Demanda da academia da UC Berkeley durante os anos de 2015 e 2017

1st Aline Fortaleza Ferreira da Silva
Centro de Informática
Universidade Federal de Pernambuco
Recife, Brasil
affs2@cin.ufpe.br

2nd Arthur Alves Marsaro
Centro de Informática
Universidade Federal de Pernambuco
Recife, Brasil
aam4@cin.ufpe.br

3rd Davi Gonzaga Guerreiro Barboza
Centro de Informática
Universidade Federal de Pernambuco
Recife, Brasil
dggb@cin.ufpe.br

4th Lucas da Silveira Absalão
Centro de Informática
Universidade Federal de Pernambuco
Recife, Brasil
lsa4@cin.ufpe.br

5th Pedro Victor Saraiva Campello
Centro de Informática
Universidade Federal de Pernambuco
Recife, Brasil
pvsc@cin.ufpe

***Abstract*—This research introduces a methodology for forecasting the optimal time to visit the gym at UC Berkeley by employing statistical analysis of available data. Utilizing a dataset sourced from the Kaggle website, the study concentrates on constructing a predictive model using Python, incorporating the principles of the Naive Bayes classifier, along with other machine learning techniques. The primary objective is to furnish students with a tool that enables them to determine the most advantageous time to access the University gym, taking into account a range of influencing factors.**

***Index Terms*—Demanda, Academia, Naive Bayes, Modelo Preditor**

I. Introdução

A prática de atividades físicas e o cultivo de hábitos saudáveis tornaram-se uma preocupação crescente na sociedade contemporânea. Com o aumento do interesse na busca por qualidade de vida, as academias de ginástica se tornaram espaços fundamentais para o desenvolvimento de atividades físicas regulares. No ambiente universitário, esses espaços desempenham um papel crucial na promoção do bem-estar entre os estudantes, professores e demais membros da comunidade acadêmica.

Assim, com uma vida universitária cada vez mais demandante, é primordial para os estudantes conseguirem prever o quão lotada estará a academia do campus em cada momento do dia. Analisar os fatores do dia pode contribuir para um melhor uso do tempo dos estudantes no dia-a-dia, algo precioso no mundo acadêmico.

II. Objetivos

Temos como finalidade desenvolver um classificador ingênuo de Bayes para estimar a demanda da academida com base em diferentes atributos, como exemplo horario, temperatura, periodo do mês e etc.

III. Justificativa

Pela demanda atual do público e pela tendência cada vez maior de uma vida mais saudável, nossa proposta tem como objetivo otimizar e analisar como os fatores externos influenciam na quantidade dos usuário de academia durante o dia, o que pode demonstram que, caso algum evento aconteça, pode haver um crescimento ou diminuição da quantidade de pessoas em um determinado horário. Logo, pretendemos, por intermédio de aplicações matemáticos, permitir encontrar horários ditos como ”ideais” para os frequentadores da academia, transformando um local de mais conforto e com um melhor rendimento para os usuários.

IV. Metodologia

Neste projeto, será analisado os diferentes níveis de demanda sobre uma academia fazendo o uso de machine learning. Nesse sentido, utilizaremos o classificador de Bayers ingênuo para a categorização dos graus de demanda e lotação utilizando diferentes varíaveis. Com isso usaremos o ambiente virtual do Google Colaboratory com códigos implementados na linguagem computacional python e algumas das bibliotecas disponíveis nela, como Scipy, Numpy, Pandas, Matplotlib, Seaborn e Scikitlearn.

A. Dataset

A base de dados utilizada para fundamentar o projeto encontra-se disponível na plataforma Kaggle, apresentando 9 atributos e 62,184 pessoas que frequentaram a University of California, Berkeley.

Os atributos foram organizados da seguinte maneira:

1) Data - String, indicando o dia analisado.
2) Hora do Dia - Inteiro [0 - 86,400], indica o período do dia, em segundos, mais frequentado.
3) Dia da Semana - Inteiro [0 (segunda) - 6 (domingo)].
4) É Final de Semana - Booleano, 0 (dia útil) e 1 (fim de semana).

5) É Feriado - Booleano, 1 (feriado) e 0 caso contrário.
6) Temperatura - Float, graus fahrenheit.
7) É Começo do Período - 1 (início do período universitário) e 0 caso contrário.
8) Mês - Inteiro [1 (janeiro) - 12 (dezembro)].
9) Hora - Inteiro [0-24].

B. Processamento do dataset

O processamento do dataset é utilizado para transformar as informações presentes na base de dados em um formato útil e eficiente antes da análise. Inicialmente, é feito o processo de criação de atributos, que divide os dados em classes semelhantes, os atributos. Contudo, esse procedimento não se faz necessário no presente projeto, pois as informações já são fornecidas na base de dados do kaggle com suas devidas classificações.

Outra etapa do processamento se trata do filtro de instâncias. Ele é necessário para alterar ou remover as instâncias que apresentam inconsistência, incompletude ou que são dispensáveis à análise. Assim como na criação de atributos, esse processo já foi feito pela própria base de dados.

Posteriormente, se faz a seleção de atributos, a qual parte do princípio que o banco de dados apresenta atributos redundantes e irrelevantes para o trabalho. Dessa forma, são selecionados apenas os dados necessários e que facilitarão o modelo de aprendizagem.

Todas essas etapas se mostram primordiais no processo de engenharia de atributos para que os dados estejam no formato adequado à entrada do algoritmo de aprendizagem. Portanto, será possível fazer as análises necessárias de maneira mais eficiente e apropriada.

C. Teorema de Bayes

Na área de teoria das probabilidades e estatística, o teorema de Bayes, também conhecido como lei de Bayes ou regra de Bayes, aborda a probabilidade de um evento com base em um conhecimento prévio que pode estar associado a esse evento. Esse teorema ilustra como ajustar as probabilidades prévias à luz de novas evidências, resultando em probabilidades posteriores.

O nome ”teorema de Bayes” é atribuído ao pastor e matemático inglês Thomas Bayes (1701-1761). Ele dedicou seus estudos ao cálculo da distribuição para o parâmetro de probabilidade em uma distribuição binomial, utilizando terminologia moderna.

Uma aplicação frequente do Teorema de Bayes ocorre em classificadores probabilísticos, como o mencionado Classificador Naive Bayes. Nesse cenário, o teorema de Bayes é empregado para calcular a probabilidade de uma instância pertencer a uma classe específica com base nas características (atributos) observadas.

D. Classificador de Naive Bayes

Um dos algoritmos mais tradicionais e significativos em aprendizado de máquina, o Naive Bayes destaca-se tanto na comunidade acadêmica quanto no mercado. Ele representa

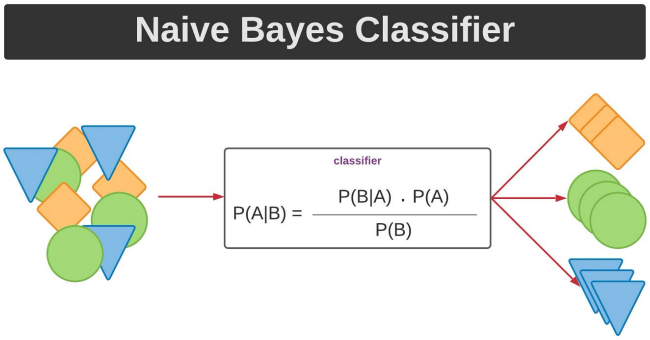

Fig. 1: Classificador de Naive Bayes.

uma abordagem simples para problemas de classificação, proporcionando uma sólida fundamentação estatística para as atividades de aprendizado de máquina (ML).

O classificador Naive Bayes é fundamentado nas descobertas de Thomas Bayes, sendo utilizado para realizar previsões em contextos de aprendizado de máquina. O termo ”naive” (ingênuo) refere-se à maneira como o algoritmo analisa as características de um conjunto de dados, assumindo que essas características são independentes entre si.

Adicionalmente, o algoritmo pressupõe que todas as variáveis features são igualmente relevantes para o resultado. Em situações em que essa premissa não se aplica, essa técnica pode não ser a escolha ideal.

Para além de suas contribuições para a estatística, Bayes também desempenhou um papel crucial no universo do aprendizado de máquina. O classificador Naive Bayes figura como um dos principais algoritmos na área, servindo como o primeiro contato com esse domínio para muitas pessoas. A fórmula do classificador define claramente os passos a serem seguidos no processo de aprendizado.

Devido à sua simplicidade e à base estatística acessível, o Naive Bayes é frequentemente preferido em muitos cenários.

E. Aplicações

Nos códigos implementado em python no Google Colab cada biblioteca tem sua função para ajudar no desenvolvimento do projeto. A biblioteca Sckit-learn será utilizada para a facilitar o uso do algoritmo Naive Bayes, enquanto bibliotecas como Pandas e Numpy serão responsáveis pela análise de dados e as operações matemáticas nelas utilizadas. Desta forma, a combinação dessas ferramentas viabiliza a exploração completa do Classificador Naive Bayes.

Os dados dentro do projeto serão analisados e tratados para garantir sua qualidade e relevância. Desse modo, o conjunto de dados será dividido, tendo o conjunto de treinamento e o conjunto de teste. O conjunto de treinamento é responsável por alimentar e ensinar o modelo de classificação, assim ajustando os padrões ao que estarão presentes nos dados. Enquanto o de teste conclui a etapa de verificação da eficiência do algoritmo classificador Naive Bayes, garantindo a capacidade de gerar

dados inusitados, assim também de avaliar a suas métricas de desempenho e de realizar previsões precisas.

Desse modo, realizando o objetivo do projeto de prever os cenários para a escolha do momento de se ir na academia, tendo assim o controle probabilístico sobre as piores e melhores ocasiões para se visitar o estabelecimento e adquirir conhecimento das características que mais influenciam na análise.

V. Análise Exploratória dos Dados

A análise exploratória de dados é de suma importância para a implementação de algoritmos preditivos, consta-se com etapas primordiais para entender e compreender os dados de um DataSet antes da síntese do modelo. Essa fase garante uma maior qualidade e confiabilidade para o modelo assim sintetizado.

Em suma essa análise consiste em uma investigação detalhada da distribuição dos dados em conjunto com a análise gráfica deles com o intuito de verificar os impactos de cada parâmetro na variável de interesse. Diversos métodos podem ser utilizados para realizar esse estudo, neste foi optado uma análise univariada, a qual se baseia num exame individual de cada elemento. Dessa forma consegue-se extrair informações fulcrais para o estudo de cada dado, como a média, moda, mediana e o valores mínimo e máximo.

Em âmbito dos dados analisados, pode-se dividir em duas categorias distintas: variáveis numéricas e variáveis categóricas, cada qual apresentando suas características individuais. As numéricas são todas aquelas que são representadas por valores numéricos, dentre elas existem também uma subclassificação, contínuas e discretas, as contínuas são que o domínio é representado por valores Reais, diferentemente das discretas, que possuem valores inteiros, como exemplo nesse DataSet as varável de month. Já as variáveis categóricas são as variáveis que possuem uma quatidade determinada de classes distintas, podendo ser binário ou mais, um exemplo dela em nosso DataSet são temperature e number-people após à feature engineering.

A. Feature Engineering

Feature Enginnering é um processo que cria, tranforma e seleciona dados para maior desempenho no processo de machine learning, nesse estudo, como dito anteriormente, as features que foram manipuladas para um melhor funcionamento do modelo, no caso trocadas de numéricas para categóricas, foram:

1) number_people: na qual foram criadas 4 cetegorias; 0 para valores entre 0 e 15, 1 para valores entre 16 e 30, 2 para valores entre 31 e 44, e 3 para valores maiores que 45.
2) temperature: Foram criadas 5 categorias, abrangindo intervalos de 10ºF, ou seja, 0 para valores entre 38ºF e 48ºF, 1 para valores entre 48ºF e 58ºF, e assim por diante até 4 para valores entre 78ºF e 88ºF.

Realizou-se uma contagem quantidade de vezes que cada categoria de numero de pessoas foi contada dentro do DataSet. Conclui-se que ele já estava balanceado e não foi necessário realizar nenhuma manipulção para garantir esse balanceamento.

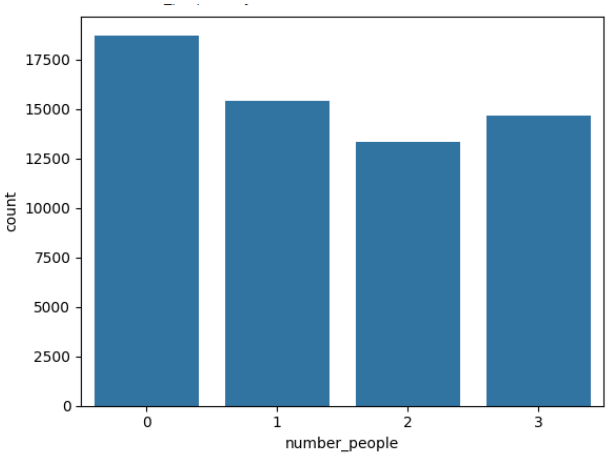

Fig. 2: Distribuição da number people.

Começando a destrinchar cada variável a fim de compreender como cada uma influência na quantidade de pessoas presentes na academia, foi gerado dois gráficos para cada variável, um com o valor númerico da mesma, e outro com os valores categorizados.

B. Pessoas X hora

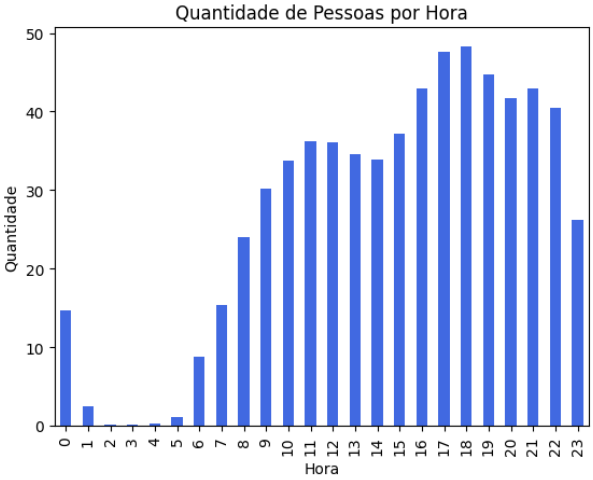

Fig. 3: Distribuição da month.

Nota-se que nesse gráfico é perceptível que as horas com a maior concentração de pessoas é no periodo de final da tarde/incio da noite.

C. Pessoas X month

Nesse gráficos é evidente que os periodos de aumento são justamente em periodos de início de semestre e de volta de periodo de férias.

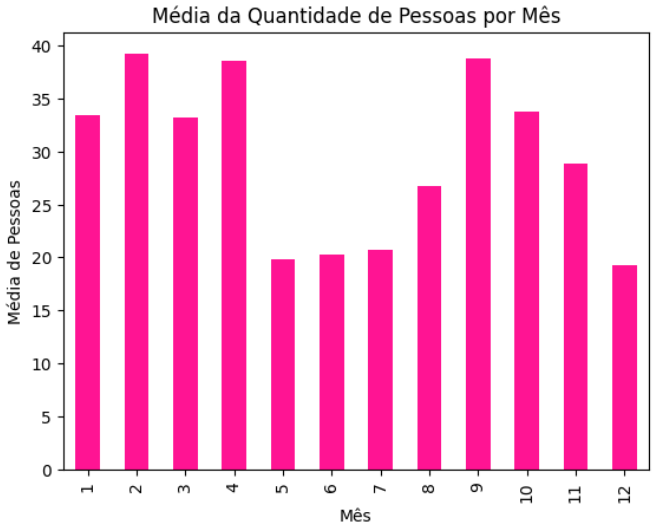

Fig. 4: Distribuição da month.

D. Pessoas X Temperatura

Ao visualizar esse gráfico é cognoscível que os alunos da UC Berkley tendem a frequentar a academia em dias mais quentes.

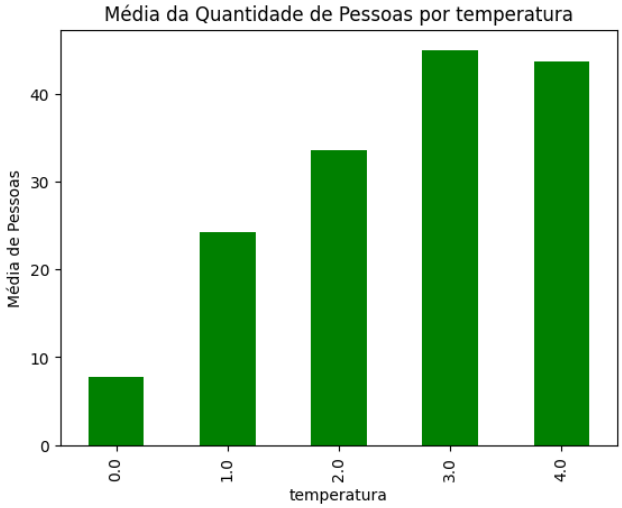

Fig. 5: Distribuição da month.

E. As outras variáveis

Nesse estudo que estamos realizando, caracterizando os restantes das variáveis (timestamp, date, is_weekend, is_start_of_semester, is_during_semester, is_holiday) como não relevantes para o impacto final, pelos seguintes motivos:

1) As variáveis timestamp, date, is_weekend, is_start_of_semester e is_during_semester já são abrangidas por hora, mês e dia da semana, qua são mais relevantes para o estudo, principalmente por causa do Naive Bayes Gaussiano que será utilizado nesse estudo.
2) A variavel is holiday tem baixissima incidência, devido estarmos pesquisando uma acedemia dos Estados Unidos da América, que é um pais com pouquíssimos feriados nacionais e locais, tal descarte não faria tanto sentido caso estiversmo estudando uma academia do Brasil.

VI. Resultados e Discussão

Feita a análise dos parâmetros mais relevantes para o projeto e admitindo a independência de seus valores, utilizamos a biblioteca Scikit-Learn e seu escopo para observar os resultados do nosso modelo.

Para isso, foram separados dois conjuntos, um de treino e outro de teste. O modelo aprende os padrões de dados esperados com base no grupo treino e, a partir disso, aplica uma predição sobre o grupo teste, utilizando 30% do tamanho do DataSet estabelecido. Logo após é utilizado as funções accuracy_score e classification_report para, respectivamente, termos a acurácia do modelo e uma análise geral em relação a medidas importantes para determinar a eficiência e o comportamento, levando em conta diferentes casos.

```
Accuracy: 0.5047169811320755
```

	precision	recall	f1-score	support
0	0.66	0.83	0.74	5671
1	0.40	0.31	0.35	4642
2	0.35	0.03	0.06	3934
3	0.42	0.72	0.53	4409
accuracy			0.50	18656
macro avg	0.46	0.47	0.42	18656
weighted avg	0.47	0.50	0.45	18656

Fig. 6: Funções

Para uma melhor visualização dos resultados obtidos, utilizamos uma matriz de confusão. Esta matriz é uma representação gráfica mais detalhada das medidas mencionadas acima. Ela mostra a quantidade de elementos de cada categoria, bem como as relações entre os dados corretos do grupo de teste e as tentativas de predição feitas pelo modelo. Com a comparação entre correto e predito, é possível realizar o cálculo das métricas de acurância, recall, precisão e f1-score. Além de indicar os pontos fortes e fracos do modelo, o que permite uma avaliação mais completa do comportamento desse.

Nessa matriz, quanto mais clara a cor e maior o número, maiores são a quantidade de predições feitas. Dessa forma, é notável, como exemplo, que a segunda categoria tem o menor recall, enquanto a primeira possui o maior. Já que a relação de elementos preditos por categoria (quantidade da linha) pela predição correta resulta nessa porcentagem.

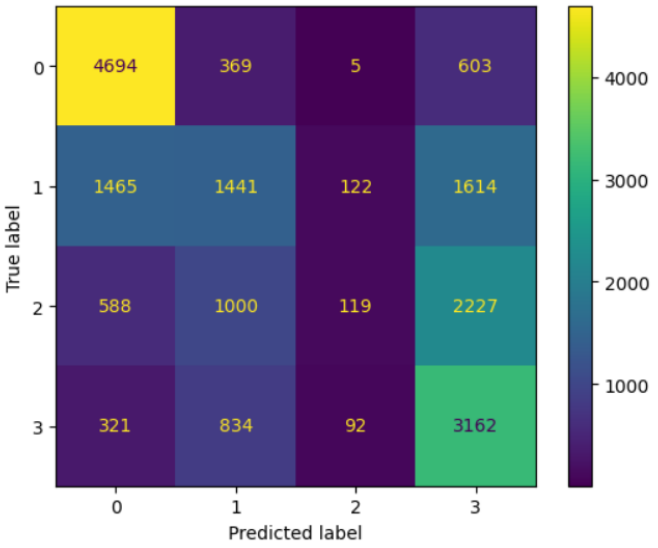

Fig. 7: Matriz de Confusão

VII. Conclusão

Devido aos dados coletados em meio ao modelo utilizado, é possível notar uma deficiência quando se fala de eficácia. Principalmente quando tratamos de dados presentes nas categorias 1 e 2.

Essa falta de abrangência aos dados reais se deve a vários possíveis fatores, entre eles é importante destacar a pequena quantidade de informações presentes na análise quando comparada a situações reais, o que significa que existem variáveis ocultas não abordadas no Dataset utilizado. Outro motivo provável é a interdependência das variáveis utilizadas, já que não existe garantia de que a assunção da independência de dados feita pelo modelo seja de fato verdadeira. Isso se dá principalmente quando lidamos com situações complexas.

Porém, mesmo que o desempenho não seja satisfatório no contexto real, ainda assim existe relevância no projeto criado. Se melhor ajustado com uma maior quantidade de dados úteis e diferentes modelos, existe uma forte perspectiva de uma pesquisa mais eficaz.

References

[1] Crowdedness at the Campus Gym https://www.kaggle.com/datasets/nsrose7224/crowdedness-at-the-campus-gym/data

[2] Bussab, Wilton de O.; Morettin, Pedro A. (2010). Estatística Básica 6 ed. São Paulo: Saraiva

[3] Gelman, Andrew; Carlin, John B.; Stern, Hal S.; Dunson, David B.; Vehtari, Aki; Rubin, Donald B. (2014). Bayesian data analysis 3 ed. Boca Raton: Chapman Hall

[4] Naive Bayes: Como funciona esse algoritmo de classificação https://blog.somostera.com/data-science/naive-bayes

[5] Classificador Naive Bayes na biblioteca Scikit Learn https://scikit-learn.org/stable/modules/naivebayes.html

[6] U.S. Environmental Protection Agency, "Exploratory Data Analysis," CADDIS - EPA, [Online]. https://www.epa.gov/caddis/exploratory-data-analysis: :text=Exploratory%20Data%20Analysis%20(EDA)%20is,step%20in%20any%20data%20analysis.

[7] Naive Bayes with Python https://www.youtube.com/watch?v=Q93IWdj5Td4&pp=ygUdbmFpdmUgYmF5bGVzIGNvbSBzY2lraXQgbGVhcm4%3D