

ML INVEST: APP PARA AUXILIAR NA COMPARAÇÃO E SELEÇÃO DE AÇÕES PARA INVESTIMENTOS UTILIZANDO APRENDIZAGEM DE MÁQUINA

DOUGLAS LUIZ SCHNEIDER¹

ANDRÉ LUIS MOREIRA²

Resumo

Este trabalho produziu um app para auxiliar na escolha de ações de usuários com pouca ou nenhuma experiência, que exige conhecimentos técnicos para verificar em qual momento é uma boa opção de compra ou venda. Aborda-se o tema utilizando algoritmos de aprendizagem de máquina, por meio da biblioteca do TensorFlow e utilização de Flutter para o desenvolvimento do aplicativo, além de utilizar framework Flask e do NodeJS para a construção de API. Também utiliza-se de banco de dados para armazenamento de funcionalidades do aplicativo.

Palavras-chave: TensorFlow. Aprendizagem de Máquina. Bolsa de Valores. ML Invest.

1 INTRODUÇÃO

O mercado de investimentos vem crescendo de uma forma significativamente grande, no Brasil os investimentos ainda é um mercado em que a maioria da população se mantém distante, mas mesmo assim é um mercado em crescente avanço. “O volume financeiro investido pelos brasileiros em títulos e valores mobiliários atingiu R\$2 trilhões no primeiro

trimestre de 2022, recorde da série histórica da Associação Brasileira das Entidades dos Mercados Financeiro e de Capitais.” (CAMARGO, 2022).

O mercado de ações têm grande influência no crescimento do Brasil, tanto para quem investe, como para as empresas, dado que com o dinheiro injetado pelos investimentos é possível que as mesmas possam expandir suas atividades, sendo assim desenvolvendo mais a economia do país.

Atualmente, para ter alguma previsão se a ação de uma empresa pode subir ou descer, é necessário calcular o passado da trajetória do preço, onde para uma pessoa leiga isso é bastante difícil, já que é necessário um bom conhecimento matemático. Além da necessidade de estudar o mercado, algo que o cidadão comum, em sua maioria, não dispõe de tempo.

O trabalho aborda o problema utilizando a inteligência artificial para realizar previsões futuras do preço das ações. Isso traz uma segurança maior para o usuário, já que ele pode fazer o investimento de forma mais segura e sem a necessidade de conhecimentos técnicos para prever se a ação está em um bom momento para compra ou venda, onde o próprio aplicativo traz o histórico da ação e prevê o seu preço mais adiante.

O artigo se divide em fundamentação teórica, onde é abordado sobre renda variável e renda fixa, em Diferenças entre Renda Fixa e Variável, e aprendizagem de máquina, na seção Aprendizagem de Máquina e Tipos de Aprendizado, materiais e métodos, onde descreve sobre as ferramentas, banco de dados e linguagens utilizadas no projeto e discussão e resultados, discutindo os aspectos da aplicação, nas seções aplicação ML Invest, backend e gerando modelos, e os resultados da pesquisa, na seção resultados.

2 DESENVOLVIMENTO

Atualmente no mercado de investimentos existem diversas formas de investir, sendo elas a renda fixa e a renda variável. Nesta seção são apresentados os conceitos que fundamentam o trabalho.

2.1 DIFERENÇAS ENTRE RENDA FIXA E VARIÁVEL

Na renda fixa, a remuneração é definida no momento da aplicação. Ao comprar um título, o comprador empresta o dinheiro para o emissor do título, esperando receber esse valor de volta no futuro acrescido de juros e sua condição é definida desde o início da compra (RENDA ..., [201-?]).

Os investimentos podem ser classificados em três grupos, sendo eles:

Os títulos prefixados, onde a taxa não se altera ao longo do tempo, por exemplo, o Certificado de Depósito Bancário (CDB), Letras de Câmbio (LC) e o Tesouro Prefixado (PACHECO, 2022).

Nos títulos pós-fixados o título rende de acordo com a variação de algum indicador, por exemplo, Certificado de Depósito Interbancário (CDI) ou Inflação (O..., 2020).

E os títulos híbridos, que combinam a renda fixa e a variável, onde, por exemplo, rende um percentual fixo e mais um indicador, como o Índice Nacional de Preços ao Consumidor Amplo (IPCA). Podendo ser citados as Debêntures, Tesouro Direto IPCA+, Letra de Crédito Imobiliário (LCI), Letra de Crédito Agrícola (LCA), CDB, Certificado de Recebíveis Imobiliários (CRI) e Certificado de Recebíveis do Agronegócio (CRA) (PACHECO, 2022).

Na renda variável existem diversos investimentos, que vão de ações, o investimento mais conhecido, até opções. Ela não tem garantia de um

ganho fixo, como na renda fixa, e a devolução do valor aplicado, variando positivamente ou negativamente (O..., 2020).

As ações são negociadas pela Bolsa de Valores. Neste investimento, o investidor ganha com a valorização e perde se houver queda nos preços. As ações também pagam a distribuição de dividendos, que é parte do lucro da empresa distribuído aos acionistas (PACHECO, 2022).

A alocação em ações e fundos de ações em julho de 2022 ficou em 12% de participação, um dos menores percentuais do ano na distribuição das carteiras de investimentos (LANZA, 2022).

Fundos de Investimentos também permitem investir em renda variável. Neste tipo de investimento destacam-se dois tipos de fundos que têm investimentos em renda variável, os Fundos de Ações e Fundos Multimercado. Fundos de ações são um dos exemplos mais comuns. Aplica-se dois terços do patrimônio em ações que são negociadas na Bolsa de Valores. Os Fundos Multimercado investem em vários segmentos, como renda fixa, renda variável e moedas (RENDA ..., [20--]).

Opções é um tipo de investimento que garante o direito da compra ou venda de um ativo, geralmente ações em uma data futura com valor predeterminado (PACHECO, 2022).

Exchange Traded Funds (ETFs) são fundos onde são reaplicados a composição de índices financeiros, com as cotas sendo negociadas na Bolsa de Valores e um dos atrativos está em permitir para o investidor ter uma carteira de ações com diferentes empresas (PACHECO, 2022). Os ETFs permitem investir em várias ações de uma vez só e também seu custo é reduzido se comparado com os fundos de ações (RENDA..., [20--]).

Outro investimento em renda variável são os Fundos Imobiliários (FIIs), que são fundos que investem o dinheiro no mercado imobiliário, vendendo

partes do Fundo, chamados de cotas. Com o dinheiro obtido dessas cotas, os fundos podem adquirir ou construir imóveis que ao final podem ser alugados ou vendidos. A remuneração é paga conforme a proporção aplicada e elas são negociadas na Bolsa de Valores (PACHECO, 2022).

Como o mercado de renda variável é complexo, começou a utilizar-se de programação e inteligência artificial para auxiliar os investidores, principalmente os iniciantes. O uso do aprendizado de máquina torna a experiência do investidor melhor e tendo previsões mais claras para o ele, sendo “[...] Assim, eles conseguiriam antecipar algumas ações e maximizar o seu retorno financeiro.” (O..., 2023).

2.2 APRENDIZAGEM DE MÁQUINA

Atualmente a tecnologia de aprendizagem de máquina vem sendo largamente usada entre diversos ramos, para diversos fins. Esta tecnologia tem como objetivo automatizar processos de uma forma mais eficiente do que a técnica de programação de algoritmos normal.

Estes sistemas inteligentes possuem a capacidade de compreender variadas habilidades, de diversos ramos, podendo-se trabalhar com grande volumes de dados. Para Lenz e outros (2020), aprendizagem de máquina consiste em treinamentos com grandes volumes de dados, onde são relacionados dados de entradas com os dados de saídas, sendo assim reconhecendo padrões de forma autônoma e desta forma é possível até mesmo antecipar situações.

2.2.1 Tipos de Aprendizado

Aprendizado supervisionado, tem como objetivo relacionar os dados de entrada e saída, no treinamento é fornecido exemplos com a saída de dados desejadas, geralmente este método é utilizado para realizar classificação, ou seja atribuir rótulos, Um exemplo que se pode citar é classificação se um e-mail é um spam, ou um e-mail normal. Porém, os rótulos podem ser variáveis, o que caracteriza um problema de regressão. (TAN; STEINBACH; KUMAR, 2006).

Aprendizado não supervisionado, este método tem como objetivo descobrir classes ou mesmo padrões com os dados de entradas sem definir rotulações, basicamente o método cria agrupamentos ou associações entre os dados de entrada.

Aprendizado semi-supervisionado, esse método consiste em utilizar exemplos rotulados e não rotulados. (HAN; PEI; KAMBER, 2011).

Aprendizado por reforço, este método consiste em fazer tentativas e erros, e com isso fornecer recompensas e punições pelas ações feitas, e com isso gerar uma melhor sequência de ações.

3 MATERIAIS E MÉTODOS

Para desenvolvimento do aplicativo, foi utilizado o Git, Github, Google Colab, Flutter, NodeJS, Python, Flask, entre outros para a construção do aplicativo e funcionamento correto da aplicação.

O Git foi utilizado para o versionamento do código da aplicação. Aliado ao Git, o Github foi utilizado para a hospedagem deste código, facilitando o desenvolvimento da aplicação.

Para a construção do algoritmo de aprendizagem de máquina, foi utilizado a linguagem Python com as bibliotecas pandas, numpy e TensorFlow, para que fosse possível a criação e treinamento de modelos para cada ação selecionada, utilizando a arquitetura LSTM (Long Short-Term Memory). Para os preços das ações, foi utilizado a biblioteca do Yahoo Finance, sendo possível ter os dados de preços do início do pregão, do final do pregão, a mínima e a máxima das ações utilizadas para o aplicativo.

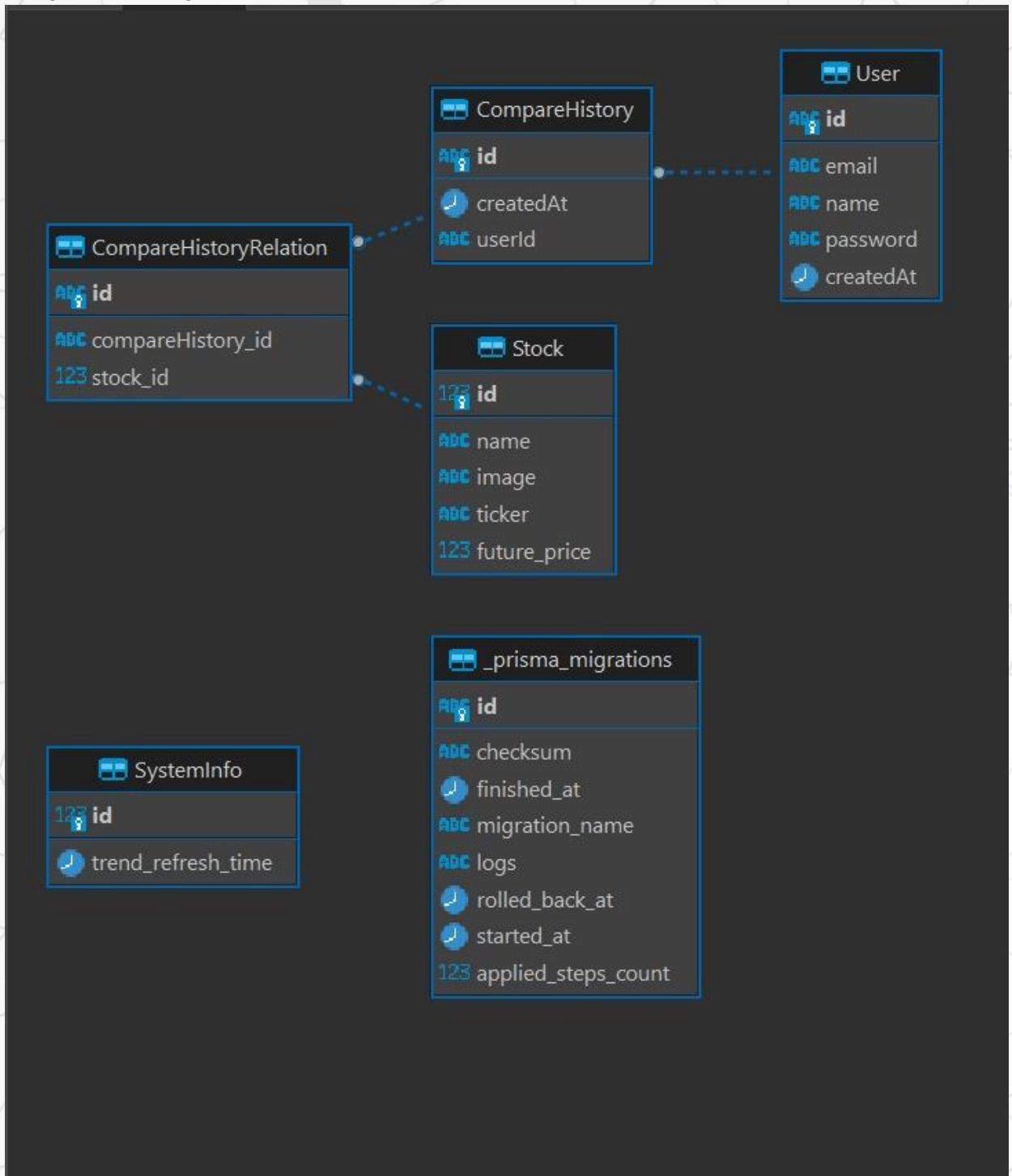
Utilizando a aprendizagem supervisionada, teve a possibilidade de gerar modelos de cada ação utilizada no aplicativo. Os modelos gerados utilizam-se do algoritmo K-Fold Cross-validation para verificação e a plataforma Google Colab, que auxiliam no processo de geração e treinamento, já que o algoritmo e a biblioteca TensorFlow utilizam-se de GPU para suportar a carga de trabalho.

Os modelos foram gerados pela API Keras, que é uma “[...] API de alto nível do TensorFlow para criar e treinar modelos de aprendizado profundo.[...]” (KERAS, 2020). Foram gerados, ao todo, 14 modelos de ações, que foram exportados adicionados ao projeto de back-end do aplicativo, sendo necessário para calcular o preço futuro.

Para que o aplicativo possa informar o preço futuro para o usuário, foi criada uma API em Python, utilizando o Flask, que chama o modelo da ação selecionada no aplicativo e baixa os dados atualizados da biblioteca Yahoo Finance, retornando o preço futuro da ação.

Outra parte do backend foi desenvolvido utilizando NodeJS. Nela contém as informações de histórico da ação, histórico comparativo da ação, histórico de preços futuros e também o login e senha de usuário, para cadastro no aplicativo. Todas essas informações são salvas utilizando o banco de dados CockroachDB.

Diagrama 1: Diagrama do banco de dados



Fonte: Os autores (2023).

Como é observado no diagrama 1, o banco tem todas as tabelas interconectadas, em especial a tabela CompareHistoryRelation. Essa tabela faz a relação entre a tabela Stock, que contém os dados da ação, e a tabela CompareHistory, que indica qual usuário fez essa comparação e em

qual dia e horário, sendo possível a funcionalidade de histórico de comparações dentro do aplicativo.

Para desenvolvimento do aplicativo, foi utilizado a linguagem Flutter. O aplicativo faz a leitura da API para apresentar os histórico de preços das ações e informar quais tem tendência de alta, queda ou de manter o preço, além de apresentar um pequeno gráfico com o histórico dessa ação e o próprio usuário escolhendo o período.

Também é possível realizar a comparação das ações no aplicativo, podendo ser útil para a tomada de decisão do usuário no momento da compra.

Para realizar a hospedagem da API, foi utilizada a plataforma Render. A plataforma é possível disponibilizar web apps e sites estáticos conectando ao Github para realizar deploy de projetos. A plataforma conta com suporte a diversas linguagens, entre elas Python e NodeJS.

4 ML INVEST

Nesta seção serão discutidos os aspectos da aplicação, assertividade do algoritmo utilizado e resultados dos testes realizados.

4.1 APLICAÇÃO

O aplicativo tem como principal função a escolha entre algumas ações para sugerir ações baseadas em algoritmos de aprendizagem de máquina. Essa sugestão necessita de dados dos preços das ações como preço no início do pregão, preço ao final do pregão, mínima e máxima do dia, contando o preço de vários dias.

Uma das funcionalidades da aplicação é comparar o preço entre as ações, basta apenas selecionar as ações desejadas e realizar a comparação. Feita a comparação, o aplicativo mostra um gráfico com os preços das ações comparadas e suas tendências.

Tela 1: Comparativo de ações



Fonte: Os autores (2023).

O usuário pode realizar um cadastro para armazenar um histórico de comparativo de ações para que possam ser acessados posteriormente, consultando os preços e também as tendências dessas ações.

Tela 2: Tela inicial do aplicativo



Fonte: Os autores (2023)

O usuário pode optar por apenas verificar as tendências das ações oferecidas pelo aplicativo na tela inicial, sem a necessidade de criar uma conta, como mostra a figura acima.

Listagem de código 1: Comunicação do aplicativo com o backend

```
class StockService {
    late HttpClient _http;

    StockService() {
        _http = HttpClient();
    }

    static String url = "${ApiUrl.url}/stock";

    Future<List<StockModel>> getAllStocks() async {
        try {
            List<dynamic> response = await _http.request(HttpRequestEnum.get, url);

            var list = response.map((e) => StockModel.fromJson(e)).toList();

            return list;
        } catch (error) {
            ErrorHandler.handleError(error);
            return [];
        }
    }
}
```

Fonte: Os autores (2023)

O aplicativo realiza uma requisição na api, informando a rota stock, onde retorna um JSON com todas as ações que o backend possui, para que possa ser visualizadas pelo usuário. Esse JSON possui as informações como nome, ticker da ação, logo, preço de alto do dia, preço de baixa do dia, preço médio e, principalmente, o preço futuro.

4.2 BACKEND

A API está dividida em duas partes, a primeira parte é desenvolvida em NodeJS, onde ela se comunica diretamente com o aplicativo, retornando todas as informações das ações definidas e também com o login com o sistema para realizar as comparações de ações.

A API desenvolvida em NodeJS, faz a comunicação com a outra parte da API desenvolvida com Flask. Ela utiliza a própria URL para ter acesso aos modelos gerados.

Listagem de código 2: Comunicação da API em NodeJS com a API em Flask

```
@Injectable()
export class TrendStockService {
  constructor(private prisma: PrismaService) {}

  async refreshData() {
    const stocks = await this.prisma.stock.findMany();

    for (const stock of stocks) {
      Logger.log(`refreshing ${stock.ticker} `);

      const response = await axios.get<TrendStockRequestDTO>(
        process.env.API_URL_TREND + '/metrics',
        {
          params: {
            ticker: stock.ticker,
          },
        },
      );

      Logger.log(`received data:`, [response.data]);

      const future_price = response.data.future_price;
```

Fonte: Os autores (2023)

Pela imagem podemos verificar que essa parte da API realiza uma chamada Get no ponto de acesso /metrics, passado como parâmetro o ticker da ação, é retornada a resposta do preço futuro da ação, para então poder informar os dados completos da ação ao usuário final.

Listagem de código 3: API em Flask

```
# Run API
app = Flask(__name__)

@app.route('/metrics', methods = ['GET'])
def price_future():
    ticker = request.args.get('ticker')

    # Carrega o modelo
    model_path = os.path.join("archives/results", ticker) + ".h5"
    model = create_model(N_STEPS, len(FEATURE_COLUMNS), units=UNITS, cell=CELL, n_layers=N_LAYERS, dropout=DROPOUT, bidirectional=BIDIRECTIONAL)
    model.load_weights(model_path)

    data_ticker = load_data(ticker, n_steps=N_STEPS, lookup_step=LOOKUP_STEP)

    future_price = float(predict(model, data_ticker))

    return jsonify({
        'future_price' : future_price
    })
```

Fonte: Os autores (2023)

A outra parte da API desenvolvida em Python, utilizando o framework Flask, tem apenas uma única rota, que é a metrics. Ela necessita do ticker da ação para poder carregar o modelo na variável model, essa variável é criada anteriormente com a função create_model, para que então possa ser carregado o modelo correto.

4.3 GERANDO OS MODELOS DE APRENDIZAGEM

Após, é necessário que os dados sejam atualizados para que o preço futuro seja calculado pelo modelo, para então ser devolvido em JSON com o valor desejado.

Para gerar os modelos, foi utilizado a linguagem Python e as bibliotecas do TensorFlow, para criar e treinar o modelo, e do Yahoo Finance, para obter os dados das ações. Os dados são baixados pelo código e salvos em data frames, para serem utilizados posteriormente.

Listagem de código 4: Cria o modelo para cada ação

```
def create_model(sequence_length, n_features, units=256, cell=LSTM, n_layers=2, dropout=0.3,
                 loss="mean_absolute_error", optimizer="rmsprop", bidirectional=False):
    model = Sequential()
    for i in range(n_layers):
        if i == 0:
            # first layer
            if bidirectional:
                model.add(Bidirectional(cell(units, return_sequences=True), batch_input_shape=(None, sequence_length, n_features)))
            else:
                model.add(cell(units, return_sequences=True, batch_input_shape=(None, sequence_length, n_features)))
        elif i == n_layers - 1:
            # last layer
            if bidirectional:
                model.add(Bidirectional(cell(units, return_sequences=False)))
            else:
                model.add(cell(units, return_sequences=False))
        else:
            # hidden layers
            if bidirectional:
                model.add(Bidirectional(cell(units, return_sequences=True)))
            else:
                model.add(cell(units, return_sequences=True))
    # add dropout after each layer
    model.add(Dropout(dropout))
    model.add(Dense(1, activation="linear"))
    model.compile(loss=loss, metrics=["mean_absolute_error"], optimizer=optimizer)
    return model
```

Fonte: Os autores (2023)

Os modelos são gerados pela função `create_model`, onde são repassados diversos parâmetros para que o modelo consiga ser treinado adequadamente.

Como é mostrado na figura acima, a função conta com três parâmetros importantes, que são eles `units`, que indica a quantidade de neurônios para o modelo de aprendizagem, `cell`, que informa qual o algoritmo da biblioteca TensorFlow será utilizado, e `dropout`, a porcentagem de erro que o algoritmo aceita para aceitar ou descartar o modelo. Mesmo elas contenham um valor padrão podem ser configuradas externamente a função, passando valores diferentes nas chamadas da função.

Ao criar o modelo, são repassados para os parâmetros `units` a quantidade de 256, o `dropout` de quantidade de 40% (0.4) e a utilização do algoritmo LSTM, em `cell`. Com isso, o modelo consegue uma melhor precisão em relação ao preço real da ação negociada na bolsa de valores.

Listagem de código 5: Variáveis utilizadas na criação dos modelos

```
# LSTM cell  
CELL = LSTM  
# 256 LSTM neurons  
UNITS = 256  
# 40% dropout  
DROPOUT = 0.4
```

Fonte: Os autores (2023)

O modelo começa a ser criado após ser iniciado um range do número de camadas das colunas utilizadas, obtidas da biblioteca do Yahoo Finance, e então o algoritmo LSTM adiciona ao modelo a quantidade de neurônios desejada e informa que pode retornar uma sequência passando uma variável booleana, apenas para a primeira camada e as camadas vazias. Após esse processo, o algoritmo adiciona o dropout, compila e retorna modelo para então ser treinado com os dados baixados para cada ação.

4.4 RESULTADOS E DISCUSSÃO

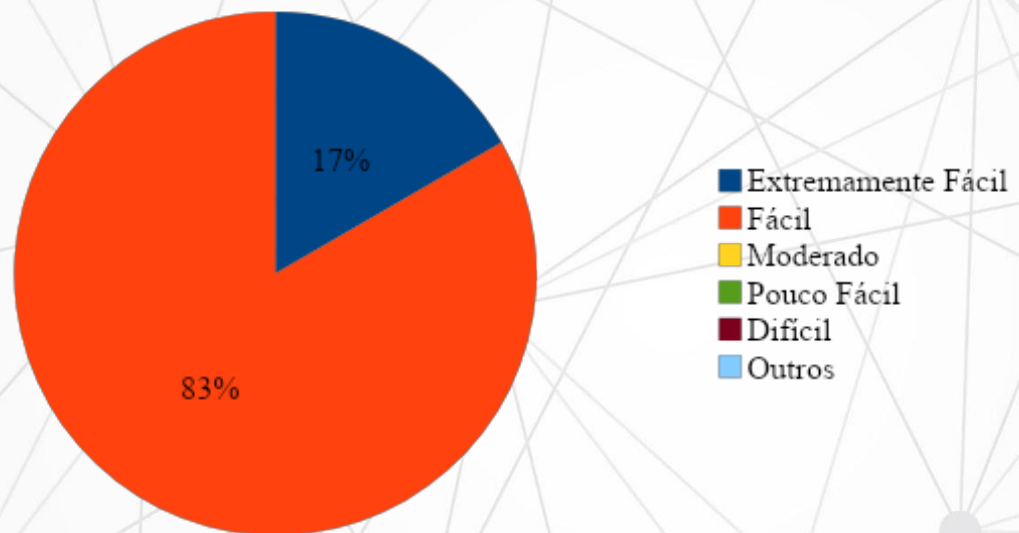
Ao término do desenvolvimento do aplicativo, foram realizados testes com as diversas pessoas que se voluntariaram. Esses testes foram realizados com o aplicativo submetido na loja Google Play Store em teste fechado, necessitando que elas tivessem um cadastro na Google, já que é parte da política da empresa.

Com essa exigência dos voluntários, ao utilizarem o aplicativo foi solicitado que respondessem um questionário ao término da utilização do aplicativo. Foram levados vários itens em consideração com 4 perguntas a serem respondidas. Essas perguntas abordam sobre a usabilidade do aplicativo, recomendação do aplicativo para terceiros, se apresentou erros

e com qual frequência e se o aplicativo ajudou o usuário na decisão de compra ou venda em alguma ação, além de pedir um feedback do usuário, como opcional. Com isso, tornou-se possível ter noção de melhorias para o app.

A primeira pergunta é sobre a usabilidade do aplicativo, que foi dividida em seis respostas em extremamente fácil, fácil, moderado, pouco fácil, difícil e outros, caso o usuário encontra-se outro nível.

Gráfico 1: Usabilidade do aplicativo

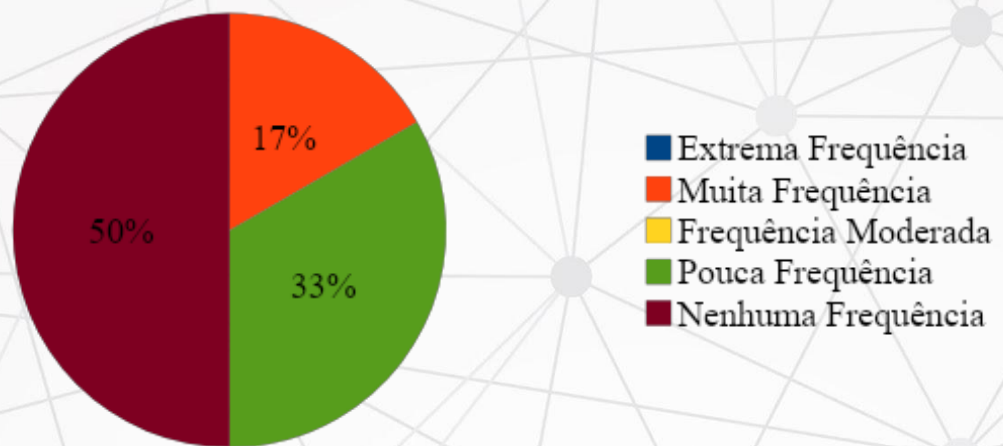


Fonte: Os autores (2023)

Como é visto no gráfico acima, a maior parte dos participantes informaram que o aplicativo é fácil de utilizar.

Outra pergunta de fundamental importância é sobre os erros apresentados. Ela é dividida em cinco respostas, sendo elas extrema frequência, muita frequência, frequência moderada, pouca frequência e nenhuma.

Gráfico 2: Erros apresentados



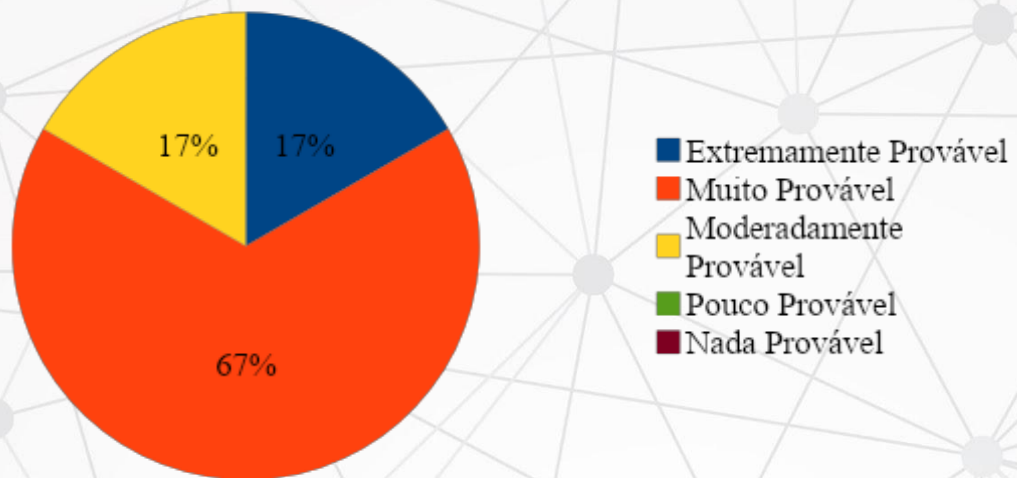
Fonte: Os autores (2023)

Nessa pergunta tivemos diversas respostas. Metade dos voluntários não encontraram problemas no aplicativo. A outra metade dos participantes informaram que encontram problemas, sendo dividido em pouca frequência, em sua grande maioria, é muito frequente.

Esses erros, foram relatados pelos usuários na parte de feedback, onde encontraram erros com mensagens pouco esclarecedoras sobre o erro. Outro erro relatado é na parte de login, onde foi informado que o usuário era inválido, mesmo tendo o cadastro realizado.

A pergunta de recomendação do aplicativo dividiu-se em cinco respostas, sendo extremamente provável, como a resposta de maior importância, muito provável, moderadamente provável, pouco provável e nada provável. As respostas dos participantes são ilustradas no gráfico abaixo.

Gráfico 3: Recomendação do aplicativo



Fonte: Os autores (2023)

Como é possível observar no gráfico acima, todos recomendam o aplicativo, boa parte dos participantes muito provavelmente recomendariam o aplicativo para amigos e/ou familiares. Apenas uma pequena parte extremamente recomendaria e outra pequena fração recomendaria moderadamente.

A última pergunta onde é pedido ao usuário se aplica o auxílio na escolha de compra e venda de uma ou mais ações listadas, sendo respondidas apenas em duas respostas, sim ou não. Todos responderam que o aplicativo ajuda nesse quesito, sendo assim de grande importância.

Com isso é possível observar que o ML Invest está seguindo o caminho certo com boa recepção pelos usuários, sendo uma porta de entrada para os usuários que queiram investir na renda variável. O aplicativo ainda possui alguns pontos a serem revistos, mas temos pontos importantes, como visto nos feedbacks de alguns usuários, gostando da usabilidade e também da aparência.

5 CONCLUSÃO

Após o desenvolvimento da aplicação e da elaboração da pesquisa, com os resultados obtidos, conclui-se que o app seria de grande utilidade para as pessoas, já que aquelas que estão iniciando no mercado de renda variável tem uma certa dificuldade para realizar as operações, dada a imprevisibilidade do mercado.

O aplicativo ML Invest, pode reduzir essa imprevisibilidade, já que ele utiliza aprendizagem de máquina para poder processar e entregar um resultado próximo ao que está sendo praticado.

Mas não é possível prever em todos os casos, já que fatores externos ao preço das ações podem influenciar o valor final da ação, como por exemplo, o aumento de carga tributária em um determinado setor, sendo esse tipo de dado não previsto nesta aplicação.

REFERÊNCIAS

CAMARGO, Bruna. Anbima: volume investido por brasileiros sobe 6,5% e alcança recorde. **Estadão**. [S. l.], 6 maio de 2022. Disponível em: <https://investidor.estadao.com.br/investimentos/volume-investimentos-brasileiros-atinge-recorde>. Acesso em: 15 ago. 2022.

HAN, J.; PEI, J.; KAMBER, M. **Data mining**: Concepts and techniques. Elsevier Science, 2011.

KERAS. **Tensorflow**. 5 jun. 2020. Disponível em: <https://www.tensorflow.org/guide/keras?hl=pt-br>. Acesso em: 29 maio 2023.

LANZA, Luíza. Renda fixa em alta, ações em baixa: os ativos preferidos dos investidores. **Estadão**. Disponível em: <https://investidor.estadao.com.br/investimentos/renda-fixa-aco-es-investimentos-preferidos-julho/>. Acesso em: 21 ago. 2022.

LENZ, Maikon Lucian; NEUMANN, Fabiano Berlinck; SANTARELLI, Rodrigo; SALVADOR, Douglas. **Fundamentos de aprendizagem de máquina**. Porto Alegre, RS: Grupo A, 2020.

O que é renda fixa? Quais são os investimentos? Nunca tem risco? Saiba mais. **UOL**. 18 ago. 2020. Disponível em: <https://economia.uol.com.br/guia-de-economia/o-que-e-renda-fixa-entenda-e-veja-como-aplicar.htm>. Acesso em: 21 ago. 2022.

O que é renda variável? Quais são os tipos, os riscos e como investir? **UOL**. 17 ago. 2020. Disponível em: <https://economia.uol.com.br/guia-de-economia/o-que-e-renda-variavel-entenda-e-veja-como-investir.htm>. Acesso em: 21 ago. 2022.

O machine learning pode ser utilizado no mercado financeiro? Vem que a gente explica. **XP Educação**. 27 jan. 2023. Disponível em: <https://blog.xpeducacao.com.br/machine-learning-no-mercado-financeiro/>. Acesso em: 18 jun. 2023.

PACHECO, Paula. Quais são os tipos de investimento e como escolher o melhor para você?. **UOL**. São Paulo, 2022. Disponível em: <https://economia.uol.com.br/mais/ultimas-noticias/2022/01/17/tipos-de-investimento-como-escolher-perfil.htm>. Acesso em: 16 ago. 2022.

RENDA fixa: tudo o que você precisa saber para começar a investir. **Infomoney**. [201-?]. Disponível em: <https://www.infomoney.com.br/guias/renda-fixa/#guia-diferenca-renda-fixa-e-renda-variavel>. Acesso em: 21 ago. 2022.

RENDA variável: guia completo para conhecer e investir. **Infomoney**. [20--]. Disponível em: <https://www.infomoney.com.br/guias/renda-variavel/>. Acesso em: 21 ago. 2022.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to data mining**. Pearson Addison Wesley, 2006.

Sobre o(s) autor(es):

1 Acadêmico do curso de graduação em Ciência da Computação da Universidade do Oeste de Santa Catarina – Unoesc, Campus São Miguel do Oeste. E-mail: schneider0095@gmail.com

2 Acadêmico do curso de graduação em Ciência da Computação da Universidade do Oeste de Santa Catarina – Unoesc, Campus São Miguel do Oeste. E-mail: andrelismoreirasmo@gmail.com