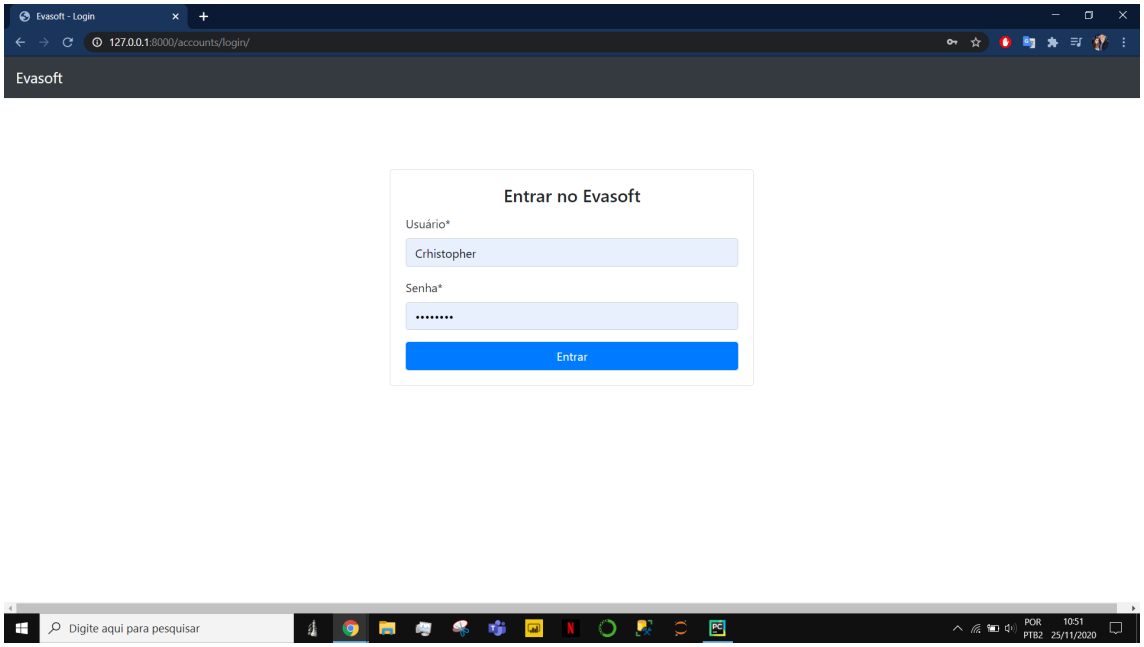

Figura 9. A interface mostra a tela de login. Fonte: Autor

Uma tela no qual é apresentado dados referente aos dados dos alunos, como o ano dos dados, número total de alunos, número de evasores, número de não evasores, visualização por gênero, uma matriz de correção das regras, e a dispersão dos alunos nas médias das notas e nas razões das notas.

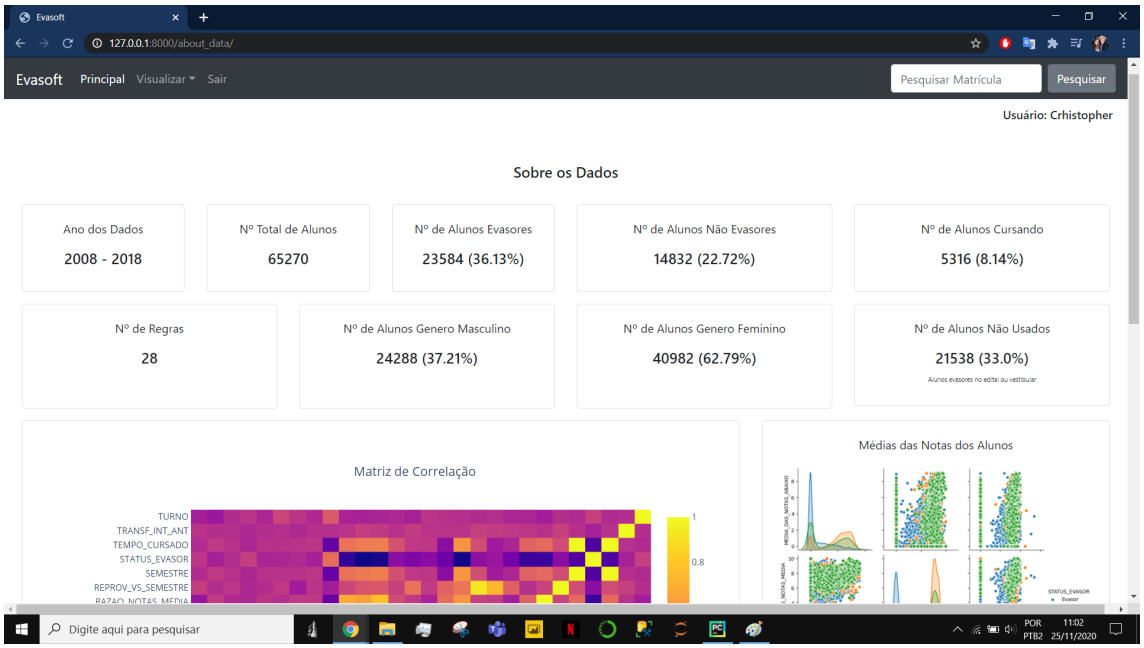

Figura 10. Interface sobre os dados dos alunos - Primeira Parte. Fonte: Autor

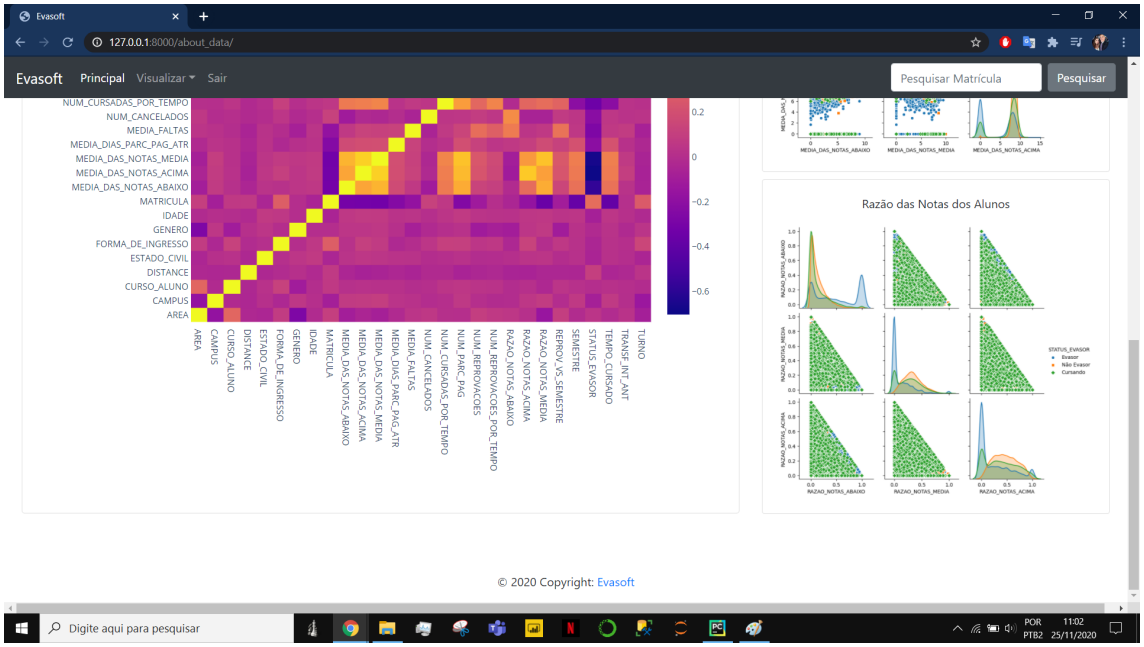

Figura 11. Interface sobre os dados dos alunos - Segunda Parte. Fonte: Autor

Uma tela para listar os modelos treinados, no qual mostra os nomes dos modelos, a área de treinamento e um botão para selecionar o modelo para predizer alunos.

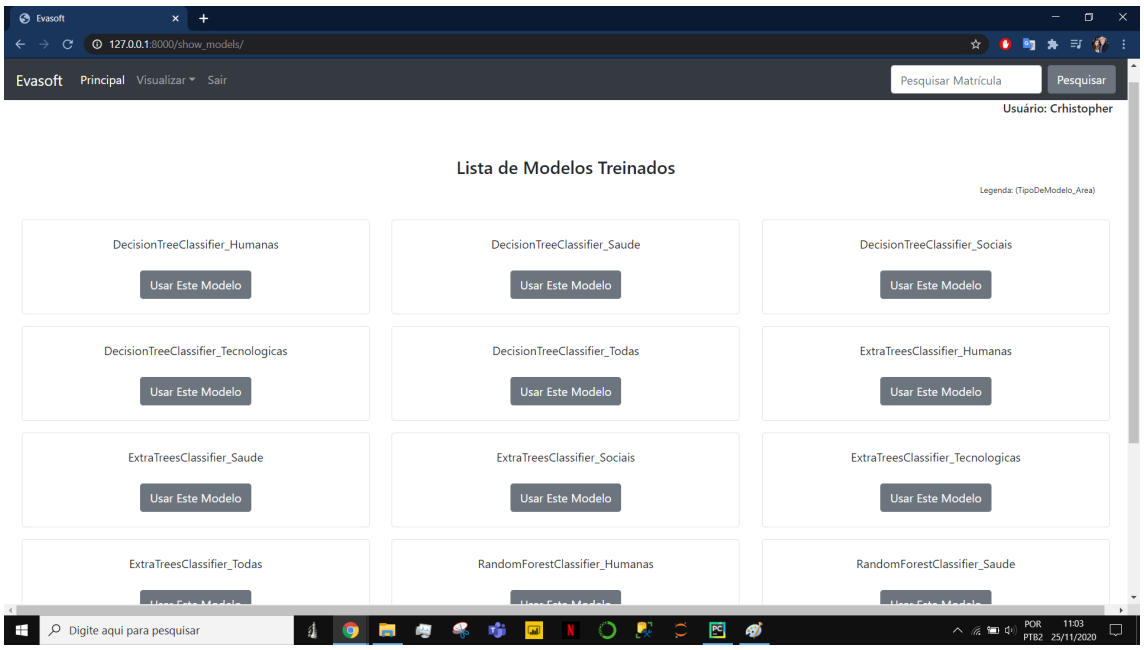

Figura 12. A interface mostra lista dos modelos salvos. Fonte: Autor

A tela inicial apresenta os dados do modelo selecionado para as predições, relatando o número de alunos usados para treino, número de alunos usados para teste, o número de regras e para qual área ele foi treinado. Possui gráficos mostrando as métricas de validação, curva ROC, matriz de confusão e a importância de cada regra o treinamento.

Possui também uma barra de navegação presente em todas as telas e uma barra de pesquisa para inserir uma matrícula.

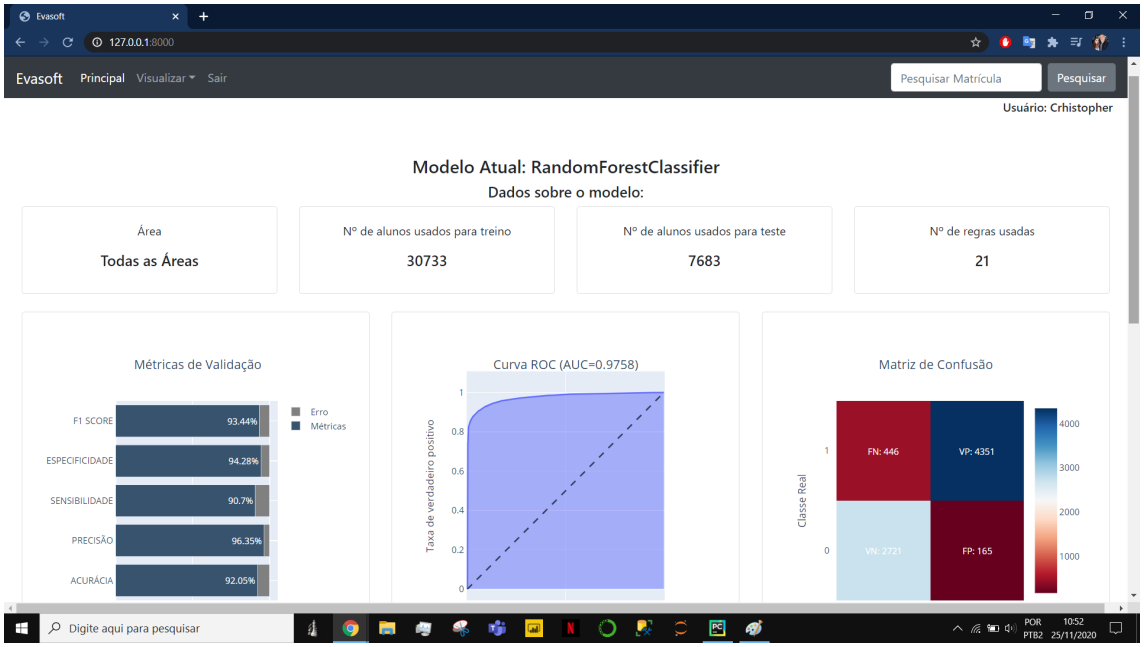

Figura 13. A interface mostra os dados sobre o modelo atual - Primeira Parte. Fonte: Autor

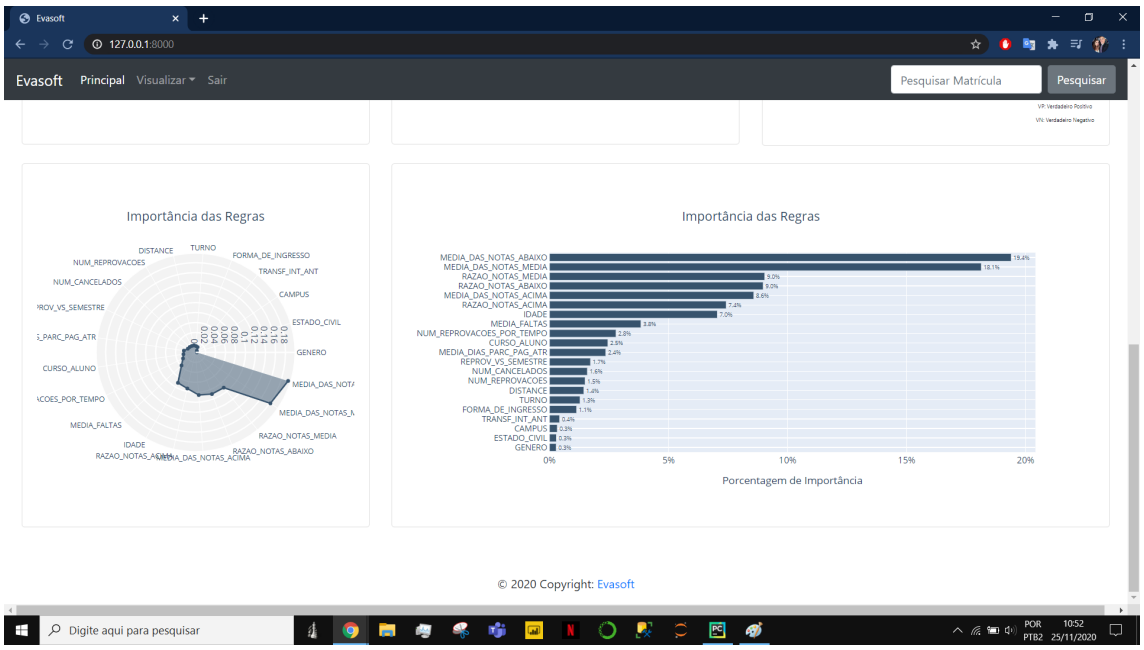

Figura 14. A interface mostra os dados sobre o modelo atual - Segunda Parte. Fonte: Autor

7.4. Apêndice D – Tabela de práticas da metodologia XP

Tabela 3. Práticas da metodologia XP

PRÁTICA	DESCRIÇÃO
Cliente Presente	O cliente que for usar o sistema está presente na equipe de desenvolvimento, podendo orientar o projeto, declarar requisitos e sanar dúvidas. Isso resulta em menos documentação pela garantia de uma comunicação eficaz [Agarwal and Umphress 2008].
Pequenas Versões	O sistema é iniciado cedo com pequenas implementações e vai sendo atualizada e complementada ao decorrer do projeto [Agarwal and Umphress 2008].
Projeto Simples	Projeto simples, mas que atenda os requisitos [Agarwal and Umphress 2008].
Jogo do Planejamento	Sua principal atividade é a negociação entre os programadores e os clientes, onde decidem quais recursos do sistema possuem maior importância [Agarwal and Umphress 2008].
Metáforas	Os projetos possuem nomes que ajudam a orientar o processo de desenvolvimento e a comunicação entre eles [Agarwal and Umphress 2008].
Posse Coletiva	O código é aberto para todos os membros e todos podem fazer alterações [Agarwal and Umphress 2008].
Integração Contínua	O seu objetivo é diminuir o código de se espalhar, ou seja, é pedido que pelo menos uma vez ao dia as alterações sejam adicionadas no código principal. Cada construção requem testes [Agarwal and Umphress 2008].
Desenvolvimento Orientado a Testes – TDD	É uma das práticas mais importante do XP, onde a validação do sistema ocorre frequentemente antes da adição de novos recursos, tendo que passar por testes [Agarwal and Umphress 2008].
Refatoração	Permite com que aplicamos pequenas alterações no código existente, melhorando sua estrutura mas preservando seu funcionamento [Agarwal and Umphress 2008].
Padrões de Codificação	Todos os membros da equipe utilizam mesmos padrões de codificação, mas sem saber quem fez qual parte [Agarwal and Umphress 2008].
Programação em Pares	Os programadores são separados em pares e codificam em apenas um computador por par, onde um revisa enquanto o outro escreve [Agarwal and Umphress 2008].

7.5. Apêndice E – Tabela de Regras observadas

Tabela 4. Tabela de Regras observadas

Nº	REGRAS	Trabalhos Acadêmicos Relacionados									
		01	02	03	04	05	06	07	08	09	10
1	Financeiro	✓	✓	✓						✓	✓
2	Reprovações		✓		✓	✓				✓	✓
3	Média individual				✓	✓			✓		✓
4	Segunda Opção										✓
5	Evasor anterior		✓								✓
6	Idade		✓				✓	✓		✓	✓
7	Gênero					✓		✓			✓
8	Falta de identificação com o curso	✓		✓			✓	✓	✓	✓	
9	Mudança de cidade	✓								✓	
10	Desemprego	✓							✓	✓	
11	Mudança no horário de trabalho	✓		✓						✓	
12	Transferência	✓		✓							
13	Problemas familiares	✓		✓						✓	
14	Doença na família	✓									
15	Doença do aluno	✓							✓	✓	
16	Estrutura da IES	✓					✓		✓		
17	Problemas particulares	✓		✓				✓	✓	✓	
18	Mudança no local de trabalho	✓		✓							
19	Mudança de país	✓									
20	Gravidez	✓								✓	
21	Dificuldade de aprendizagem	✓						✓	✓		
22	Bolsa de estudos em outra IES	✓									
23	Vaga em outra IES									✓	
24	Distância da casa			✓							
25	Forma de ingresso		✓			✓				✓	
26	Regime especial									✓	
27	Inadimplência									✓	
28	Horário			✓		✓				✓	
29	Cancelamentos anteriores		✓		✓						
30	Semestre				✓	✓	✓			✓	✓
31	Estado Civil					✓				✓	
32	Presença					✓					
33	Município onde reside						✓			✓	
34	Possuir Bolsa						✓				
35	Tempo cursado										✓

Os trabalhos acadêmicos relacionados apresentados na Tabela 4 referem-se a:

- 01: [Santos et al. 2011]
- 02: [Fritsch et al. 2015]
- 03: [Prado Anjos et al. 2019]
- 04: [Lima Júnior 2019]
- 05: [Pinheiro et al. 2018]
- 06: [Paz and Cazella 2017]
- 07: [Rigo et al. 2012]
- 08: [Oliveira et al. 2019]
- 09: [Prim and Fávero 2013]
- 10: [Lenhard and Martins 2019]

7.6. Apêndice F – Plano de Interação - Fluxo do projeto

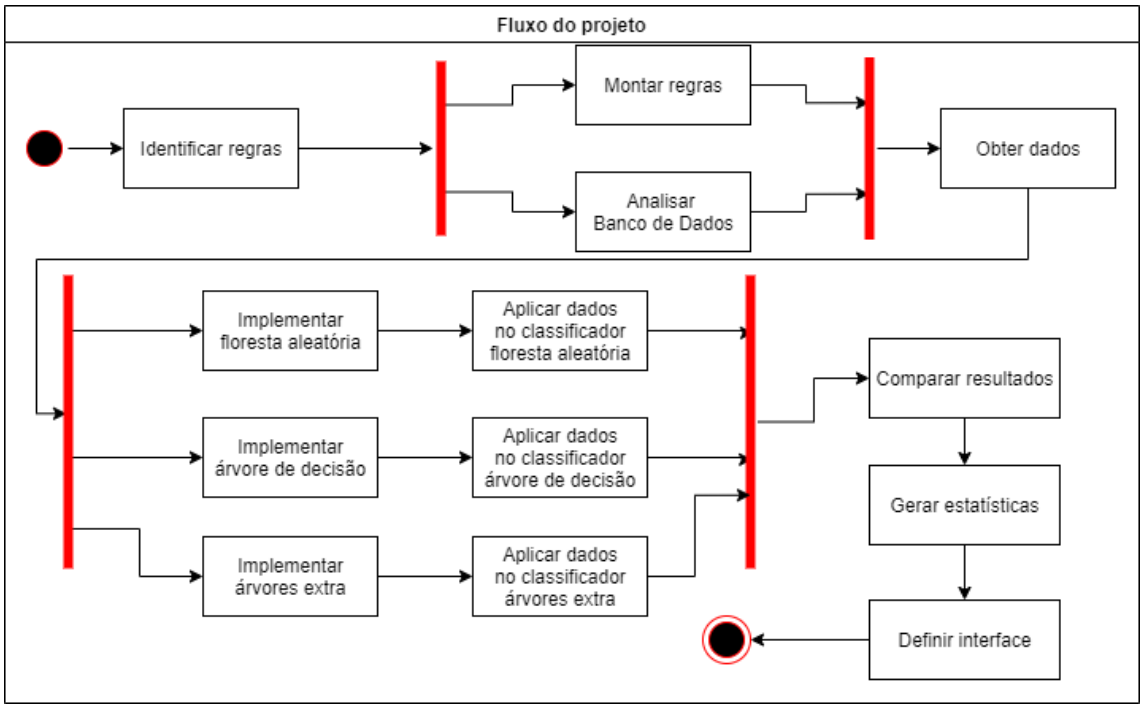

Figura 15. Fluxo de atividades do projeto. Fonte: Autor

7.7. Apêndice G – Diagrama de Sequência

O Diagrama de Sequência mostra a interação do usuário com o sistema e suas ações internas.

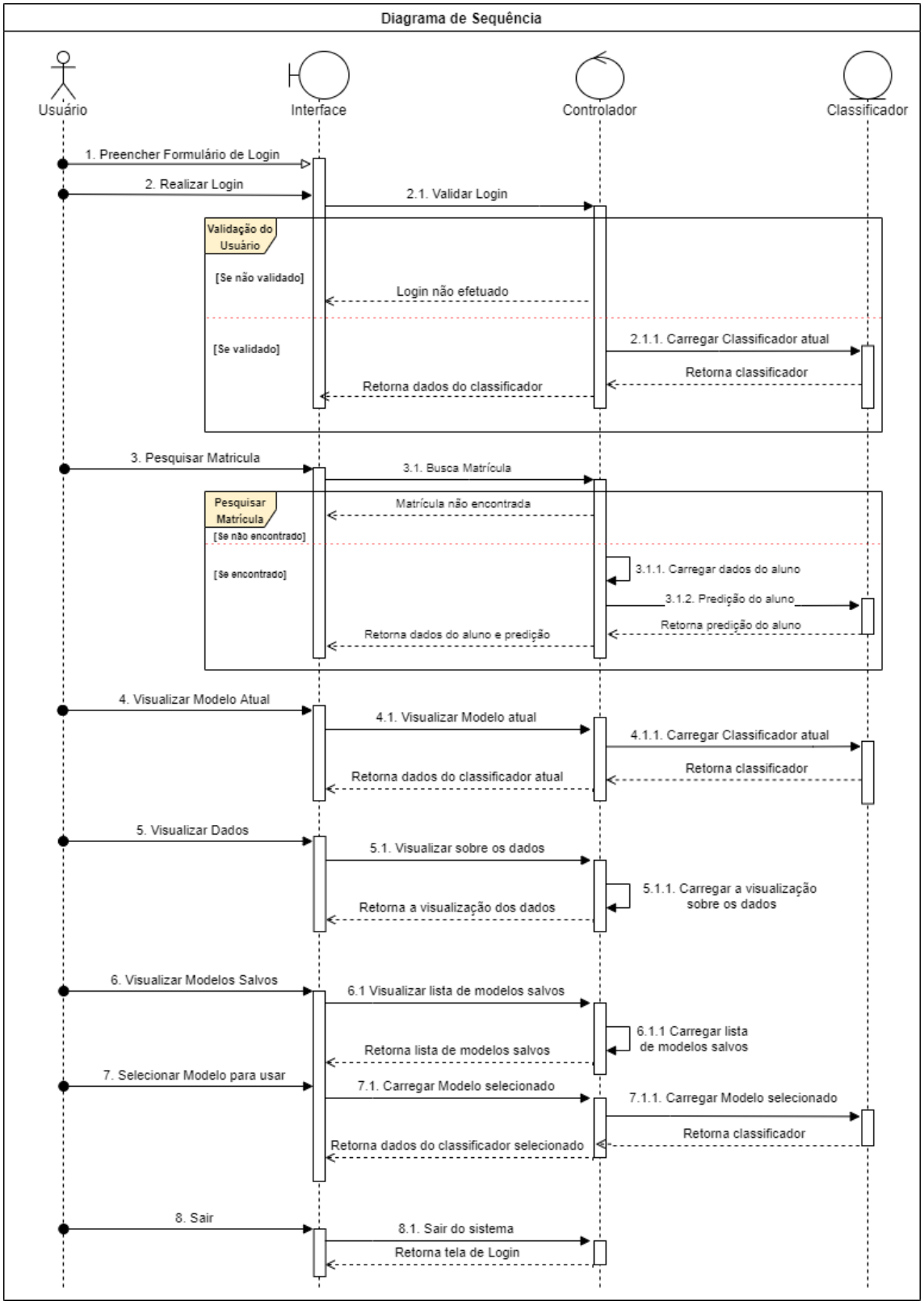

Figura 16. Diagrama de Sequência. Fonte: Autor

Como mostra a Figura 16, o usuário pode interagir com a interface, que ativa ações em um controlador que aciona o classificador do sistema. Primeiramente, o usuário

realiza o *login*, após, ele pode: Pesquisar por uma matrícula, no qual é retornado a tela com os dados daquele aluno e sua predição; Visualizar modelo atual, onde é retornado os dados sobre o modelo que está em funcionamento no sistema; Visualizar dados, em que é retornado os dados sobre todos os alunos; Visualizar modelos salvos, onde é retornado uma lista de todos os modelos salvos do sistema. Nesta visualização é possível escolher algum desses modelos para ser o novo modelo atual do sistema, e então é retornado os dados desse modelo selecionado; E sair do sistema, no qual é retornado para a tela de *login*.

7.8. Apêndice H – Resultados complementares

Uma análise geral mostra que na UFN o valor percentual de evasão na instituição nos anos de 2008 a 2018 somam 52.69%, sendo desse 9.76% das Ciências Humanas, 17.57% das Ciências Sociais, 13.3% das Ciências Tecnológicas e 12.06% das Ciências da Saúde.

Os dados mostram que na área das Ciências Humanas o gênero masculino alcançou uma porcentagem de 69.73% de evasão, e de 53.11% para o gênero feminino. Para a área das Ciências Sociais essa taxa foi de 58.1% e 50.66%. Já para a área das Ciências Tecnológicas essa taxa foi de 67.77% e 58.98%. Por fim, para a área das Ciências da Saúde, essa taxa foi de 45.68% e 39.37% para os gêneros masculino e feminino respectivamente.

Ao analisar o turno dos cursos, é visível que na área das Ciências Humanas a maior taxa de evasão é no turno da noite 64.28%, seguido do turno da manhã 52.91% e integral 37.09%. Na área das Ciências Sociais a maior porcentagem de evasão se encontra no turno da noite 54.21%, seguido pelo turno da manhã 54.12%. Na área das Ciências Tecnológicas o turno da manhã possui a maior taxa de evasão 73.18%, seguido do turno da noite com 71.36%, turno da manhã-tarde com 64.12%, turno da tarde com 57.47% e turno integral com 51.12%. Por fim, a área das Ciências da Saúde possui a maior taxa de evasão no turno da noite com 52.26%, seguido do turno da tarde com 46.0%, turno da manhã-tarde com 41.38% e 35.79% no turno integral.

A maioria dos alunos evasores não realizaram nenhuma transferência interna, o que correspondem a 55.49% na área das Ciências Humanas, 53.07% na área das Ciências Sociais, 62.85% na área das Ciências Tecnológicas e 39.65% na área das Ciências da Saúde. Entretanto, alunos que realizam 1 transferência na área das Ciências Humanas possuem uma probabilidade de evasão de 80.05%, e essa probabilidade aumenta para 90.24% para 2 transferências e 100% para 3 transferências. Para a área das Ciências Sociais, alunos que realizaram 1 transferência possuem 68.4% de chances de evadir, 83.33% para 2 transferências, 76.92% para 3 transferências e 100% para 4 transferências. Para a área das Ciências Tecnológicas, alunos que realizaram 1 transferência possuem 76.26% de chances de evadir, 84.75% para 2 transferências, 72.73% para 3 transferências e 80% para 4 transferências.

A faixa etária que apresenta a maior porcentagem de evasão é a dos estudantes entre 15 e 22 anos com 81.97% para a área das Ciências Humanas, 66.69% para a área das Ciências Sociais e 76.74% para a área das Ciências Tecnológicas. Para a área das Ciências da Saúde os alunos entre 15 e 22 anos apresentam 52.78% de evasão, porém a maior porcentagem é para alunos com mais de 50 anos com 56.41%. Outra faixa relevante é a de alunos entre 41 e 50 anos, que possui 70.52% de evasão nas Ciências Tecnológicas.

É visto nas Médias das Notas Acima uma probabilidade de evasão de 95.97% para área das Ciências Humanas, 90.31% na área das Ciências Sociais, 92.0% na área das Ciências Tecnológicas e 82.34% na área das Ciências da Saúde para médias 0. Para médias entre 1 e 5 uma probabilidade de 100% de evasão em todas as áreas pela baixa ocorrência. Para médias acima de 6 a probabilidade é de 43.91% para área das Ciências Humanas, 41.24% na área das Ciências Sociais, 50.18% na área das Ciências Tecnológicas e 26.11% na área das Ciências da Saúde. Para 0 nas Médias das Notas na Média a probabilidade de evasão é de 96.2% na área das Ciências Humanas, 90.72% na área das Ciências Sociais, 92.14% na área das Ciências Tecnológicas e 84.61% na área das Ciências da Saúde. Nas Médias das Notas Abaixo, as maiores probabilidades de estão presentes para os valores 0, 1 e 2 com: 82.83%, 91.29%, e 86.98% na área das Ciências Humanas; 75.76%, 88.8%, e 81.84% na área das Ciências Sociais; 82.99%, 89.81%, e 78.1% na área das Ciências Tecnológicas; E 64.02%, 86.49%, e 80.32% na área das Ciências da Saúde, respectivamente.

Na Razão das Notas Acima, as maiores probabilidades de estão presentes nos valores 0.0 e 1.0 com: 93.26% e 90.51% na área das Ciências Humanas; 89.71% e 77.03% na área das Ciências Sociais; 91.84% e 83.19% na área das Ciências Tecnológicas; E 80.05 e 75.91% na área das Ciências da Saúde, respectivamente. Na Razão das Notas na Média, os valores de 0.1, 0.2 e 0.3 apresentam um número significativo de evasores, porém representam uma taxa próxima a 43.5% de evasão em todas as áreas. As maiores probabilidades de se encontram nos valores 0.0 e 1.0 com: 94.02% e 85.71% na área das Ciências Humanas; 87.9% e 95.96% na área das Ciências Sociais; 90.53% e 88.41% na área das Ciências Tecnológicas; E 83.79% e 84.62% na área das Ciências da Saúde, respectivamente. Na Razão das Notas Abaixo, as maiores probabilidades de estão presentes para valores acima de 0.7 para todas as áreas e com uma probabilidade média de 95.52% de evasão.

A Média de dias Parcelas Pagas Atrasadas mostra que para a área das Ciências Humanas e para a área das Ciências Sociais a maior probabilidade está para aqueles alunos que não atrasam o pagamento com 70.28% e 63.18% respectivamente. Já para a área das Ciências Tecnológicas mostra que para qualquer quantidade de dias de atraso existe uma probabilidade entre 59% e 73%. Na área das Ciências da Saúde existe uma maior probabilidade entre 11 a 20 dias de atraso com 51.24%.

Olhando o Número de Disciplinas Canceladas, a maioria dos alunos evasores possuem nenhuma disciplina cancelada com uma probabilidade de evasão de 54.82% na área das Ciências Humanas; 50.65 % na área das Ciências Sociais; 58.67% na área das Ciências Tecnológicas; E 30.38% na área das Ciências da Saúde.

O vestibular é a forma de ingresso mais recorrente, e apresenta 58.45%, 43.0%, 66.61% e 56.45% de probabilidade de evasão para as áreas das Ciências Humanas, Ciências da Saúde, Ciências Tecnológicas e Ciências Sociais respectivamente.

A Distância que apresenta a maior probabilidade de evasão é a de 10001 e 20000 metros na área das Ciências Tecnológicas com 68.99%. Para a área das Ciências Humanas é a faixa entre 0 e 100 metros com 68.18%. Para a área das Ciências Sociais está entre 2001 e 3000 metros com 56.21%. Já a área das Ciências da Saúde apresenta 47.87% de probabilidade para a distância entre 101 e 500 metros.

Os cursos que possuem uma alta probabilidade de evasão para a área das Ciências Humanas são: Filosofia com 71.81%; Pedagogia com 71.01%; E Letras com 74.39%. Para a área das Ciências Tecnológicas são: Ciência da Computação com 74.15%; Sistemas de Informação com 72.78%; Engenharia de Materiais com 78.96%; E Matemática com 75.82%. Para a área das Ciências Sociais os cursos de Economia e Ciências Econômicas possuem maior probabilidade de evasão com 80.84% e 77.78% respectivamente. Na área das Ciências da Saúde os que possuem maior probabilidade são os cursos de Terapia Ocupacional e Farmácia com 58.8% e 57.11% respectivamente.

Observando o Campus é possível visualizar que para a área das Ciências Humanas, Ciências Tecnológicas e Ciências da Saúde a maior probabilidade de evasão se encontra no Conjunto I com 65.45%, 71.86% e 53.65% respectivamente. A área das Ciências Sociais só possui cursos no Conjunto III com 54.2% de probabilidade de evasão.