INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE PERNAMBUCO
CAMPUS PAULISTA
CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

JEFFERSON NASCIMENTO GOMES

ANÁLISE COMPARATIVA DE MODELOS DE APRENDIZADO DE MÁQUINA PARA CLASSIFICAÇÃO DE CORRIDAS RENTÁVEIS EM PLATAFORMAS DE TRANSPORTE

Paulista - PE
2026

JEFFERSON NASCIMENTO GOMES

ANÁLISE COMPARATIVA DE MODELOS DE APRENDIZADO DE MÁQUINA PARA CLASSIFICAÇÃO DE CORRIDAS RENTÁVEIS EM PLATAFORMAS DE TRANSPORTE

Monografia apresentada ao Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco, Campus Paulista, como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.
Orientador: Prof. Antônio Sá Barreto.

Paulista - PE

2026

JEFFERSON NASCIMENTO GOMES

FICHA CATALOGRÁFICA

G633a Gomes, Jefferson Nascimento.

2026 Análise comparativa de modelos de aprendizado de máquina para classificação de corridas rentáveis em plataformas de transporte / Jefferson Nascimento Gomes. – Paulista, PE: O Autor, 2026.

34f. il. Color.

TCC (Curso Tecnológico em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Pernambuco - Campus Paulista, 2026.

Inclui Referências e Apêndice

Orientador: Prof. Antônio Sá Barreto

1. Aprendizado de Máquina. 2. CRISP-DM. 3. Classificação. 4. Corridas de Aplicativo. 5. Táxi. I. Título. II. Sá Barreto, Antônio (orientador). III. Instituto Federal de Pernambuco - Campus Paulista.

CDD 006.31

Catalogação na fonte: Bibliotecário Cristian do Nascimento Botelho CRB4/1866

JEFFERSON NASCIMENTO GOMES

ANÁLISE COMPARATIVA DE MODELOS DE APRENDIZADO DE MÁQUINA PARA CLASSIFICAÇÃO DE CORRIDAS RENTÁVEIS EM PLATAFORMAS DE TRANSPORTE

Monografia aprovada em 09 de julho de 2026.

BANCA EXAMINADORA

_____

Prof. Antônio Correia de Sá Barreto Neto – IFPE

_____

Prof. Flávio Rosendo da Silva Oliveira – IFPE

_____

Prof. Marco Mialaret Júnior – CESAR

Paulista - PE

2026

RESUMO

O crescimento das plataformas digitais de transporte urbano ampliou a quantidade de dados gerados sobre deslocamentos, valores, distâncias, horários e características operacionais das corridas. Apesar desse volume de informações, motoristas e demais agentes envolvidos nesse tipo de serviço ainda enfrentam dificuldades para transformar dados em apoio efetivo à tomada de decisão, especialmente quando o objetivo é avaliar se uma corrida tende a ser rentável ou não. Diante desse contexto, este trabalho tem como objetivo analisar e comparar modelos de aprendizado de máquina (AM) para a classificação de corridas rentáveis e não rentáveis em duas bases de dados distintas: uma base de reservas de corridas do tipo Uber/NCR e uma base pública de corridas de táxi amarelo de Nova York. A metodologia foi estruturada com base no processo CRISP-DM, contemplando compreensão do problema, compreensão dos dados, preparação dos dados, modelagem e avaliação. Foram criadas variáveis derivadas de custo estimado, lucro estimado e rentabilidade, com discussão metodológica sobre o risco de vazamento de dados, especialmente em relação ao uso de variáveis financeiras. Também foram incluídas análises de sensibilidade para verificar o impacto do uso de variáveis diretamente relacionadas ao cálculo da rentabilidade. Foram avaliados modelos como Regressão Logística, Árvore de Decisão, Random Forest, Extra Trees, Gradient Boosting, AdaBoost, modelos de baseline e uma rede neural simples com TensorFlow/Keras como experimento complementar. Os resultados indicaram que, na base Uber/NCR, o Gradient Boosting obteve F1-score aproximado de 0,99, resultado fortemente associado à presença de booking_value e ride_distance. Na base Yellow Taxi NYC, a rede neural complementar e o Random Forest atingiram F1-score próximo de 0,93, com queda para 0,7735 quando removidas duration_min e avg_speed_mph. Conclui-se que modelos de AM podem apoiar a identificação de padrões associados à rentabilidade de corridas, desde que as limitações das bases, a natureza estimada da variável alvo e a disponibilidade das variáveis no momento da decisão sejam claramente consideradas.

Palavras-chave: aprendizado de máquina; CRISP-DM; classificação; corridas de aplicativo; táxi.

ABSTRACT

The growth of digital urban transportation platforms has increased the amount of data generated about trips, fares, distances, schedules and operational characteristics. Although these data are widely available, drivers and other actors involved in platform-based transportation still face difficulties in converting information into effective decision support, especially when the purpose is to evaluate whether a ride tends to be profitable or not. In this context, this study aims to analyze and compare machine learning (ML) models for classifying profitable and non-profitable rides using two distinct datasets: an Uber/NCR-type ride-booking dataset and a public New York City Yellow Taxi dataset. The methodology was structured according to the CRISP- DM process, covering problem understanding, data understanding, data preparation, modeling and evaluation. Derived variables for estimated cost, estimated profit and profitability were created, together with a methodological discussion of data leakage risks, especially regarding financial variables. Sensitivity analyses were also included to assess the impact of variables directly related to the profitability calculation. The evaluated models included Logistic Regression, Decision Tree, Random Forest, Extra Trees, Gradient Boosting, AdaBoost, baseline models and a simple TensorFlow/Keras neural network as a complementary experiment. For the Uber/NCR dataset, Gradient Boosting achieved an approximate F1-score of 0.99, a result strongly associated with the presence of booking_value and ride_distance. For the Yellow Taxi NYC dataset, the complementary neural network and Random Forest achieved F1-scores close to 0.93, decreasing to 0.7735 when duration_min and avg_speed_mph were removed. The results indicate that ML models can support the identification of patterns associated with ride profitability, provided that dataset limitations, the estimated nature of the target variable and variable availability at decision time are explicitly considered.
Keywords: machine learning; CRISP-DM; classification; ride-hailing; taxi.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AM – Aprendizado de Máquina
CRISP-DM – Cross Industry Standard Process for Data Mining
IFPE – Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco
NYC – New York City

SUMÁRIO

1 INTRODUÇÃO

O avanço das tecnologias digitais tem promovido transformações significativas na organização da sociedade contemporânea, especialmente no que se refere ao trabalho, à mobilidade urbana e ao uso de dados na tomada de decisão. Nesse cenário, as plataformas digitais de transporte passaram a ocupar papel relevante no deslocamento urbano e na geração de renda, mediando a relação entre passageiros e motoristas por meio de sistemas computacionais. Essa lógica está associada ao fenômeno da economia de plataformas, em que serviços são ofertados e coordenados por aplicações digitais, com forte dependência de algoritmos, dados e conectividade (Sundararajan, 2016; Antunes, 2020).

No contexto dos aplicativos de transporte, a rentabilidade das corridas é uma questão central para motoristas. A decisão de aceitar uma corrida, trabalhar em determinado horário ou priorizar certas regiões geralmente envolve fatores como valor estimado, distância, tempo de deslocamento, custo operacional, demanda, trânsito e possibilidade de novas chamadas. Embora muitos desses fatores sejam observados de maneira empírica pelos motoristas, o uso sistemático de dados pode contribuir para decisões mais fundamentadas (Davenport; Harris, 2007).

A análise de dados e o aprendizado de máquina (AM) apresentam-se como alternativas relevantes para investigar padrões em grandes conjuntos de informações. Segundo Mitchell (1997), o AM consiste no estudo de algoritmos capazes de melhorar seu desempenho a partir da experiência. Em problemas de classificação, esses algoritmos aprendem relações entre variáveis de entrada e uma variável alvo, permitindo estimar a classe de novos registros. No caso deste estudo, a classe de interesse corresponde à classificação de corridas como rentáveis ou não rentáveis.

O uso de AM nesse contexto é pertinente porque as corridas apresentam características variadas, como horário, distância, duração, tipo de veículo, localidade, forma de pagamento e valor da corrida. Essas informações podem revelar padrões úteis para compreender quais atributos estão associados a maiores ou menores níveis de rentabilidade. Além disso, modelos de classificação podem comparar diferentes

abordagens e demonstrar quais técnicas apresentam melhor desempenho em dados tabulares de transporte (Santos, 2023).

Entretanto, a aplicação de inteligência artificial em bases de transporte exige cuidado metodológico. Uma corrida somente pode ser classificada como rentável quando há algum critério objetivo para estimar ganho, custo e eficiência. Muitas bases públicas não apresentam o lucro real do motorista, nem custos como combustível, manutenção, taxa da plataforma, tempo de espera ou deslocamento até o passageiro. Por isso, a rentabilidade precisa ser construída a partir de proxies, como lucro estimado por quilômetro ou lucro estimado por hora. Essa escolha deve ser explicitada para evitar interpretações equivocadas.

Outro cuidado importante está relacionado ao vazamento de dados. Em AM, ocorre vazamento quando o modelo recebe, durante o treinamento, informações que não estariam disponíveis no momento real da previsão ou que revelam diretamente a variável alvo. Em problemas de rentabilidade, variáveis financeiras como valor total, lucro ou tarifa podem elevar artificialmente o desempenho quando usadas sem critério. Por isso, este trabalho não adota uma regra única de exclusão para todas as variáveis financeiras; em vez disso, diferencia o uso dessas variáveis para construir a resposta do uso delas como entrada dos modelos, explicitando as justificativas, os riscos e as limitações de cada base analisada.

Diante desse contexto, a problemática que orienta esta pesquisa pode ser formulada da seguinte maneira: é possível utilizar modelos de AM para classificar corridas de transporte por aplicativo e táxi como rentáveis ou não rentáveis, a partir de atributos operacionais e temporais disponíveis nas bases de dados?

A estrutura deste trabalho segue uma organização inspirada no processo CRISP-DM. Após esta introdução, a seção 2 apresenta a fundamentação teórica e os trabalhos relacionados. A seção 3 descreve a metodologia, incluindo a caracterização da pesquisa, os procedimentos de coleta e análise, os aspectos éticos e as limitações. A seção 4 apresenta os resultados e a discussão. Por fim, a seção 5 trata das considerações finais.

1.1 Objetivo geral

Este trabalho tem como objetivo geral analisar e comparar o desempenho de modelos de AM na classificação de corridas rentáveis e não rentáveis em bases de transporte por aplicativo e táxi. Para isso, foram utilizadas duas bases: uma base de reservas de corridas do tipo Uber/NCR e uma base de corridas de táxi amarelo de Nova York. A primeira base representa um cenário de reservas de plataforma digital, enquanto a segunda possui maior volume de registros e atributos operacionais típicos de corridas de táxi.

1.2 Objetivos específicos

Os objetivos específicos são:

- construir variáveis de rentabilidade a partir dos dados disponíveis;
- realizar limpeza, tratamento e transformação das bases;
- aplicar modelos de classificação supervisionada;
- comparar os modelos por métricas como acurácia, precisão, recall, F1-score e ROC-AUC;
- analisar a importância das variáveis;
- discutir as limitações metodológicas relacionadas ao uso de dados estimados e aos possíveis riscos de vazamento de informação.

1.3 Justificativa

A relevância deste estudo justifica-se sob duas perspectivas. Do ponto de vista acadêmico, a pesquisa contribui para a aplicação prática de técnicas de inteligência artificial, AM e mineração de dados em um problema real de mobilidade urbana. Do ponto de vista prático, o estudo pode auxiliar na compreensão de padrões de rentabilidade, indicando quais variáveis tendem a influenciar a classificação das corridas e como modelos de classificação podem apoiar análises futuras.

2 FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

2.1 Economia de plataformas e mobilidade urbana

A economia de plataformas é caracterizada pela mediação tecnológica de serviços, conectando usuários, trabalhadores e empresas por meio de sistemas digitais. No setor de transporte urbano, aplicativos como Uber e 99 popularizaram um modelo em que a demanda por deslocamento é organizada por algoritmos, com precificação dinâmica, avaliação de usuários e acompanhamento em tempo real. Sundararajan (2016) discute esse movimento como parte de uma transformação mais ampla nas formas de consumo e trabalho.

No Brasil, autores como Antunes (2020) destacam que as formas contemporâneas de trabalho digital podem ampliar a flexibilização e transferir parte dos riscos ao trabalhador. Para motoristas de aplicativos, esses riscos aparecem na necessidade de arcar com custos operacionais, manutenção, combustível, tempo ocioso e incertezas relacionadas à demanda. Nesse ambiente, o uso de dados pode contribuir para tornar a tomada de decisão mais estratégica.

2.2 Mineração de dados e aprendizado de máquina

A mineração de dados busca identificar padrões, relações e conhecimentos úteis em conjuntos de dados. Tan, Steinbach e Kumar (2009) destacam que esse processo envolve etapas como preparação, análise, modelagem e interpretação dos resultados. Han, Kamber e Pei (2011) também reforçam que a análise de dados permite extrair conhecimento relevante de grandes volumes de informações.

O AM, por sua vez, é uma subárea da inteligência artificial voltada ao desenvolvimento de algoritmos que aprendem a partir de dados. Em tarefas supervisionadas, os dados possuem uma variável alvo previamente definida, permitindo que o modelo aprenda a associar atributos de entrada a uma resposta esperada. Neste trabalho, a variável alvo é binária: corrida rentável ou corrida não rentável.

A classificação binária é adequada ao problema proposto porque permite separar as corridas em duas classes e avaliar o desempenho dos modelos por métricas objetivas. Essa abordagem é comum em problemas de risco, aprovação, detecção e previsão, nos

quais a decisão envolve identificar se um evento pertence ou não a uma classe de interesse.

2.3 Modelos de classificação utilizados

A Regressão Logística é um modelo estatístico amplamente utilizado em problemas de classificação binária. Embora seu nome contenha o termo regressão, seu objetivo é estimar a probabilidade de pertencimento a uma classe. Por ser relativamente simples e interpretável, a Regressão Logística foi utilizada como modelo de comparação neste estudo (Hastie; Tibshirani; Friedman, 2009).

A Árvore de Decisão é um algoritmo que cria regras de separação dos dados com base nos atributos mais relevantes. Sua principal vantagem é a interpretabilidade, pois suas decisões podem ser visualizadas como uma sequência de perguntas. Entretanto, árvores isoladas podem sofrer sobreajuste quando não são controladas por parâmetros como profundidade máxima e número mínimo de amostras por folha (Hastie; Tibshirani; Friedman, 2009).

O Random Forest, proposto por Breiman (2001), é um método de ensemble que combina várias árvores de decisão. A ideia central é reduzir a variância e melhorar a generalização por meio da agregação de diversos modelos. Em bases tabulares, esse tipo de algoritmo costuma apresentar bom desempenho e permitir análise da importância das variáveis.

O Extra Trees também é um ensemble baseado em árvores, mas introduz maior aleatoriedade na escolha dos atributos e dos pontos de divisão. Essa característica aumenta a diversidade entre as árvores e pode reduzir variância, embora o desempenho dependa da estrutura da base (Geurts; Ernst; Wehenkel, 2006).

O Gradient Boosting e suas variações, como o HistGradientBoosting utilizado no notebook Uber/NCR, constroem modelos de forma sequencial, corrigindo erros cometidos pelos modelos anteriores. Friedman (2001) apresenta a base teórica desse tipo de abordagem.[1]

[1] Notebook utilizado nos experimentos: notebook_uber_ncr_tcc_notebook (4).ipynb. Disponível em: https://colab.research.google.com/drive/1_tCWLv5Lj5kDhXIn8P9nQn1fz6QW4Fi8?usp=sharing. Acesso em: 2 ago. 2026.

O AdaBoost combina classificadores fracos de forma iterativa, atribuindo maior atenção aos exemplos classificados incorretamente em etapas anteriores. Essa estratégia busca produzir um classificador final mais forte a partir da combinação dos modelos componentes (Freund; Schapire, 1997).

Além dos modelos clássicos, foi incluída uma rede neural simples com TensorFlow/Keras como experimento complementar. Redes neurais podem capturar relações não lineares e, dependendo da estrutura dos dados, atingir desempenho comparável ao dos modelos clássicos avaliados. No entanto, como este trabalho também busca compreender os fatores associados à rentabilidade, os modelos baseados em árvores permaneceram relevantes por permitirem análises complementares, como a importância das variáveis. Assim, a rede neural foi utilizada como comparação adicional, sem substituir a análise dos modelos clássicos (Abadi et al., 2016).

2.4 CRISP-DM como estrutura metodológica

O CRISP-DM, sigla para Cross Industry Standard Process for Data Mining, é um processo amplamente utilizado para orientar projetos de mineração de dados. O método organiza o projeto em fases como compreensão do negócio, compreensão dos dados, preparação dos dados, modelagem, avaliação e implantação. Neste estudo, a implantação não foi realizada, pois o objetivo é acadêmico e comparativo (Chapman et al., 2000).

A escolha do CRISP-DM se justifica pela necessidade de conduzir a pesquisa de maneira organizada, permitindo documentar desde o entendimento do problema até a avaliação dos modelos. Essa organização também aproxima este trabalho da estrutura adotada no artigo de Manguinho (2023), utilizado como inspiração para organizar a análise comparativa entre algoritmos de inteligência artificial.

3 METODOLOGIA

3.1 Caracterização da pesquisa

Este trabalho caracteriza-se como uma pesquisa aplicada, de abordagem quantitativa, com objetivo exploratório e comparativo. A pesquisa é aplicada porque utiliza técnicas computacionais para investigar um problema prático relacionado à classificação

da rentabilidade de corridas. É quantitativa porque se baseia em variáveis numéricas, métricas estatísticas e avaliação comparativa de desempenho de modelos.

A metodologia foi estruturada a partir das fases do CRISP-DM, adaptadas ao contexto da pesquisa: compreensão do problema, compreensão dos dados, preparação dos dados, modelagem e avaliação. A etapa de implantação não foi contemplada, pois o estudo não tem como objetivo desenvolver um sistema em produção, mas sim avaliar modelos e discutir sua aplicabilidade.

3.2 Procedimentos de coleta e análise

3.2.1 Compreensão do problema

O problema central consiste em classificar corridas como rentáveis ou não rentáveis a partir de dados operacionais e temporais. A rentabilidade foi tratada como uma variável estimada, pois as bases utilizadas não apresentam o lucro líquido real do motorista. Assim, foram criadas variáveis de custo e lucro com base em critérios definidos nos notebooks.

Na base Uber/NCR, a rentabilidade foi definida por meio do lucro estimado por quilômetro. Na base Yellow Taxi NYC, a rentabilidade foi definida por meio do lucro estimado por hora. Essas duas estratégias refletem características diferentes das bases e permitem comparar o comportamento dos modelos em cenários distintos. A diferença entre lucro por quilômetro e lucro por hora não representa inconsistência metodológica, mas uma adaptação ao nível de informação disponível em cada base de dados.

Na base Uber/NCR, a disponibilidade de valor da reserva e distância da corrida favoreceu a construção de um indicador de rentabilidade por quilômetro. Já na base Yellow Taxi NYC, a existência de horários de embarque e desembarque permitiu calcular a duração da corrida e, consequentemente, estimar a rentabilidade por hora. Portanto, os resultados devem ser avaliados dentro de cada cenário experimental, e não como uma disputa direta entre bases de dados equivalentes.

3.2.2 Bases de dados utilizadas

Foram utilizadas duas bases de dados. A primeira corresponde à base NCR Ride Bookings, relacionada a reservas de corridas em uma plataforma de transporte e

disponibilizada publicamente no Kaggle (Laddha, s.d.). Essa base possui variáveis como status da reserva, tipo de veículo, locais de embarque e desembarque, tempo médio de chegada, valor da reserva, distância da corrida, avaliações e forma de pagamento.[2]

A segunda base corresponde à Yellow Taxi NYC, composta por registros públicos de corridas de táxi amarelo de Nova York e disponibilizada no Kaggle a partir dos dados de viagens da cidade (Elemento, s.d.; New York City Taxi and Limousine Commission, 2015). Essa base possui variáveis como data e hora de embarque e desembarque, quantidade de passageiros, distância da corrida, coordenadas de origem e destino, código tarifário, tipo de pagamento, tarifa, gorjeta, pedágios, taxas e valor total.[3]

A Tabela 1 sintetiza o volume de registros, os principais atributos e o critério de rentabilidade de cada base.

Tabela 1 – Síntese das bases de dados utilizadas

Base	Registros analisados	Principais atributos	Critério de rentabilidade
Uber/NCR	150.000 registros iniciais; 93.000 corridas concluídas; 81.099 após limpeza; amostra de 60.000 na modelagem	booking_value, ride_distance, vehicle_type, avg_vtat, avg_ctat, ratings, horário, período, locais e pagamento	Lucro estimado por km; rentável acima do percentil 70
Yellow Taxi NYC	887.273 registros iniciais; 870.873 após limpeza; amostra de 100.000 na modelagem	trip_distance, duration_min, avg_speed_mph, passenger_count, coordenadas, horário, período e pagamento	Lucro estimado por hora; rentável acima da mediana

Fonte: Elaboração própria (2026).

O Quadro 1 compara os dois cenários experimentais e explicita suas diferenças metodológicas.

[2] Base disponível em: https://www.kaggle.com/datasets/yashdevladdha/uber-ride-analytics-dashboard/data?select=ncr_ride_bookings.csv. Acesso em: 16 jun. 2026.

[3] Base disponível em: https://www.kaggle.com/datasets/elemento/nyc-yellow-taxi-trip-data. Acesso em: 16 jun. 2026.

Quadro 1 – Síntese metodológica dos cenários experimentais

Aspecto	Cenário 1 - Uber/NCR	Cenário 2 - Yellow Taxi NYC
Tipo de cenário	Classificação com informação financeira estimada	Classificação baseada em eficiência operacional
Critério de rentabilidade	Lucro estimado por quilômetro	Lucro estimado por hora
Limiar da classe rentável	Percentil 70	Mediana
Distribuição da variável alvo	Aproximadamente 70% não rentável e 30% rentável	Aproximadamente balanceada entre as classes
Variáveis centrais	booking_value e ride_distance, associadas diretamente à regra de rentabilidade	duration_min, avg_speed_mph e trip_distance, associadas à eficiência da corrida
Principal limitação	Forte relação matemática entre variáveis de entrada e variável alvo	Uso de duração real, que em aplicação prática deveria ser estimada
Interpretação dos resultados	Avaliação da classificação segundo uma proxy financeira estimada	Avaliação do comportamento dos modelos com variáveis operacionais históricas

Fonte: Elaboração própria (2026).

A síntese dos cenários reforça que a pesquisa compara o comportamento dos algoritmos em contextos informacionais distintos. Assim, as métricas obtidas em cada base devem ser interpretadas dentro de seu respectivo cenário experimental, sem tratar os resultados como uma comparação direta de rentabilidade entre plataformas ou serviços diferentes.

3.2.3 Cuidados metodológicos contra vazamento de dados

Um cuidado importante adotado neste trabalho foi analisar o risco de vazamento de dados na escolha das variáveis utilizadas pelos modelos. Em AM, o vazamento ocorre quando o modelo recebe, durante o treinamento, informações que não estariam disponíveis no momento real da previsão ou que revelam diretamente a variável alvo.

Neste estudo, variáveis derivadas diretamente da resposta, como lucro estimado, lucro por quilômetro, lucro por hora e a própria classe de rentabilidade, não foram utilizadas como atributos preditores. No entanto, algumas variáveis financeiras exigiram tratamento específico, pois sua disponibilidade e seu significado diferem entre as bases analisadas.

Na base Uber/NCR, a variável booking_value foi mantida como entrada do modelo por representar, metodologicamente, uma aproximação do valor estimado da corrida informado ao motorista antes da aceitação da viagem. Essa decisão se justifica porque, em plataformas de transporte, o valor estimado é uma informação relevante para a tomada de decisão. Ainda assim, reconhece-se que essa variável possui relação direta

com o cálculo da rentabilidade, motivo pelo qual os resultados dessa base não devem ser interpretados como uma previsão independente do valor da corrida, mas como uma classificação apoiada também na informação financeira estimada.

Na base Yellow Taxi NYC, por outro lado, variáveis financeiras como fare_amount, tip_amount, tolls_amount, total_amount e profit foram removidas do modelo principal, pois representam valores consolidados após a corrida e estão diretamente associadas à construção da variável alvo. Dessa forma, buscou-se reduzir o risco de vazamento de dados e avaliar o desempenho dos modelos com base em variáveis operacionais, como distância, duração, horário, velocidade média e localização.

3.2.4 Compreensão e análise das bases de dados

3.2.4.1 Base Uber/NCR

A base Uber/NCR apresentou 150.000 registros iniciais. A coluna booking_status indicou diferentes situações de corrida, como Completed, Cancelled by Driver, No Driver Found, Cancelled by Customer e Incomplete. Como a análise de rentabilidade depende de valor e distância efetivamente associados à viagem, foram selecionadas apenas corridas concluídas, totalizando 93.000 registros nessa etapa.

Após a seleção das corridas concluídas, foram aplicados filtros básicos de consistência para remover registros sem valor de reserva, sem distância, com valores nulos ou valores inválidos. Também foram removidos outliers por meio do método IQR aplicado à distância, ao valor da corrida e à receita por quilômetro. Ao final, a base limpa utilizada como referência analítica ficou com 81.099 registros, após a remoção de 11.901 registros considerados ruídos ou valores extremos.

A Tabela 2 apresenta as principais etapas de limpeza e o número de registros mantidos na base Uber/NCR.

Tabela 2 – Etapas de limpeza da base Uber/NCR

Etapa	Quantidade
Total inicial de registros	150.000
Corridas concluídas	93.000
Base após remoção de outliers e ruídos	81.099
Registros removidos como ruído na etapa de outliers	11.901

Fonte: Elaboração própria (2026).

A variável de rentabilidade foi construída a partir de uma estimativa de custo por quilômetro. Como a base não informa o custo real do motorista, adotou-se um custo didático de 12 unidades monetárias por quilômetro. Esse valor foi adotado como parâmetro didático de modelagem, sem pretensão de representar um custo operacional real ou universal. Sua função foi permitir a construção de uma proxy de lucro estimado diante da ausência de informações reais sobre combustível, manutenção, taxa da plataforma e demais custos do motorista. O lucro estimado foi calculado pela diferença entre o valor da reserva e o custo estimado. Em seguida, calculou-se o lucro estimado por quilômetro.

Para transformar a rentabilidade em um problema de classificação binária, foram consideradas rentáveis as corridas com lucro por quilômetro igual ou superior ao percentil 70 da distribuição. O limiar obtido foi 11,35. Assim, aproximadamente 30% das corridas foram classificadas como rentáveis e 70% como não rentáveis. Essa escolha torna a classe rentável mais seletiva, evitando que qualquer lucro pequeno seja tratado como suficiente.

A Tabela 3 apresenta a distribuição final da variável alvo na base Uber/NCR.

Tabela 3 – Distribuição da variável alvo na base Uber/NCR

Classe	Quantidade	Proporção
Não rentável	56.769	70%
Rentável	24.330	30%

Fonte: Elaboração própria (2026).

3.2.4.2 Base Yellow Taxi NYC

A base Yellow Taxi NYC apresentou 887.273 registros iniciais. Diferentemente da base Uber/NCR, ela possui maior volume de dados e informações mais detalhadas sobre a corrida, incluindo horário de embarque e desembarque, distância, coordenadas geográficas, quantidade de passageiros, tarifa, gorjeta, pedágios e valor total.

Na etapa de limpeza, foram removidos registros com duração inferior a 3 minutos ou superior a 180 minutos, distância inválida, valores totais não positivos, passageiros inválidos, coordenadas fora da faixa esperada para Nova York e velocidades médias inconsistentes. Após esse processo, permaneceram 870.873 registros, com remoção de 16.400 registros, o equivalente a 1,85% da base inicialmente analisada.

A Tabela 4 sintetiza os principais indicadores estatísticos da base Yellow Taxi NYC após a limpeza.

Tabela 4 – Síntese estatística da base Yellow Taxi NYC após limpeza

Indicador	Valor
Registros iniciais	887.273
Registros após limpeza	870.873
Registros removidos	16.400 (1,85%)
Distância média da corrida	2,91 milhas
Duração média da corrida	12,86 minutos
Velocidade média	12,39 mph
Valor total médio	15,18

Fonte: Elaboração própria (2026).

Na base Yellow Taxi NYC, o custo operacional foi estimado com base em um custo de 0,655 por milha. Esse valor foi adotado como referência operacional para estimar custos de deslocamento, sendo utilizado apenas como aproximação metodológica. Como a base não contém o custo real de cada motorista, esse parâmetro foi empregado para viabilizar a construção da variável de lucro estimado. O lucro estimado foi calculado pela diferença entre o valor total da corrida e o custo estimado. Em seguida, foi calculado o lucro por hora, dividindo o lucro estimado pela duração da corrida em horas. O limiar de rentabilidade foi definido pela mediana do lucro por hora, resultando em 62,99.

Por usar a mediana como critério de separação, a base de modelagem ficou aproximadamente balanceada entre as classes rentável e não rentável. Para viabilizar o treinamento em tempo adequado, foi utilizada uma amostra reproduzível de 100.000 registros na modelagem, mantendo a distribuição da variável alvo.

A adoção de limiares diferentes nas duas bases foi uma escolha metodológica pragmática, ajustada à estrutura de cada cenário experimental. Na base Uber/NCR, o percentil 70 foi utilizado para tornar a classe rentável mais seletiva, concentrando a análise nas corridas de maior lucro estimado por quilômetro. Na base Yellow Taxi NYC, a mediana foi utilizada para manter as classes aproximadamente balanceadas e favorecer a comparação entre os modelos. Limiares alternativos poderiam ser avaliados em estudos futuros, mas a escolha adotada foi suficiente para construir uma variável alvo binária coerente com os objetivos deste trabalho.

3.2.5 Preparação dos dados

A etapa de preparação dos dados teve como objetivo transformar as bases em formato adequado para o treinamento dos modelos de AM. Essa etapa incluiu seleção de registros válidos, tratamento de valores ausentes, criação de variáveis derivadas, codificação de variáveis categóricas, padronização de variáveis numéricas e divisão dos dados em treino e teste.

Na base Uber/NCR, as variáveis temporais foram obtidas a partir das colunas de data e hora. Foram criadas variáveis como hour, day_of_week, month, is_weekend e period. Os locais de embarque e desembarque foram agrupados para reduzir excesso de categorias no OneHotEncoder, mantendo os locais mais frequentes e agrupando os demais como Outros. Essa estratégia evitou que o modelo ficasse excessivamente pesado devido a muitas categorias raras.

Na base Yellow Taxi NYC, foram criadas as variáveis duration_min, hour, day_of_week, is_weekend, rush_hour, period e avg_speed_mph. A variável month foi desconsiderada na modelagem por não apresentar variabilidade relevante na amostra analisada. As coordenadas de origem e destino foram mantidas como variáveis numéricas no modelo, pois podem representar padrões geográficos relacionados à rentabilidade. Variáveis financeiras usadas para construir a resposta foram excluídas das entradas do modelo principal.

As variáveis numéricas foram padronizadas com StandardScaler, enquanto as variáveis categóricas foram codificadas com OneHotEncoder. O uso de pipelines do scikit-learn (Pedregosa et al., 2011) permitiu aplicar o pré-processamento dentro do fluxo de treinamento, reduzindo risco de inconsistência entre treino e teste. Em variáveis com valores ausentes, foram aplicadas estratégias de imputação no notebook Uber/NCR, usando mediana para variáveis numéricas e valor mais frequente para categóricas.[4]

Os dados foram divididos em treino e teste com estratificação pela variável alvo, garantindo que a proporção das classes fosse preservada nas duas partes. Na base Uber/NCR, foi utilizada uma amostra reproduzível de 60.000 registros para a modelagem, dividida em 45.000 registros de treino e 15.000 registros de teste. Na base Yellow Taxi

[4] Documentação oficial: https://scikit-learn.org/. Acesso em: 16 jun. 2026.

NYC, a amostra de modelagem de 100.000 registros foi dividida em 80.000 registros para treino e 20.000 registros para teste.

A Tabela 5 apresenta as variáveis numéricas e categóricas utilizadas como entrada nos modelos.

Tabela 5 – Variáveis utilizadas como entrada nos modelos

Base	Variáveis numéricas	Variáveis categóricas
Uber/NCR	booking_value, ride_distance, avg_vtat, avg_ctat, driver_ratings, customer_rating, hour, day_of_week, month, is_weekend	vehicle_type, pickup_location_grouped, drop_location_grouped, period, payment_method
Yellow Taxi NYC	trip_distance, duration_min, avg_speed_mph, hour, day_of_week, is_weekend, rush_hour, passenger_count, pickup/dropoff longitude e latitude	payment_type, store_and_fwd_flag, period

Fonte: Elaboração própria (2026).

3.2.6 Modelagem

A modelagem foi realizada com algoritmos de classificação supervisionada. A escolha dos modelos buscou contemplar abordagens lineares, árvores de decisão, métodos de ensemble e uma rede neural simples para comparação. Também foi incluído um modelo baseline, com o objetivo de verificar se os modelos treinados realmente aprendiam padrões além de uma estratégia ingênua.

Na base Uber/NCR, foram treinados Baseline, Regressão Logística, Árvore de Decisão, Random Forest, Extra Trees, Gradient Boosting e TensorFlow/Keras. O Gradient Boosting foi implementado por meio do HistGradientBoostingClassifier. Devido à distribuição aproximada de 70%/30% da variável alvo, a Regressão Logística, a Árvore de Decisão e o Extra Trees utilizaram class_weight="balanced"; no Random Forest foi empregado class_weight="balanced_subsample". Essas configurações ajustam os pesos das classes durante o treinamento sem criar observações sintéticas ou remover registros da base.

Na base Yellow Taxi NYC, foram treinados Baseline, Regressão Logística, Árvore de Decisão, Random Forest, Extra Trees, AdaBoost e TensorFlow/Keras. O modelo TensorFlow/Keras foi composto por camadas densas, função de ativação ReLU nas camadas intermediárias e sigmoid na saída, por se tratar de classificação binária. A rede neural foi incluída apenas como experimento complementar.[5]

[5] Notebook utilizado nos experimentos: notebook_taxi_tcc_notebook (1).ipynb. Disponível em: https://colab.research.google.com/drive/19dKaoOFm41UCG55v4gBTHbfh9Yc8uWK9?usp=sharing.

As métricas utilizadas na avaliação foram Accuracy, Precision, Recall, F1-score e ROC-AUC. A acurácia representa a proporção geral de acertos. A precisão indica, entre as corridas previstas como rentáveis, quantas eram realmente rentáveis. O recall indica, entre as corridas realmente rentáveis, quantas foram identificadas pelo modelo. O F1-score combina precisão e recall, sendo especialmente útil quando há interesse em equilíbrio entre erros falsos positivos e falsos negativos. O ROC-AUC avalia a capacidade discriminatória do modelo considerando diferentes limiares de decisão.

Além das métricas estatísticas, a interpretação do problema também envolve impactos práticos diferentes para cada tipo de erro. Um falso positivo, isto é, classificar como rentável uma corrida que na prática não seria vantajosa, poderia gerar prejuízo direto ao motorista. Já um falso negativo, ao deixar de classificar como rentável uma corrida potencialmente boa, representa principalmente perda de oportunidade. Por esse motivo, métricas como precisão, recall e F1-score foram consideradas mais informativas do que apenas a acurácia, pois permitem avaliar melhor o equilíbrio entre os tipos de erro.

Os hiperparâmetros foram definidos antes da avaliação final por configuração manual controlada, orientada pelas recomendações das bibliotecas, pela literatura dos algoritmos e por testes exploratórios preliminares, buscando equilíbrio entre desempenho, tempo de execução e redução de sobreajuste. Não foi realizada busca automatizada ou otimização exaustiva. Após a seleção, os valores foram mantidos fixos para garantir comparabilidade e reprodutibilidade. De modo geral, a Regressão Logística utilizou limite de iterações entre 500 e 1000; a Árvore de Decisão utilizou profundidade máxima 8; Random Forest e Extra Trees utilizaram múltiplas árvores e tamanho mínimo de folhas; o Gradient Boosting utilizou taxa de aprendizado de 0,05 na base Uber/NCR; o AdaBoost utilizou taxa de aprendizado de 0,8 na base Yellow Taxi NYC; e o TensorFlow/Keras foi implementado como rede neural densa simples, com ativações ReLU e sigmoid, otimizador Adam e early stopping.

Além da avaliação no conjunto de teste, foi realizada validação cruzada para verificar a estabilidade dos modelos clássicos em diferentes divisões dos dados. Na base Uber/NCR, foi utilizada validação cruzada com três divisões, enquanto na base Yellow Taxi NYC foi utilizada validação cruzada estratificada com cinco divisões.

Na base Uber/NCR, também foi incluída uma comparação com uma regra determinística baseada na própria definição da variável alvo. Essa regra não representa um modelo de AM, mas permite demonstrar que, quando valor da corrida e distância estão disponíveis, a classe de rentabilidade estimada pode ser reconstruída diretamente pela fórmula definida metodologicamente. Essa etapa foi incorporada para apoiar a interpretação crítica dos resultados elevados obtidos nessa base.

3.3 Aspectos éticos e limitações

Esta pesquisa utilizou exclusivamente bases de dados disponibilizadas publicamente na plataforma Kaggle, sem interação direta com participantes e sem identificação individual dos usuários.

A principal limitação deste estudo está relacionada à ausência do lucro real do motorista nas bases utilizadas. Custos como combustível, manutenção, depreciação do veículo, taxa da plataforma, seguro, tempo de espera e deslocamento até o passageiro não estão integralmente disponíveis. Dessa forma, a rentabilidade foi tratada como uma estimativa construída a partir dos dados existentes.

Outra limitação refere-se ao uso de variáveis que podem não estar integralmente disponíveis antes da corrida. Na base Yellow Taxi NYC, por exemplo, duration_min foi calculada a partir dos horários reais de embarque e desembarque. Em uma aplicação anterior à aceitação da corrida, essa informação deveria ser substituída por uma estimativa de tempo fornecida por sistemas de navegação ou pela própria plataforma. A análise de sensibilidade sem duration_min e avg_speed_mph mostrou queda do F1-score do Random Forest de 0,9305 para 0,7735, indicando que essas variáveis têm influência relevante na classificação histórica. Embora o cenário restrito tenha mantido desempenho bem acima do baseline, ele reforça a importância de trabalhar com estimativas de tempo em aplicações reais. Assim, os resultados dessa base devem ser interpretados como classificação em dados históricos e não como simulação completa de decisão em tempo real.

Na base Uber/NCR, a variável booking_value foi utilizada como aproximação do valor estimado da corrida. Essa decisão é defensável em um contexto em que plataformas exibem valores ou estimativas ao motorista, mas também eleva o

desempenho do modelo por estar diretamente associada ao cálculo da rentabilidade. Por isso, os resultados dessa base devem ser interpretados como classificação baseada em informações financeiras estimadas, e não como previsão totalmente independente do valor da corrida. Essa limitação não invalida o experimento, mas delimita o tipo de conclusão que pode ser extraída dele.

Também é importante destacar que as bases possuem contextos distintos. A base Uber/NCR representa reservas de uma plataforma específica, enquanto a Yellow Taxi NYC representa corridas de táxi em Nova York. Assim, os resultados não devem ser generalizados automaticamente para todos os aplicativos, cidades ou períodos. A aplicação prática exigiria validação com dados locais e atualizados.

Por fim, este trabalho não implementa um sistema final para motoristas, nem realiza coleta em tempo real. O foco é acadêmico e comparativo, demonstrando como técnicas de AM podem ser aplicadas ao problema e quais cuidados devem ser considerados.

4 RESULTADOS E DISCUSSÃO

4.1 Resultados na base Uber/NCR

A Tabela 6 apresenta os resultados obtidos na base Uber/NCR. O modelo Gradient Boosting apresentou o melhor desempenho pelo F1-score, atingindo aproximadamente 0,99. A rede neural TensorFlow/Keras e a Regressão Logística também apresentaram resultados elevados, com F1-score de aproximadamente 0,99. O modelo baseline apresentou F1-score igual a zero para a classe rentável, indicando que a estratégia ingênua de prever apenas a classe majoritária não é adequada para o problema.

Esse desempenho elevado deve ser interpretado com cautela, pois a variável alvo foi construída a partir de uma relação matemática envolvendo o valor da reserva e a distância da corrida. Assim, os modelos, especialmente os mais flexíveis, tendem a aproximar o próprio critério utilizado para classificar a rentabilidade. Dessa forma, o resultado não deve ser interpretado como uma previsão independente da rentabilidade real, mas como uma classificação eficiente da rentabilidade estimada segundo a regra definida metodologicamente.

A Tabela 6 reúne as métricas obtidas pelos modelos no cenário principal da base Uber/NCR.

Tabela 6 – Desempenho dos modelos na base Uber/NCR

Modelo	Accuracy	Precision	Recall	F1-score	ROC-AUC
Gradient Boosting	1,00	0,99	0,99	0,99	1,00
TensorFlow/Keras	1,00	0,99	0,99	0,99	1,00
Regressão Logística	0,99	0,98	1,00	0,99	1,00
Árvore de Decisão	0,99	0,97	0,99	0,98	1,00
Random Forest	0,98	0,94	0,99	0,96	1,00
Extra Trees	0,95	0,86	0,99	0,92	1,00
Baseline	0,70	0,00	0,00	0,00	0,50

Fonte: Elaboração própria (2026).

O relatório de classificação do melhor modelo indicou desempenho equilibrado entre as classes. O Gradient Boosting obteve F1-score aproximado de 0,99 para a classe rentável, demonstrando elevada capacidade de separação no cenário construído. Entretanto, essa separação deve ser compreendida em conjunto com a forma de criação da variável alvo, baseada na relação entre valor, distância e custo estimado.

A validação cruzada confirmou a estabilidade dos modelos clássicos. O Gradient Boosting obteve F1 médio de 0,99, seguido pela Regressão Logística, também com F1 médio aproximado de 0,99. A Árvore de Decisão e o Random Forest apresentaram desempenho elevado, com F1 médio de 0,97, enquanto o Extra Trees obteve F1 médio de 0,92. O baseline manteve F1 igual a zero.

A Tabela 7 apresenta os resultados detalhados da validação cruzada dos modelos clássicos na base Uber/NCR.

Tabela 7 – Validação cruzada dos modelos clássicos na base Uber/NCR

Modelo	F1 médio CV	Desvio padrão
Gradient Boosting	0,9930	0,0004
Regressão Logística	0,9867	0,0005
Árvore de Decisão	0,9727	0,0022
Random Forest	0,9662	0,0020
Extra Trees	0,9221	0,0049
Baseline	0,0000	0,0000

Fonte: Elaboração própria (2026).

Os valores de desvio padrão não foram exatamente iguais a zero, exceto no baseline. Na versão anterior da tabela, eles apareciam como 0,00 devido ao arredondamento para duas casas decimais. A validação utilizou StratifiedKFold com três divisões, embaralhamento dos registros e random_state = 42. A semente fixa garante

reprodutibilidade, mas não torna as divisões idênticas; os valores detalhados são apresentados com quatro casas decimais na Tabela 7.

Na análise de importância das variáveis do modelo baseado em árvores, as variáveis mais relevantes foram ride_distance e booking_value. A distância apresentou importância de 0,52, enquanto o valor da reserva apresentou importância de 0,48. As demais variáveis tiveram importância próxima de zero, reforçando que o desempenho do cenário principal está fortemente associado à própria estrutura matemática utilizada para construir a rentabilidade estimada.

A análise de sensibilidade metodológica reforçou essa interpretação. No cenário principal, com booking_value e ride_distance, o melhor modelo manteve F1-score aproximado de 0,99. Ao remover booking_value, mantendo a distância, o melhor resultado no percentil 70 caiu para F1-score aproximado de 0,63. No cenário operacional restrito, sem booking_value e sem ride_distance, o F1-score caiu para aproximadamente 0,36. Esses resultados indicam que a informação financeira e a distância são centrais para a classificação da rentabilidade estimada na base Uber/NCR.

A Tabela 8 apresenta as variáveis de maior importância no modelo baseado em árvores da base Uber/NCR.

Tabela 8 – Principais variáveis na base Uber/NCR

Variável	Importância
ride_distance	0,52
booking_value	0,48
is_weekend	0,00
hour	0,00
period_tarde	0,00
period_noite	0,00
vehicle_type_Go Mini	0,00
avg_vtat	0,00
avg_ctat	0,00
day_of_week	0,00

Fonte: Elaboração própria (2026).

4.2 Resultados na base Yellow Taxi NYC

A Tabela 9 apresenta os resultados obtidos na base Yellow Taxi NYC. Nessa base, o modelo TensorFlow/Keras apresentou o maior F1-score no conjunto de teste, com 0,9326. Entre os modelos clássicos, o Random Forest obteve o melhor desempenho, com

F1-score de 0,9305 e ROC-AUC de 0,9820. A Árvore de Decisão e a Regressão Logística também apresentaram resultados competitivos.

Tabela 9 – Desempenho dos modelos na base Yellow Taxi NYC

Modelo	Accuracy	Precision	Recall	F1-score	ROC-AUC
TensorFlow/Keras	0,9326	0,9312	0,9340	0,9326	0,9830
Random Forest	0,9306	0,9296	0,9313	0,9305	0,9820
Árvore de Decisão	0,9217	0,9118	0,9334	0,9224	0,9792
Regressão Logística	0,9144	0,9031	0,9281	0,9154	0,9678
AdaBoost	0,9123	0,9308	0,8905	0,9102	0,9754
Extra Trees	0,8359	0,8394	0,8298	0,8346	0,9221
Baseline	0,5011	0,0000	0,0000	0,0000	N/A

Fonte: Elaboração própria (2026).

A interpretação do baseline na base Yellow Taxi NYC exige atenção. Como as classes ficaram aproximadamente balanceadas pela mediana do lucro por hora, o baseline que prevê a classe mais frequente atingiu cerca de 50% de acurácia, mas apresentou precisão, recall e F1-score iguais a zero para a classe rentável. Isso ocorreu porque a estratégia ingênua não identificou corridas rentáveis, reforçando a importância de comparar os modelos com métricas além da acurácia.

A validação cruzada dos modelos clássicos indicou que o Random Forest foi o modelo mais consistente, com F1 médio de 0,9279 e desvio padrão de 0,0020. A Árvore de Decisão obteve F1 médio de 0,9223, e a Regressão Logística obteve F1 médio de 0,9151. Esses resultados mostram que o desempenho não foi consequência apenas de uma divisão específica de treino e teste.

A Tabela 10 apresenta a validação cruzada dos modelos clássicos na base Yellow Taxi NYC.

Tabela 10 – Validação cruzada dos modelos clássicos na base Yellow Taxi NYC

Modelo	Accuracy médio	Precision médio	Recall médio	F1 médio	ROC-AUC médio	F1 desvio
Random Forest	0,9278	0,9247	0,9312	0,9279	0,9813	0,0020
Árvore de Decisão	0,9219	0,9156	0,9291	0,9223	0,9782	0,0018
Regressão Logística	0,9142	0,9041	0,9264	0,9151	0,9669	0,0004
AdaBoost	0,9085	0,9265	0,8870	0,9063	0,9743	0,0028
Extra Trees	0,8398	0,8380	0,8416	0,8398	0,9246	0,0084
Baseline	0,5011	0,0000	0,0000	0,0000	0,5000	0,0000

Fonte: Elaboração própria (2026).

A importância das variáveis do Random Forest na base Yellow Taxi NYC indicou que duration_min, avg_speed_mph e trip_distance foram os atributos mais relevantes. A duração da corrida apresentou importância de 0,4538, a velocidade média de 0,2052 e a

distância de 0,1397. Esse resultado é coerente com a definição de lucro por hora, uma vez que tempo e distância influenciam diretamente a eficiência financeira da corrida.

A Tabela 11 apresenta as variáveis de maior importância do Random Forest na base Yellow Taxi NYC.

Tabela 11 – Principais variáveis na base Yellow Taxi NYC

Variável	Importância
duration_min	0,4538
avg_speed_mph	0,2052
trip_distance	0,1397
payment_type_1.0	0,0522
payment_type_2.0	0,0486
hour	0,0234
pickup_longitude	0,0157
dropoff_longitude	0,0120
dropoff_latitude	0,0105
pickup_latitude	0,0075

Fonte: Elaboração própria (2026).

Como análise de sensibilidade operacional, foi treinado um cenário restrito na base Yellow Taxi NYC sem as variáveis duration_min e avg_speed_mph, que dependem da duração observada da corrida. Nesse cenário, o Random Forest obteve Accuracy de 0,7744, Precision de 0,7748, Recall de 0,7721, F1-score de 0,7735 e ROC-AUC de 0,8585. Os demais modelos também apresentaram queda de desempenho, como a Árvore de Decisão, com F1-score de 0,7508, o AdaBoost, com 0,7399, o Extra Trees, com 0,6732, e a Regressão Logística, com 0,6409.

Essa análise reforça que duração e velocidade média exercem influência importante sobre a classificação histórica da rentabilidade por hora. Ao mesmo tempo, o desempenho do cenário restrito não caiu ao nível do baseline, indicando que ainda existem sinais úteis em variáveis como distância, horário, localização e forma de pagamento. Portanto, a base Yellow Taxi NYC deve ser interpretada como um cenário histórico de eficiência operacional, e sua aplicação prática exigiria a substituição de duração e velocidade reais por estimativas anteriores à corrida.

4.3 Discussão comparativa dos cenários experimentais

Os resultados das duas bases indicam que modelos de AM conseguem identificar padrões associados à rentabilidade estimada das corridas. Entretanto, a análise deve ser

compreendida como comparação do comportamento dos modelos em dois cenários experimentais distintos, e não como comparação direta entre datasets equivalentes. Isso ocorre porque a variável alvo foi construída de forma diferente em cada base: lucro estimado por quilômetro na Uber/NCR e lucro estimado por hora na Yellow Taxi NYC. Para fins de interpretação, a Uber/NCR é tratada como Cenário 1, associado à classificação com informação financeira estimada, enquanto a Yellow Taxi NYC é tratada como Cenário 2, associado à classificação baseada em eficiência operacional.

Dessa forma, os valores de desempenho, como o F1-score aproximado de 0,99 na base Uber/NCR e 0,93 na base Yellow Taxi NYC, não devem ser interpretados como uma disputa direta entre as bases. Cada resultado expressa a capacidade dos modelos de classificar a rentabilidade estimada segundo o critério metodológico definido para aquele conjunto de dados.

Na base Uber/NCR, o desempenho foi muito elevado, com F1-score próximo de 0,99 para o melhor modelo. Esse resultado está fortemente relacionado à presença das variáveis booking_value e ride_distance, que são diretamente associadas ao cálculo do lucro por quilômetro. Assim, o resultado demonstra que o modelo identifica com eficiência a relação entre valor, distância e rentabilidade, mas não deve ser lido como prova de que a rentabilidade pode ser prevista sem informação financeira. A interpretação correta é que, nessa base, o modelo classifica bem as corridas quando dispõe de uma aproximação do valor estimado informado pela plataforma. A comparação com a regra determinística e com os cenários sem booking_value evidenciou que parte expressiva do desempenho decorre da reconstrução do próprio critério de classificação.

Embora a rentabilidade estimada possa ser calculada diretamente quando valor e distância estão disponíveis, o uso de modelos de AM neste trabalho teve finalidade comparativa e analítica. O objetivo não foi substituir uma fórmula determinística simples, mas avaliar como diferentes algoritmos se comportam diante de variáveis operacionais e financeiras disponíveis nas bases, bem como identificar a importância relativa dos atributos e os limites de generalização dos modelos.

Na base Yellow Taxi NYC, o desempenho também foi elevado, mas com maior equilíbrio metodológico, pois variáveis financeiras diretamente ligadas ao alvo foram excluídas do modelo principal. Ainda assim, duração, velocidade e distância continuam

sendo atributos muito relacionados ao cálculo de rentabilidade por hora. Isso mostra que modelos de classificação podem aproveitar relações operacionais fortes, mas também reforça que a previsão antes da corrida dependeria de estimativas confiáveis de tempo e distância.

Na base Yellow Taxi NYC, a rede neural implementada com TensorFlow/Keras apresentou o maior F1-score no conjunto de teste, com desempenho ligeiramente superior ao Random Forest. Entretanto, a diferença entre os modelos foi pequena, indicando resultados competitivos. O Random Forest, embora não tenha sido o melhor no teste isolado, apresentou desempenho muito próximo e se destacou na validação cruzada entre os modelos clássicos, além de permitir análise da importância das variáveis. Dessa forma, a rede neural foi considerada o melhor resultado pontual na base Taxi, enquanto o Random Forest foi tratado como modelo clássico de referência pela combinação entre desempenho, estabilidade e possibilidade de interpretação.

5 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo analisar e comparar modelos de AM para classificar corridas segundo critérios de rentabilidade estimada. Para isso, foram utilizadas duas bases de dados: a base Uber/NCR e a base Yellow Taxi NYC. A metodologia foi organizada conforme as etapas do CRISP-DM, contemplando compreensão do problema, compreensão dos dados, preparação, modelagem e avaliação.

Como os critérios de rentabilidade foram construídos de forma distinta, os resultados devem ser avaliados dentro de cada base, e não como uma disputa direta entre os datasets. A diferença entre lucro por quilômetro e lucro por hora decorre da estrutura informacional de cada base e representa uma adaptação metodológica ao nível de dados disponível em cada cenário.

Considerando os resultados dentro de cada cenário experimental, o Gradient Boosting destacou-se na base Uber/NCR, enquanto o TensorFlow/Keras obteve o melhor desempenho pontual na base Yellow Taxi NYC. No entanto, o Random Forest apresentou desempenho competitivo e consistente, sendo adotado como modelo clássico de referência por combinar boa performance, estabilidade em validação cruzada e

possibilidade de análise da importância das variáveis. Na base Uber/NCR, o alto desempenho deve ser interpretado em função da reconstrução do critério de rentabilidade estimada, especialmente pela presença de booking_value e ride_distance. Já na base Yellow Taxi NYC, o TensorFlow/Keras alcançou F1-score de 0,9326 no conjunto de teste, enquanto o Random Forest obteve 0,9305 no teste e 0,9279 na validação cruzada, mantendo resultado muito próximo e maior apoio interpretativo entre os modelos clássicos.

O Quadro 2 sintetiza os principais resultados e a interpretação adotada em cada cenário experimental.

Quadro 2 – Síntese consolidada dos principais resultados

Base	Melhor modelo pontual	Modelo de referência	Observação
Uber/NCR	Gradient Boosting	Gradient Boosting/Random Forest	Alto desempenho dependente de booking_value e ride_distance
Yellow Taxi NYC	TensorFlow/Keras	Random Forest	Desempenho próximo; cenário restrito com F1 = 0,7735

Fonte: Elaboração própria (2026).

Conclui-se que o uso de inteligência artificial nesse contexto é promissor, desde que os modelos sejam compreendidos como ferramentas que classificam corridas segundo critérios de rentabilidade estimada, e que sua aplicação seja acompanhada de cuidados metodológicos. A criação da variável alvo, a escolha dos atributos preditores, a identificação de possíveis riscos de vazamento de dados e a análise de cenários alternativos são etapas essenciais para que os resultados sejam confiáveis. Além disso, a interpretabilidade deve ser valorizada, especialmente em problemas aplicados à tomada de decisão.

Como trabalhos futuros, recomenda-se utilizar bases com informações mais completas, incluindo valor estimado antes da corrida, tempo estimado, tarifa dinâmica, tempo de espera, custo real do motorista, preço de combustível, clima, eventos locais, demanda por região e deslocamento até o passageiro. Também seria possível desenvolver uma aplicação prática capaz de sugerir corridas ou horários com maior probabilidade de rentabilidade, desde que alimentada por dados atualizados e representativos.

REFERÊNCIAS

ABADI, Martín et al. *TensorFlow: large-scale machine learning on heterogeneous systems*. 2016. Disponível em: https://www.tensorflow.org/. Acesso em: 25 maio 2026.

ANTUNES, Ricardo. *O privilégio da servidão: o novo proletariado de serviços na era digital*. São Paulo: Boitempo, 2020.

BREIMAN, Leo. Random forests. *Machine Learning*, v. 45, p. 5-32, 2001.

CHAPMAN, Pete et al. *CRISP-DM 1.0: step-by-step data mining guide*. SPSS, 2000.

DAVENPORT, Thomas H.; HARRIS, Jeanne G. *Competing on analytics: the new science of winning*. Boston: Harvard Business School Press, 2007.

ELEMENTO. *NYC Yellow Taxi Trip Data*. Kaggle, s.d. Disponível em: https://www.kaggle.com/datasets/elemento/nyc-yellow-taxi-trip-data. Acesso em: 16 jun. 2026.

FREUND, Yoav; SCHAPIRE, Robert E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, v. 55, n. 1, p. 119-139, 1997. DOI: https://doi.org/10.1006/jcss.1997.1504.

FRIEDMAN, Jerome H. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics*, v. 29, n. 5, p. 1189-1232, 2001.

GEURTS, Pierre; ERNST, Damien; WEHENKEL, Louis. Extremely randomized trees. *Machine Learning*, v. 63, p. 3-42, 2006. DOI: https://doi.org/10.1007/s10994-006-6226-1.

HAN, Jiawei; KAMBER, Micheline; PEI, Jian. *Data mining: concepts and techniques*. 3. ed. Waltham: Morgan Kaufmann, 2011.

HASTIE, Trevor; TIBSHIRANI, Robert; FRIEDMAN, Jerome. *The elements of statistical learning: data mining, inference, and prediction*. 2. ed. New York: Springer, 2009.

LADDHA, Yashdev. *Uber Data Analytics Dashboard*. Kaggle, s.d. Disponível em: https://www.kaggle.com/datasets/yashdevladdha/uber-ride-analytics-dashboard. Acesso em: 16 jun. 2026.

MANGUINHO, Maria Letícia da Silva. *Análise comparativa de redes neurais artificiais e algoritmos de ensemble para a predição de risco de crédito*. Trabalho de Conclusão de Curso (Análise e Desenvolvimento de Sistemas) - Instituto Federal de Pernambuco, Campus Paulista, 2023.

MITCHELL, Tom M. *Machine learning*. New York: McGraw-Hill, 1997.

NEW YORK CITY TAXI AND LIMOUSINE COMMISSION. *TLC Trip Record Data*. New York: NYC TLC, 2015. Disponível em: https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page. Acesso em: 16 jun. 2026.

PEDREGOSA, Fabian et al. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825-2830, 2011.

SANTOS, Mariana Lovato dos. *Algoritmos de aprendizado de máquina supervisionados aplicados em transportes: comparativo com modelo Logit Multinomial para escolha modal*. 2023. Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2023.

SUNDARARAJAN, Arun. *The sharing economy: the end of employment and the rise of crowd-based capitalism*. Cambridge: MIT Press, 2016.

TAN, Pang-Ning; STEINBACH, Michael; KUMAR, Vipin. *Introduction to data mining*. Boston: Pearson, 2009.

APÊNDICE A – SÍNTESE DAS VARIÁVEIS UTILIZADAS

Este apêndice apresenta uma síntese das principais variáveis utilizadas nos notebooks finais que serviram de base para a escrita deste trabalho.

A Tabela 12 sintetiza as principais variáveis de entrada e variáveis derivadas utilizadas na pesquisa.

Tabela 12 – Síntese das variáveis utilizadas

Variável	Base	Descrição
booking_value	Uber/NCR	Valor da reserva ou corrida, utilizado como aproximação do valor estimado apresentado pela plataforma.
ride_distance	Uber/NCR	Distância da corrida utilizada para estimar custo e lucro por quilômetro.
avg_vtat	Uber/NCR	Tempo médio relacionado à chegada do veículo, utilizado como variável operacional.
avg_ctat	Uber/NCR	Tempo médio associado ao atendimento/chegada ao cliente, utilizado como variável operacional.
vehicle_type	Uber/NCR	Tipo de veículo solicitado na corrida.
payment_method	Uber/NCR	Forma de pagamento registrada.
trip_distance	Yellow Taxi NYC	Distância percorrida na corrida em milhas.
duration_min	Yellow Taxi NYC	Duração da corrida em minutos, calculada pela diferença entre embarque e desembarque.
avg_speed_mph	Yellow Taxi NYC	Velocidade média da corrida em milhas por hora.
passenger_count	Yellow Taxi NYC	Quantidade de passageiros registrada na corrida.
payment_type	Yellow Taxi NYC	Tipo de pagamento da corrida.
period	Ambas	Período do dia, categorizado em manhã, tarde, noite ou madrugada.
is_profitable	Ambas	Variável alvo binária, indicando corrida rentável ou não rentável.

Fonte: Elaboração própria (2026).