Cloud Computing and Machine Learning for Analysis of Large Volumes of Educational Data

Francisco Q. S. Neto[1], **Romero C. F. Silva**[1], **Roberta M. M. Gouveia**[1], **Maria C. Batista**[1], **Igor G. Oliveira**[1]

[1] Departamento de Estatística e Informática – Universidade Federal Rural de Pernambuco (UFRPE) – 52171-900 – Recife – PE – Brazil

{franciscofqueiroga,silvaromerocf,robertammg,cecamoraes,igorgomesdeoliveira}@gmail.com

Abstract. *This paper describes the application of supervised and unsupervised machine learning in large volumes of open governmental data from INEP. This work uses the following algorithms: K-Nearest Neighbors, Logistic Regression, Decision Tree, Random Forest and K-means. The methodology is based on the CRISP-DM and KDD processes, requiring the use of the DataBricks cloud platform. In addition, the Hadoop and Apache Spark cluster Technologies were also used. Such technologies provided high processing power for the execution of the experiments. This enabled the performance evaluation of the models and the discovery of knowledge about Brazilian basic education.*

Resumo. *Este artigo descreve a aplicação de aprendizado de máquina supervisionado e não supervisionado em grandes volumes de dados abertos governamentais do INEP, por meio dos algoritmos K-Nearest Neighbors, Logistic Regression, Decision Tree, Random Forest e K-means. A metodologia fundamenta-se nos processos CRISP-DM e KDD, sendo necessária a utilização da plataforma em nuvem DataBricks, além das tecnologias de clusters Hadoop e Apache Spark. Tais tecnologias proporcionaram alto poder de processamento para execução dos experimentos, o que viabilizou a avaliação de desempenho dos modelos e a descoberta de conhecimento da educação básica brasileira.*

1. Introdução

O trabalho surge do interesse em adquirir o respaldo científico necessário para detectar padrões e descobrir regras significativas, na tentativa de melhor compreender alguns dos desafios enfrentados na educação básica brasileira. Este artigo insere-se nas áreas interdisciplinares de *Data Science*, *Educational Data Mining* (EDM), *Machine Learning, Big Data*, dentre outras que compõem a base de conhecimento utilizada na análise de dados educacionais, com ênfase em soluções que geram conhecimento por meio da seleção, transformação e integração de dados heterogêneos, com diferentes escalas e granularidades.

Para processar os dados utilizados, fez-se necessário pesquisar sobre como manusear grandes volumes de dados através de plataformas de análises em nuvem. De acordo com Miner e Shook (2017), no modelo *MapReduce* de programação, é feito um *Mapeamento* de arquivos de entrada para blocos, seguida da distribuição equilibrada

desses blocos entre máquinas, e após o processamento, é feita uma *Redução* para sumarizar os resultados e escrevê-los em arquivos de saída. Bengfort e Kim (2016) definem *Hadoop* e *Apache Spark* como tecnologias de computação distribuída voltadas para *clustering* e processamento de grandes volumes de dados, com atenção à tolerância a falhas, sendo inspirada no *MapReduce* e *Google File System* – GFS ou GoogleFS. Além de ser uma implementação de código livre do sistema *MapReduce*, o *Spark* foi utilizado neste trabalho por proporcionar alta performance no processamento de dados, sendo esses requisitos essenciais para executar os experimentos propostos.

O trabalho utiliza as metodologias propostas pelos processos *Cross Industry Standard Process for Data Mining* (CRISP-DM) e *Knowledge Discovery in Databases* (KDD), com vistas ao desenvolvimento computacional analítico de cenários da educação básica brasileira. Tais metodologias determinam as etapas para extrair informações implícitas, previamente desconhecidas e potencialmente úteis para apoio à decisão. O KDD se propõe em encontrar e interpretar padrões/regras mediante integração de diversas fontes de dados, sendo proposto para determinar as etapas que produzem conhecimentos a partir dos dados e, principalmente, definir a etapa de *Data Mining* (Tan et al., 2018; Witten et al., 2016). O objetivo é extrair de bases de dados, sem nenhuma formulação prévia de hipóteses, informações desconhecidas a priori, factíveis, válidas e acionáveis, que poderão ser úteis para a tomada de decisão (Frawley et al., 1992; Fayyad et al., 1996). O CRISP-DM é um processo recursivo que possui seis etapas: entendimento do negócio, entendimento dos dados, preparação dos dados, modelagem, avaliação da modelagem e implantação do modelo (Wirth e Hipp, 2000).

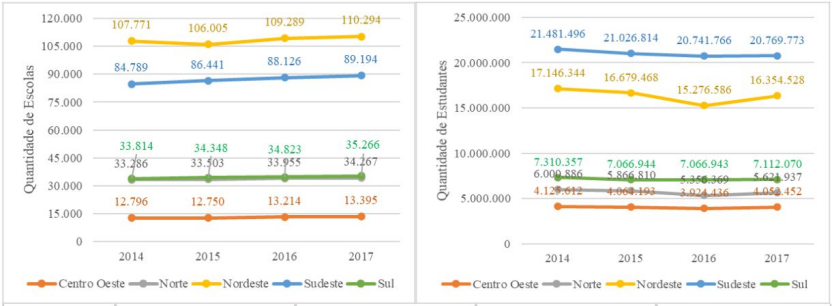
O Aprendizado de Máquina, *Machine Learning* (ML), faz parte da Inteligência Artificial, com foco na obtenção de conhecimento de forma semiautomática, já que o fator humano faz parte de todo processo. Os algoritmos de ML interpretam dados, visando produzir padrões úteis, válidos e de fácil entendimento, podendo conduzir a tomadas de decisões inteligentes. Neste trabalho foram investigados algoritmos de ML Supervisionado e Não Supervisionado – dentre os métodos de classificação, predição, regressão, associação e agrupamento (*clustering*), a fim de obter modelos bem generalizados – sem *overfitting* e *underfitting*, e com boas métricas de desempenho.

Dentre os principais objetivos do trabalho, destacam-se: (I) Avaliação e aplicação de algoritmos de classificação, predição e regressão do ML Supervisionado – mais especificamente os algoritmos *K-Nearest Neighbors* (KNN), Árvores de Decisão (*Decision Tree*), Floresta Randômica (*Random Forest*) e Regressão Logística *(Logistic Regression*), além do algoritmo de agrupamento (*clustering*) *K-Means* do Aprendizado de Máquina Não Supervisionado, buscando encontrar padrões e descobrir novos conhecimentos nos dados educacionais abertos do Censo Escolar, disponibilizados pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira – INEP; (II) Utilização de pacotes, *plugins* e bibliotecas *Python* – a exemplo de *Pandas*, *Scikit-learn*, *Pyspark-ML*, *Seaborn*, *Yellowbrick*, *NumPy*, *Matplotlib*, entre outros – para manipulação dos dados, criação dos modelos de *Machine Learning* e visualização dos resultados frente ao processamento de grandes volumes de dados da educação básica brasileira; (III) Avaliação de tecnologias de *Cloud Computing*, com aplicação de ML no contexto de *Big Data*, a exemplo de *MapReduce*, *Hadoop* e *Spark*, dentro de plataformas voltadas para *Data Science*, tais como *DataBricks*, *Amazon Elastic MapReduce – EMR, Google Colaboratory, My Binder*, entre outros.

A Mineração de Dados Educacionais – EDM, vem se consolidando nos últimos anos como uma relevante área de pesquisa, sendo publicados diversos *papers* com excelentes contribuições. Dentre os trabalhos relacionados, tem-se os artigos de Silva et al. (2019), Brandão et al. (2017), Gomes et al. (2017), Cruz Júnior et al. (2017), Tanaka et al. (2017), Simon (2017) e Gottardo et al. (2012). O presente artigo se difere dos trabalhos elencados, pelo fato de abranger uma escala muito maior de dados educacionais, sendo necessária a utilização de computação distribuída em nuvem, a fim de viabilizar a utilização de dados estudantis da educação básica de todo o Brasil, sem amostragens. Com isso, espera-se que a metodologia e experimentos realizados neste trabalho possam conduzir a melhores práticas focadas na descoberta de conhecimento eficiente em grandes volumes de dados.

2. Metodologia

Esta seção descreve os procedimentos metodológicos utilizados para realização dos experimentos computacionais de avaliação de modelos de ML, sendo abordados a seleção dos dados, pré-processamento e os métodos de *Machine Learning* utilizados. Os dados foram coletados do repositório de microdados do INEP. Nele constam os dados abertos governamentais levantados pelo INEP anualmente por meio do censo da educação brasileira, com informações sobre estudantes, turmas, docentes e instituições de ensino básico de todo o território brasileiro. Neste trabalho foram utilizados dados históricos de quatro anos referentes aos estudantes e escolas da educação básica, compreendendo o Censo Escolar de 2014 a 2017. Após a coleta dos dados, foi realizada a análise exploratória/descritiva para melhor entendimento da semântica dos dados. A Figura 1 apresenta uma visão geral da base de dados do censo da educação básica.



Brasil	2014		2015		2016		2017	
Escolas	272 mil	138*	273 mil	166*	279 mil	166*	282 mil	166*
Estudantes	56 milhões	85*	54 milhões	93*	52 milhões	92*	54 milhões	92*

* Total de atributos de cada base de dados em cada ano. Valores totais aproximados de escolas e estudantes.

Figura 1. Quantidade de escolas e estudantes da educação básica brasileira por região

Após análise exploratória, foram realizadas as etapas de pré-processamento e transformação dos dados, conforme propõem os processos CRISP-DM e KDD. As referidas etapas são essenciais para qualidade e enriquecimento da base de dados, e consequentemente para obtenção de regras mais significativas e melhores resultados dos algoritmos de ML. É nesta etapa que se detectam os erros cadastrais, inconsistência dos dados e problemas de tipagem, além de serem realizadas operações como a remoção de dados incompletos, a eliminação de dados repetidos, entre outras.

A seleção de atributos é uma etapa importante do pré-processamento, impactando consideravelmente sobre as etapas posteriores, principalmente em relação à

performance dos algoritmos de ML. Por isso, fez-se necessária a redução da quantidade de atributos, evitando assim a maldição da dimensionalidade, porém, sem perder a representatividade dos dados originais, permitindo que os algoritmos fossem executados com mais eficiência, mantendo a qualidade dos resultados. Para auxiliar nessa tarefa, foram aplicadas técnicas capazes de determinar o menor subconjunto de atributos, em conformidade com custos computacionais razoáveis. Não foram levantadas hipóteses prévias sobre quaisquer contribuições dos atributos, sendo considerado apenas o potencial de gerar novos conhecimentos. Os atributos selecionados neste trabalho possuem alta correlação com a classe e baixa intercorrelação com outros atributos, o que possibilita obter modelos mais bem generalizados.

Ainda nas etapas de pré-processamento e transformação, foram aplicadas técnicas de discretização, construção de novos atributos, estratificação, padronização e normalização de dados. Com o objetivo de maximizar a qualidade dos dados, verificou-se para cada experimento a real necessidade de discretização de atributos numéricos, bem como a estratificação de atributos classe. Tendo em vista a importância dessa etapa, este trabalho empenhou-se em realizar a limpeza visando assegurar a qualidade dos dados em relação à consistência, completude, veracidade e integridade.

Em relação à discretização, alguns atributos numéricos foram redistribuídos em intervalos categorizados, a fim de prover uma melhor interpretação para os modelos gerados. Os intervalos foram criados de modo que a quantidade de ocorrências em cada intervalo fosse próxima. Como exemplo do emprego de discretização, tem-se o atributo numérico *salas existentes* em cada instituição de ensino básico. Ao ser discretizado, esse atributo passou a ter quatro categorias com quantidade equilibrada de registros, são elas: 1 a 3 salas; 4 a 6 salas; 7 a 10 salas; e 11 ou mais salas. Além dessa conversão, foram feitos agrupamentos de informações antes fornecidas por múltiplos atributos, por exemplo, os 14 atributos relacionados à deficiência – necessidade_especial, cegueira, baixa_visao, surdez, def_auditiva, surdocegueira, def_fisica, def_intelectual, def_multipla, autismo, sindrome_asperger, sindrome_rett, transtorno_di e superdotacao – foram transformados em apenas um novo atributo, intitulado “Deficiência”, dispondo dos seguintes possíveis valores: auditiva, visual, física, autismo, múltiplas, nenhuma, outras.

Assim como na discretização, a intenção de aplicar um processo de estratificação é melhorar os dados ao tornar a distribuição de registros para valores dos atributos classe balanceada, de maneira que os algoritmos de ML não se tornem tendenciosos e aprendam de forma justa. Neste trabalho a estratificação foi realizada a partir da exclusão de registros, até que se atingisse uma distribuição satisfatória. Como exemplo, tem-se a estratificação realizada no atributo classe *Dependência Administrativa*, cuja distribuição entre alunos da rede pública e privada era discrepante. Então, o atributo classe *Dependência Administrativa* teve uma redução na quantidade de registros do tipo “pública”. O valor foi aproximado de acordo com a frequência do rótulo “privada”, e assim ajustado, de modo que a classe ficasse equilibrada entre a quantidade de estudantes de escolas públicas e privadas. Vale destacar que foram analisadas estratégias para o problema da classe rara (dados desequilibrados), por meio de técnicas *undersampling* e *oversampling*. Na seção de Resultados e Discussão, são demonstrados os experimentos que avaliaram o desempenho de algoritmos de ML considerando a estratificação do atributo classe.

Para alcançar os propósitos deste trabalho, foram selecionados algoritmos de ML que geraram modelos baseados em metas preditivas e descritivas (*profiling*), com auxílio da plataforma de análise de dados na nuvem *DataBricks*, provendo uma interface web gerenciável através de *notebooks* para execução de código Python. Diversas plataformas foram investigadas no intuito de verificar as características inerentes a cada uma, assim como a capacidade de suprir as necessidades demandadas neste estudo. A plataforma *Amazon Elastic MapReduce* foi desconsiderada em virtude do custo elevado para uso, inviabilizando a continuidade dos experimentos. O *My Binder* e *Google Colab* apresentaram limitações de uso e carga de grandes volumes de dados. No caso do *My Binder*, há a necessidade de um repositório *git* público, como *GitHub*, para execução e configuração do ambiente. Esse requisito limita o tamanho dos arquivos que podem ser carregados em um repositório. O *Colab* mostrou uma limitação semelhante, embora possua uma margem de processamento maior. Como não foi possível realizar a carga de todos os dados em nenhuma dessas duas plataformas, os experimentos em ambas foram descartados.

DataBricks é uma plataforma de *Big Data* baseada no *Apache Spark*, com infraestrutura de computação distribuída de código aberto construída em *Scala*. *DataBricks* é voltada para carga e análise de grandes volumes de dados, sendo otimizada para serviços disponibilizados em nuvem, com gerenciamento automatizado de *clusters,* cuja demanda por processamento pode ser gerenciada pela interface gráfica. A plataforma dispõe de *Notebooks Jupyter* que podem ser compartilhados por diversos usuários, suportam *markdown*, alterações *in line* de SQL – do inglês *Structured Query Language,* e dão suporte a diversas linguagens como Python, Scala e R. *DataBricks Community Edition* foi a plataforma escolhida para o presente trabalho por ser a versão gratuita da plataforma, ficando hospedada na nuvem AWS, entretanto não possui nenhum tipo de cobrança vinculada aos serviços da AWS. A plataforma possui recursos que permitem a leitura e carga de dados comprimidos, e por isso, foi capaz de carregar todos os dados coletados do Censo Escolar dos anos 2014 a 2017.

2.1. Machine Learning Aplicado em Grandes Volumes de Dados

Após a conclusão das etapas de pré-processamento e transformação dos dados, os dados se tornaram adequados para execução dos experimentos de ML. Para tal tarefa foram usadas bibliotecas Python de código aberto, como *Scikit-learn* e *Pyspark-ML*.

Para gerar alguns modelos foi utilizado o método de amostragem por meio da divisão dos dados entre treinamento e teste, do inglês *Holdout*. O conjunto de treinamento corresponde a parcela dos dados que o modelo é treinado e o conjunto de testes corresponde a usada para validar o desempenho desse modelo em dados não vistos. Em grandes bases de dados, a opção de um modelo de amostragem *Holdout* é a escolha mais interessante quando se tenta evitar problemas com tempo de treinamento e testes. Também foi testada a validação cruzada *k-fold*, com k=10, na qual o conjunto de dados D é randomicamente dividido em k subconjuntos mutuamente exclusivos de tamanhos aproximadamente iguais e o modelo é treinado e testado k vezes.

Em relação às métricas de avaliação dos modelos gerados, foram utilizadas a *Area Under the Receiver Operating Characteristic Curve* (AUC-ROC), AUC *Precision-Recall*, Acurácia, Cobertura (*Recall* ou Sensibilidade), Precisão (*Precision* ou

Valor Preditivo Positivo, VPP), *F1-Score* (ou *F-Measure*). Para os modelos de avaliação binária, cujo atributo classe contém somente dois rótulos possíveis, foram utilizadas medidas da AUC. Para os modelos de avaliação multiclasse, cujo atributo classe contém mais de dois rótulos possíveis, foram levadas em consideração a Acurácia, Cobertura, Precisão e *F1-Score*. Além dessas medidas de desempenho, foi verificado o tempo de execução médio que cada modelo levou para ser construído (treinamento) e o tempo de execução médio para ser avaliado (testes). Buscou-se avaliar ambientes com configurações distintas, mensurando a viabilidade de execução de cada modelo de ML em ambientes local e em nuvem. O trabalho contemplou o Aprendizado Supervisionado por meio dos algoritmos *K-Nearest Neighbors* (KNN), Regressão Logística *(Logistic Regression*), Árvores de Decisão (*Decision Tree*) e Floresta Randômica (*Random Forest*). Bem como contemplou o Aprendizado Não Supervisionado com o algoritmo de *clustering K-means*.

K-Nearest Neighbors (KNN) é um método de aprendizagem baseado em instâncias, nele calcula-se a distância entre o exemplo desconhecido e os outros exemplos do conjunto de treinamento por meio da fórmula da distância euclidiana, manhattan, minkowski, entre outras. Em seguida, os K vizinhos mais próximos são identificados, e por fim, o rótulo da classe com mais representantes no conjunto definido no passo anterior é o escolhido pelo voto majoritário. A Regressão Logística (RL) gera modelos que têm suas variáveis dependentes como categóricas, sendo na maioria das vezes binárias. Por ser um método de predição para variáveis categóricas, a RL busca estimar a probabilidade de essas assumirem um dos valores já determinados, de maneira que, os resultados fiquem contidos no intervalo de 0 a 1.

Classificação por Árvore de Decisão (AD) funciona como um fluxograma em forma de árvore, onde cada nó indica um teste feito sobre um valor e as ligações entre os nós representam os valores possíveis do teste do nó superior, e as folhas indicam a classe em que o registro pertence. Sendo assim, após o modelo da árvore ser construído pode-se classificar um novo registro seguindo o fluxo da árvore do nó raiz até a folha.

O método de classificação Floresta Randômica (FR) utiliza um conjunto de árvores de decisão no processo de classificação, cuja combinação de diversas árvores tenta melhorar os resultados. Um dos principais hiperparâmetros é o número de árvores construídas para o modelo, e em geral, quanto maior o número de árvores, melhor a *performance* e as predições.

K-means é um método de agrupamento (*clustering*) de quantização vetorial, pertencente ao aprendizado não supervisionado, isto é, quando os dados não possuem rotulação prévia. Ele visa particionar n observações em k *clusters,* em que cada observação pertence ao *cluster* com a média mais próxima. Assim, seu objetivo é encontrar grupos em dados não rotulados, atribuindo pontos de similaridade com base na distância de um novo dado sobre cada *cluster*. Os centróides dos k *clusters* podem ser usados para rotular novos dados, cujo treinamento atribui rótulos para cada ponto de dados e a definição de *clusters* permite localizar e analisar grupos interativamente.

3. Resultados e Discussão

Nesta seção são apresentados os resultados da execução dos algoritmos KNN, Regressão Logística, Árvore de Decisão, Floresta Randômica e *K-means*. O primeiro passo foi levantar possíveis cenários acerca da educação básica, tendo em vista os dados

coletados do censo escolar. São eles: (I) Infraestrutura das escolas brasileiras de acordo com dependência administrativa; (II) Diagnóstico da educação básica brasileira.

Os experimentos de *Machine Learning* do Cenário 1 foram realizados localmente, em um *Desktop* com processador Intel® Core™ i5, memória de 16GB, e disco HD de 1TB, sendo utilizados apenas os dados de Escolas, referentes ao censo escolar da educação básica de 2016. Já os experimentos do Cenário 2 foram realizados na plataforma em nuvem, *DataBricks*, sendo utilizadas as bases de dados de Estudantes e Escolas do censo escolar dos anos de 2014 a 2017. Os resultados dos experimentos apresentados nesse artigo têm como foco principal a análise de desempenho de diversos modelos de aprendizado de máquina, por isso a descoberta de conhecimento propriamente dita será tópico de discussões e reflexões futuras.

3.1. Cenário 1: Infraestrutura das Escolas Brasileiras de Acordo com Dependência Administrativa – Ambiente Local

Os experimentos para este cenário foram realizados localmente, utilizando duas bibliotecas de código aberto Python: *Pandas* – para manuseio dos dados, e *Scikit-learn* – para construção dos modelos de *Machine Learning*. Os dados utilizados no Cenário 1 abrangem o número total de escolas que compõem a educação básica, levantado no censo escolar de 2016, totalizando 279.358 escolas/instâncias.

Logo, este cenário tenta captar se existem e quais são as características que definem a dessemelhança de acordo com a infraestrutura de escolas: federais, estaduais, municipais e privadas. Esta classificação provém da própria base de dados do INEP e foi utilizada como atributo classe para o treinamento supervisionado. Para este cenário foram utilizados dois métodos de classificação: KNN e Regressão Logística (RL). A Tabela 1 apresenta o relatório de classificação dos modelos de predição KNN e RL.

O KNN com o K igual a 10 obteve o melhor resultado para os testes realizados, com uma taxa de acurácia de 71%, mas uma pontuação baixa para a F1-Score em algumas categorias classificadas, também ressaltando que a taxa de cobertura não obteve bons resultados em 3 das 4 categorias da classe. Já na RL, a acurácia foi de 77%, mas a F1-Score em geral mostra que a falta de balanceamento entre as classes afeta de maneira negativa os resultados obtidos, visto que a F1-Score macro possui taxa menor que 50%. Pode-se constar que as características comuns à infraestrutura das escolas federais não foram reconhecidas nessa classificação.

Tabela 1. Relatório de classificação do Cenário 1 – KNN e Regressão Logística

	Relatório de classificação do Cenário 1							
	Precision		*Recall*		*F1-Score*		*Support*	
	KNN	RL	KNN	RL	KNN	RL	KNN	RL
Estadual	0,46	0,67	0,44	0,32	0,45	0,43	37.495	37.495
Federal	0,58	0,60	0,13	0,01	0,22	0,02	740	740
Municipal	0,78	0,75	0,84	0,97	0,81	0,85	179.245	179.245
Privada	0,66	0,97	0,53	0,47	0,59	0,63	61.878	61.878
Média								
Micro avg	0,72	0,77	0,72	0,77	0,72	0,77	279.358	279.358
Macro avg	0,62	0,75	0,49	0,44	0,52	0,48		
Weighted avg	0,71	0,79	0,72	0,77	0,71	0,74		

O tempo para treinamento e testes do KNN foi de aproximadamente 40 minutos e 24 minutos, respectivamente. Já o tempo para treinamento e testes foi menor na RL, em torno de 34 minutos e 13 minutos, respectivamente. Como limitações do ambiente local, pode-se constatar os elevados tempos de treinamento e teste, já que as bibliotecas *Pandas* e *Scikit-learn* não são otimizadas para grandes volumes de dados.

3.2. Cenário 2: Diagnóstico da Educação Básica Brasileira – Ambiente de Computação Distribuída em Nuvem

Os experimentos para este cenário foram realizados com auxílio da plataforma *Databricks Community Edition*, que dispõe da biblioteca *Pyspark-ML*, possibilitando o uso do *Apache Spark* em *Python*. Com o *Spark* foi possível manusear os dados dos censos escolares da educação básica de 2014 a 2017, totalizando aproximadamente 215 milhões de instâncias entre registros de estudantes e escolas. Vale destacar que tal volume de dados inviabilizou a realização dos experimentos em ambiente local, por isso no Cenário 1 foi utilizada uma amostragem da base de dados, sendo considerada apenas os dados de escolas (e não de estudantes) do ano de 2016 (e não dos 4 anos).

Existem duas limitações na plataforma Databricks Community Edition, são elas: *quebra de memória* e *limite de inatividade*. Caso a requisição de capacidade de processamento ultrapasse o limite do *cluster*, a execução é finalizada e os dados apagados – *quebra de memória*. Nenhum tipo de informação fica salva e é necessário iniciar um novo *cluster*. O mesmo acontece para qualquer período de inatividade que ultrapasse um período limite de 120 minutos. Qualquer processo que demande muito tempo de execução e não exista nenhuma interação com o administrador, o *cluster* é finalizado e, portanto, os dados apagados – *limite de inatividade*.

O cenário em questão tenta compreender os perfis de escolas e estudantes da educação básica, para isso, foram utilizados três algoritmos do aprendizado supervisionado – Regressão Logística (RL), Árvore de Decisão (AD) e Floresta Randômica (FR), e um algoritmo do aprendizado não supervisionado – K-means. Foram realizados os seguintes experimentos: (I) Perfil de escolas públicas e privadas; (II) Perfil entre as regiões brasileiras; (III) Perfil de estudantes de região metropolitana e interior; (IV) Aprendizado não supervisionado para identificação de perfis de estudantes.

3.2.1. Perfil de escolas públicas e privadas

Os experimentos aqui descritos utilizaram métodos de aprendizagem supervisionada, tendo o atributo classe *Dependência Administrativa*, que identifica as escolas em: *federais*, *estaduais*, *municipais* e *privadas*. Entretanto, nesses experimentos o atributo classe foi transformado em binário, isto é, os estudantes foram categorizados como de escola pública ou privada. Foram gerados modelos de *Machine Learning* de RL, AD e FR, sendo realizados dois testes: sem estratificação do atributo classe (Teste 1) e com estratificação (Teste 2). Os dados utilizados correspondem ao censo escolar de 2014, sendo utilizado no Teste 1 o conjunto completo dos dados, 56.064.695 instâncias. Para o Teste 2, após a estratificação, foram utilizadas 18.549.783 instâncias. O tipo de amostragem foi *hold-out*, com distribuição 70/30, isto é, 70% da base de dados utilizada para treinamento e 30% para testes. Os resultados estão apresentados na Tabela 2, assim como o tempo decorrido para execução de ambos os testes para os três modelos.

Tabela 2. Relatório de classificação do Cenário 2 – Testes 1 e 2

	Relatório de classificação do Cenário 2 – Perfil de escolas públicas e privadas											
	AUC ROC			AUC *Precison-Recall*			Tempo Treinamento			Tempo Teste		
	RL	AD	FR	RL	AD	FR	RL	AD	FR	RL	AD	FR
Teste1	97,64	88,24	96,50	94,32	89,45	92,63	39,17	29,57	35,68	13,74	19,19	36,84
Teste2	97,76	91,36	97,28	97,99	94,76	98,04	12,27	29,82	18,95	13,30	6,64	9,30

*Tempos em minutos

Para a árvore de decisão, o melhor modelo gerado teve uma profundidade máxima igual a 5, para ambos os testes. No caso da floresta randômica, o melhor modelo gerado teve uma profundidade de 20. Os resultados demonstram uma melhora no desempenho do modelo para o teste 2, correspondendo a base de dados estratificada. Também foram realizados testes por meio de amostragem com validação cruzada, mas ambos os casos chegaram ao *limite de inatividade* do *Databricks*. Após tal constatação, a validação cruzada não foi utilizada como método de amostragem nos demais testes.

Os testes 3, 4 e 5 tentam avaliar se os modelos de predição têm desempenhos satisfatórios em uma abordagem progressiva, isto é, com comparação anual dos dados do censo escolar. O objetivo é medir a capacidade da plataforma com bases de dados maiores, além de avaliar se os algoritmos utilizados conseguem identificar diferenças entre os anos. O Teste 3 utilizou a base de dados de 2014 para treinamento e 2015 para teste; o Teste 4 utilizou 2014 para treinamento e 2016 para teste; o Teste 5 utilizou 2014 para treinamento e 2017 para teste. Os referidos testes foram realizados com estratificação do atributo classe. Os resultados estão apresentados na Tabela 3, assim como o tempo decorrido para execução de ambos os testes para os três modelos.

Tabela 3. Relatório de classificação do Cenário 2 – Testes 3, 4 e 5

	Relatório de classificação do Cenário 2 – Bases de treinamento e testes de anos distintos											
	AUC ROC			AUC *Precison-Recall*			Tempo Treinamento			Tempo Teste		
	RL	AD	FR	RL	AD	FR	RL	AD	FR	RL	AD	FR
Teste3	97,11	89,82	96,45	97,65	93,78	97,34	15,56	37,35	22,59	11,37	6,24	8,51
Teste4	97,14	89,36	96,42	97,64	93,60	97,41				11,73	6,77	9,92
Teste5	97,04	88,49	95,61	97,78	91,44	95,88				12,26	6,88	9,46

*Tempos em minutos

Os modelos gerados obtiveram bons resultados, com a RL apresentando valores acima de 97% para as duas métricas obtidas. Os modelos conseguiram classificar corretamente os dados, com base nos dados coletados em 2014, em comparação com os anos seguintes.

3.2.2. Perfil entre as regiões brasileiras

Para os experimentos desta seção foi escolhido como atributo classe a Região, que identifica os estudantes das 5 regiões do território brasileiro (Norte, Nordeste, Sul, Sudeste e Centro-Oeste), portanto corresponde a um problema de classificação multiclasse. Foram utilizadas em torno de 214 milhões de instâncias, entre dados de estudantes e suas respectivas escolas, sendo gerados modelos de *Machine Learning* de RL, AD e FR, por intermédio de três testes (Testes 6, 7 e 8). O Teste 6 utilizou a base de dados de 2014 para treinamento e 2015 para teste; da mesma forma, o Teste 7 a de 2014 e 2016, e por fim, o Teste 8 a de 2014 e 2017. Os referidos testes foram realizados sem estratificação do atributo classe. Os resultados obtidos estão apresentados na

Tabela 4, assim como o tempo decorrido para execução dos três testes para os três modelos.

Tabela 4. Relatório de classificação do Cenário 2 – Testes 6, 7 e 8

	Relatório de classificação do Cenário 2 – Perfil das regiões brasileiras											
	Acurácia			Precisão			Cobertura			F1-score		
	RL	AD	FR	RL	AD	FR	RL	AD	FR	RL	AD	FR
Teste6	57,11	84,28	74,14	64,38	84,28	78,93	57,53	84,23	74,14	60,00	84,00	65,45
Teste7	66,00	100	74,37	66,35	100	72,94	66,21	100	74,37	65,82	100	66,36
Teste8	65,75	26,61	74,53	65,93	38,71	73,04	65,75	26,62	74,30	65,61	31,44	66,47

	Tempo de Treinamento			Tempo de Teste		
	RL	AD	FR	RL	AD	FR
Teste6	115	21,76	21,74	34,24	28,30	34,21
Teste7				25,45	27,55	34,47
Teste8				27,31	32,89	32,88

*Continuação **Tempos em minutos

Os resultados da RL obtiveram taxas abaixo dos 70% em todas as métricas obtidas, sendo também seu tempo de treinamento ultrapassando os 100 minutos. Os melhores resultados obtidos para o modelo de AD tiveram uma profundidade máxima igual a 3, mas seus resultados demonstram uma disparidade entre cada teste realizado. Isso pode significar a existência de alguma característica que mudou ao longo dos anos sobre a educação em cada região. Também é necessário avaliar que a distribuição de estudantes por região não está balanceada, o que pode gerar enviesamento no treinamento do modelo. Os melhores resultados para o modelo da FR tiveram 20 árvores geradas, tendo uma taxa de precisão acima dos 72% em todos os testes. Ao contrário dos modelos de árvore de decisão, não houve disparidades nos valores das métrica dos modelos gerados pela floresta randômica.

3.2.3. Perfil de estudantes de região metropolitana e estudantes do interior

Nesses experimentos a escolha do atributo classe foi a residência do estudante, que identifica se ele reside em região metropolitana ou interior, portanto, corresponde a um problema de classificação binária. Foram utilizadas em torno de 214 milhões de instâncias, entre dados de estudantes e suas escolas, sendo gerados modelos de RL, AD e FR, por intermédio de três testes (Testes 9, 10 e 11). Os referidos testes seguiram a mesma ideia relatada na seção anterior, e foram realizados sem estratificação do atributo classe. Os resultados estão apresentados na Tabela 5, assim como o tempo decorrido para execução dos três testes para os três modelos.

Tabela 5. Relatório de classificação do Cenário 2 – Testes 9, 10 e 11

	Relatório de classificação do Cenário 2 – Perfil de estudantes de região metropolitana e interior											
	AUC ROC			AUC *Precison-Recall*			Tempo Treinamento			Tempo Teste		
	RL	AD	FR	RL	AD	FR	RL	AD	FR	RL	AD	FR
Teste9	100	89,13	99,99	100	91,00	99,99	58,61	40,59	27,66	10,07	8,70	10,10
Teste10	30	11,00	9,68	0	30,12	30				9,06	6,83	10,24
Teste11	100	89,11	99,99	100	93,20	99,99				9,92	6,48	9,23

*Tempos em minutos

O modelo de RL obteve taxas de 100% para os testes realizados com a base de 2015 e 2017, tanto para a métrica AUC ROC, quanto para AUC *Precison-Recall*. Isso

não evidencia a ausência de sobreajuste, já que a AUC *Precision-Recall* foi de 0%, demonstrando que o modelo de treinamento não conseguiu retornar com precisão nenhuma informação referente a base de dados correspondente ao censo escolar de 2016. Vale destacar que não foi aplicada nenhuma técnica de balanceamento de classe e não foram feitos testes sobre a aplicação da estratificação.

O modelo de AD que obteve os melhores resultados teve profundidade máxima igual a 7 e uma taxa de AUC acima de 89% com os testes realizados no conjunto dos dados de 2015 e 2017. Entretanto, o modelo não apresentou bons resultados sobre a base de 2016, com uma taxa de AUC sobre a curva ROC igual a 11%, e sobre a curva *Precision-recall* de 30,12%. Para o modelo de FR, os melhores resultados obtidos foram gerados sobre um modelo de 10 árvores, com uma taxa de AUC de quase 100% sobre as bases de dados de 2015 e 2017. Os testes em ambos os modelos apresentaram dificuldade na classificação dos dados de 2016, indicando a necessidade de uma melhor investigação sobre a escolha do atributo classe e sobre as técnicas de balanceamento.

3.2.4. Aprendizado não supervisionado para identificação de perfis de estudantes

Na tentativa de identificar possíveis grupos de perfis dos estudantes da educação básica foi empregado o algoritmo de *clustering* K-means, sendo utilizado o conjunto de dados do censo da educação básica de 2014 e 2015, totalizando aproximadamente 108 milhões de instâncias. Para descoberta do melhor K, executou-se o *Elbow Method*, sendo K igual a 8 o valor indicado pelo referido método. O processo de descoberta do melhor K levou 39 minutos para ser finalizado, entretanto, o treinamento, realizado sobre os dados do censo escolar de 2014, causou *quebra de memória*, não sendo possível gerar os 8 *clusters*/agrupamentos. Logo, o processo de predição, que seria realizado sobre os dados do censo escolar de 2015, não foi possível de ser concluído.

4. Considerações Finais

As contribuições do presente trabalho podem ser constatadas frente aos experimentos realizados nas bases de dados do censo escolar entre os anos de 2014 a 2017. Um censo, por si só, é um estudo estatístico que se refere a uma população e possibilita o recolhimento de diversas informações relevantes, cujos dados representam um determinado momento, neste caso um período de quatro anos da educação básica brasileira. O artigo insere-se na área de conhecimento *Data Science,* sendo utilizadas diversas técnicas, tais como estratificação e discretização, bem como discussões acerca da viabilidade e uso de *frameworks* que possibilitem o manuseio e criação de modelos de *Machine Learning* no contexto de grandes volumes de dados.

O trabalho fez uso das metodologias KDD e CRISP-DM, sendo necessária a definição de cenários, a aplicação de técnicas para melhorar a qualidade dos dados, e o levantamento de tecnologias de sistemas distribuídos em nuvem, o que permitiu sanar diversos desafios, tais como o grande volume de dados e alta dimensionalidade. Portanto, pode-se concluir que os recursos do *Spark* foram essenciais para o manuseio das bases de dados, tornando-se factível os experimentos com dados históricos, com vistas à análise e diagnóstico de cenários da educação básica, buscando identificar perfis de estudantes e infraestrutura de instituições de ensino das regiões do Brasil.

Referências

Bengfort, Benjamin; Kim, Jenny. (2016) “Data Analytics with Hadoop: An Introduction for Data Scientists”. O'Reilly Media.

Brandão, J. O. S.; Silva, A. J.; Gouveia, R. M. M.; Soares, R. G. F. (2017) “Aprendizagem de Máquina para Predição de Desempenho de Estudantes de Graduação na UFPE”. In: Brazilian Conference on Intelligent Systems (BRACIS) – XIV Encontro Nacional de Inteligência Artificial e Computacional (ENIAC).

Cruz Júnior, G.; Nascimento, R.; Alves, G.; Gouveia, R. M. M. (2017) “Identificando Correlações e Outliers entre Bases de Dados Educacionais”. In: Workshops do Congresso Brasileiro de Informática na Educação, p. 694.

Fayyad, Usama; Piatetsky-Shapiro, Gregory; Smyth, Padhraic. (1996) “From data mining to knowledge discovery in databases”. AI magazine, v. 17, n. 3, p. 37.

Frawley, William J.; Piatetsky-Shapiro, Gregory; Matheus, Christopher J. (1992) “Knowledge discovery in databases: An overview”. AI magazine, v. 13, n. 3, p. 57.

Gomes, T.; Gouveia, R. M. M.; Batista, M. (2017) “Dados Educacionais Abertos: Associações em dados dos inscritos do Exame Nacional do Ensino Médio”. In: Workshop de Informática na Escola do Congresso Brasileiro de Informática na Educação, p. 895.

Gottardo, E.; Kaestner, C.; Noronha, R. V. (2012) “Avaliação de desempenho de estudantes em cursos de educação a distância utilizando mineração de dados”. In: Anais do Workshop de Desafios da Computação Aplicada à Educação. p. 30-39.

Miner, Donald; Shook, Adam. (2017) “MapReduce design patterns: building effective algorithms and analytics for Hadoop and other systems”. 2nd ed. O'Reilly Media.

Silva, P. M.; Lima, M. N. C. A.; Soares, W. L.; Silva, I. R. R.; Fagundes, R. A. de A.; Souza, F. F. (2019) “Ensemble Regression Models Applied to Dropout in Higher Education”. In: 2019 8th Brazilian Conference on Intelligent Systems (BRACIS), Salvador, Brazil, IEEE.

Simon, Augusto; Cazella, Sílvio. (2017) “Mineração de Dados Educacionais nos Resultados do ENEM de 2015”. In: Workshops do Congresso Brasileiro de Informática na Educação, p. 754.

Tan, Pang-Ning; Steinbach, Michael; Karpatne, Anuj; Kumar, Vipin. (2018) “Introduction to Data Mining”. 2nd ed. Pearson.

Tanaka, Fabio; Silva, Gabriel; Peres, Sarajane; Fantinato, Marcelo. (2017) “Predição de desempenho de alunos no ensino a distância via mineração de processos”. In: Brazilian Conference on Intelligent Systems (BRACIS) - XIV Encontro Nacional de Inteligência Artificial e Computacional – ENIAC.

Witten, Ian H; Frank, Eibe; Hall, Mark A. (2016) “Data mining: practical machine learning tools and techniques”. 4rd ed. Morgan Kaufmann - Elsevier.

Wirth, R.; Hipp, J. (2000) “CRISP-DM: Towards a standard process model for data mining”. In Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining, p. 29-39.