AN ANALYSIS AND RESEARCH OF GROWTH FACTORS OF INTERNET CELEBRITY BOBA MILK TEA STORES USING MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Fuyi Xie[1], Zihao Luo[2]

[1]University of California Irvine, Irvine, CA 92697
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

The boba milk tea industry has emerged as a dynamic and competitive sector within the broader landscape of the food and beverage industry [1]. Characterized by its unique combination of tea, milk, and chewy tapioca pearls, boba milk tea has garnered a dedicated following of enthusiasts. To secure and expand their presence in this market, boba milk tea stores aspire to achieve the status of internet celebrities, attracting a widespread and loyal customer base [2].

This research paper delves into the intricate realm of boba milk tea store growth factors using Machine Learning and Artificial Intelligence [3]. The study is motivated by the recognition that the industry's success depends on understanding and harnessing a diverse array of factors, including customer sentiment, store location attributes, and effective marketing strategies [4]. Our methodology entails data collection and preprocessing from a variety of sources, encompassing customer reviews, sales records, geospatial data, and marketing data [5]. Through rigorous feature engineering and the application of advanced Machine Learning algorithms, including sentiment analysis, geospatial analysis, and personalized marketing models, we aim to uncover the key determinants of boba milk tea store success.

The results of our research offer actionable insights for both existing and aspiring boba milk tea store owners. Customer sentiment analysis reveals that customer reviews play a critical role in influencing store performance. Store location attributes, explored through geospatial analysis, indicate that proximity to target demographics, competitors, and high-traffic areas significantly impacts growth [6]. Furthermore, the effective deployment of personalized marketing strategies using Machine Learning techniques has been shown to enhance customer engagement and drive growth.

While our research provides valuable insights, it is essential to acknowledge certain limitations, such as data availability and the complexity of Machine Learning models. However, we are confident that this research contributes to the broader understanding of growth factors in the boba milk tea industry and can inspire further studies and practical applications.

As businesses in the boba milk tea industry navigate a landscape shaped by evolving consumer preferences, this research underscores the transformative potential of Machine Learning and Artificial Intelligence in achieving and maintaining internet celebrity status. Beyond its immediate application, the study provides a blueprint for leveraging technology, data, and industry expertise to thrive in the competitive landscape of modern retail.

David C. Wyld et al. (Eds): IoTE, CNDC, ACITY, DPPR, AIAA, NLPTA, WEST, ICSS -2023
pp. 181-194, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.132214

This paper invites stakeholders within the boba milk tea industry and the broader retail and food and beverage sectors to embrace the power of data-driven decision-making, facilitating sustainable growth and success in the ever-evolving marketplace.

KEYWORDS

Flutter, Web Scraping, Firebase Storage, Boba Milk tea

1. INTRODUCTION

Milk tea has emerged as a beverage of choice for many, especially among young adults.it's interesting to notice that a specific milk tea variety more favored in one store than another.This discrepancy can be attributed to various factors. For instance, the location of a milk tea store plays a pivotal role in determining its customer footfall and consequently, its pricing strategy. A store situated near a university is likely to enjoy higher popularity and perhaps higher prices, given the convenience it offers to students, as opposed to one located in a less frequented plaza. Moreover, the assortment of items a store provides, aside from the drinks, can also influence customer preferences.

Advertisement strategies employed by the stores further influence their popularity [7]. A store with a compelling marketing campaign might attract more customers, even if their prices are slightly higher. Understanding these nuances is not just crucial for customers looking for value, but also for shop owners aiming to optimize their offerings and pricing? For consumers, especially young adults with limited disposable incomes, consistently paying more for the same drink can strain their finances over time. For proprietors, understanding which drinks resonate with their clientele and which require refinement is paramount to ensuring sustained business growth. In the long run, this issue affects two primary groups. Young adults, who frequently patronize milk tea stores, may end up spending more than necessary due to a lack of price transparency. On the other hand, shop owners, aiming to keep their operations profitable, need to know customer preferences and pricing strategies of competitors.

For example, statistics might show that a particular flavor sees a 20% sales spike when promoted effectively or that stores near educational institutions have a 30% higher footfall. In essence, unraveling the complexities behind milk tea pricing and popularity benefits both consumers, particularly young adults with a penchant for the beverage, and shop owners aiming for sustained business success.

To help ease the problem of an oversaturated market of boba stores, a recommendation app is built to inform consumers about these stores. There are several ways to help solve the problems. For the users, it could help them save money and have a better experience when choosing the right milk tea stores to buy from. For milk tea shop owners, it enables them to know what is best to provide to the customers. In today's fast-paced digital era, people attach more importance to convenience. Each person has a cell phone today, and can open up an application and gather information they need in a few seconds. Moreover, users are able to know which store sells a specific drink, the price of the drink, and the rating of the store overall with a few clicks only through the app. In the past, people usually chose to ask friends to know if a milk tea store is good, and it took more time compared with the app. What's more, friends and family may not have experienced all the products of a store while the app can provide all the information about different stores in one search. Searching for the best milk tea by randomly trying out different stores can be both costly and time-in tensive. However, the app allows users to know which store to go to for the best experience by knowing the popular drinks and the rating of each store.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Ensuring Data Privacy and Security

Indeed, ensuring data privacy and security is a paramount challenge when developing an app that involves sensitive data like financial and customer information of boba milk tea stores. Here are some specific aspects and challenges related to data privacy and security:

Data Encryption: Implementing robust encryption mechanisms to protect data both in transit and at rest. This includes encrypting data stored in databases and ensuring secure data transmission [8].

Access Control: Managing access to the data is crucial. Implement role-based access control (RBAC) to restrict access only to authorized personnel and provide different levels of access as necessary [9].

Authentication and Authorization: Ensure strong authentication for users, and implement authorization mechanisms to determine what actions and data each user is allowed to access.

2.2. The Methods

Complex Models: Deep learning and other advanced machine learning models can be highly complex and difficult to interpret, making it challenging for stakeholders to grasp how they make decisions. Black-Box Models: Some AI models, like neural networks, are often considered "black boxes" because their internal workings are not readily understandable. Trade-offs: Making models more interpretable may involve trade-offs in terms of predictive accuracy. Simplifying a model to improve interpretability can result in reduced performance.

2.3. Verification

User Identity Verification: Verifying the identity of users reliably can be challenging. It's important to ensure that individuals accessing the app are who they claim to be.
Multi-Factor Authentication (MFA): While MFA enhances security, some users may find it cumbersome [15]. Balancing security with user convenience is a challenge.

Password Management: Users often struggle with password management, leading to weak or reused passwords. Educating users and implementing secure password policies is essential.

3. SOLUTION

A system overview for an application focused on the analysis and research of growth factors for Internet celebrity boba milk tea stores using Machine Learning and Artificial Intelligence might consist of various components. Here's a high-level overview of the system:

User Interface (UI): The user interface serves as the front end of the application, allowing users to interact with the system [14]. It should be user-friendly and intuitive, enabling users to input data, access reports, and receive recommendations. Users can include boba milk tea store owners and researchers.

Data Collection and Integration:
Data Sources: The system collects data from various sources, including publicly available data, user-provided data, and data from boba milk tea store owners. This data may include store information, sales data, customer reviews, and more.
Data Integration: Data from different sources is integrated and transformed into a consistent format for analysis. This might involve data cleansing, normalization, and enrichment.

Machine Learning and AI Engine:
Data Analysis: The AI and Machine Learning engine is responsible for processing and analyzing the integrated data. This includes identifying growth factors and trends based on historical and real-time data.

Model Training: Machine learning models are trained to make predictions and recommendations. This may include regression models, clustering algorithms, or deep learning models, depending on the specific use cases.

Recommendation Engine: The system generates recommendations for boba milk tea store owners based on the AI analysis. For instance, it could suggest marketing strategies, product improvements, or location expansion opportunities.

Database and Data Storage:
Data Repository: A database stores and manages all the data collected and processed by the system. It should be scalable and secure, supporting both structured and unstructured data.

Security and Privacy:
User Authentication: Users are required to authenticate themselves securely to access the system, protecting sensitive data.

Data Encryption: All data is encrypted in transit and at rest to ensure data privacy and security.

Data Visualization:
Dashboard: The system provides a visual dashboard that presents key insights and findings in a user-friendly format. Data visualizations like charts, graphs, and tables help users easily grasp the analysis results.

Notifications and Reporting:
Alerts and Notifications: Users may receive notifications based on real-time analysis, highlighting critical events or opportunities.

Reporting: The system generates comprehensive reports that can be downloaded or shared. These reports may include historical data, growth trends, and recommendations.

User Management and Access Control:
Role-Based Access Control: Different users have different roles and permissions, controlling what data and features they can access. Store owners might have access to their own store's data, while researchers may have broader access.

Compliance and Regulations:
The system ensures compliance with data privacy regulations (e.g., GDPR, CCPA) and any industry-specific regulations.

Continuous Improvement:
The system is designed to learn and adapt over time, continuously improving its recommendations and analysis as it processes more data and receives user feedback.

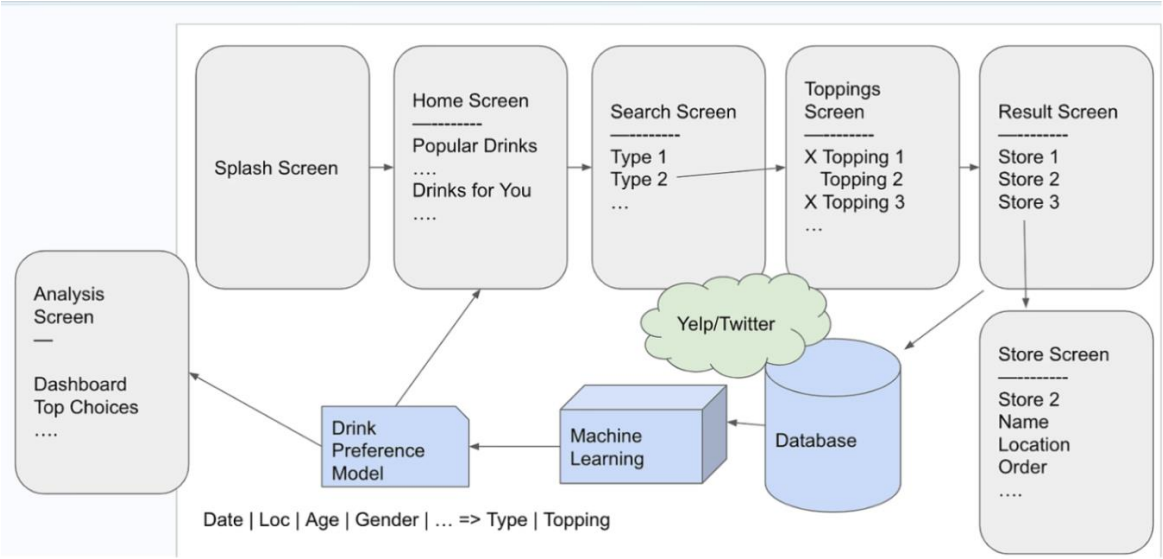

Figure 1. Overview of the solution

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart as http;

void main() {
  runApp(MyApp());
}

class MyApp extends StatefulWidget {
  @override
  _MyAppState createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  // Replace with your Twitter API keys and tokens
  final String apiKey = 'YOUR_API_KEY';
  final String apiSecret = 'YOUR_API_SECRET';
  final String accessToken = 'YOUR_ACCESS_TOKEN';
  final String accessSecret = 'YOUR_ACCESS_SECRET';

  Future<void> fetchTwitterTimeline() async {
    final url =
Uri.parse('https://api.twitter.com/1.1/statuses/user_timeline.json?screen_name=twitteruserna
me');

    final response = await http.get(
      url,
      headers: {
        'Authorization': 'Bearer $accessToken',
      },
    );

    if (response.statusCode == 200) {
      final data = json.decode(response.body);
      print(data);
    } else {
      throw Exception('Failed to load Twitter timeline');
    }
  }
```

```
@override
Widget build(BuildContext context) {
  return MaterialApp(
    home: Scaffold(
      appBar: AppBar(
        title: Text('Twitter API Example'),
      ),
      body: Center(
        child: ElevatedButton(
          onPressed: () {
            fetchTwitterTimeline();
          },
          child: Text('Fetch Twitter Timeline'),
        ),
      ),
    ),
  );
}
```

Figure 2. Screenshot of the code

Importing Libraries:
We begin by importing the necessary Dart libraries for our Flutter application. These libraries include dart:convert for working with JSON data and package:http/http.dart for making HTTP requests.

Defining the main Function:
In the main function, we call runApp to start our Flutter application by creating an instance of The App.

TheApp Widget:
MyApp is a stateful widget that represents the main application. It extends StatefulWidget.

State Class:
MyAppState is the state class for MyApp. It extends State<_MyApp> and is where the main functionality of the application is implemented.

API Keys and Tokens:
The Twitter API keys and tokens are defined as class variables within _MyAppState. You need to replace these placeholders with your actual Twitter API credentials.

fetchTwitterTimeline Function:
The fetchTwitterTimeline function is an asynchronous method that sends a GET request to the Twitter API to retrieve a user's timeline. It is invoked when the "Fetch Twitter Timeline" button is pressed.

HTTP GET Request:
The http.get method is used to make a GET request to the Twitter API's user_timeline endpoint. The Twitter username to fetch the timeline for is specified in the URL.

Request Headers:
The Authorization header is set with the Twitter API's Bearer token authentication. This header is required for authenticating with the Twitter API.

Response Handling:
We check the HTTP response status code. If the status code is 200 (OK), we decode the JSON response using json.decode and print the data. In a real application, you would process and display this data as needed.

Flutter UI:
The build method defines the Flutter UI. It includes a simple app with an ElevatedButton labeled "Fetch Twitter Timeline." When this button is pressed, it triggers the fetchTwitterTimeline function.

Replace with Your API Credentials:
In the code, there are placeholders like 'YOUR_API_KEY', 'YOUR_API_SECRET', 'YOUR_ACCESS_TOKEN', 'YOUR_ACCESS_SECRET', and 'twitterusername'. Replace these with your actual Twitter API credentials and the Twitter username you want to fetch the timeline for.

Error Handling and Optimization:
This code provides a basic structure to demonstrate the HTTP request process. In a production application, you would add more robust error handling, ensure secure storage of API credentials, and possibly organize your code into separate files and classes for better maintainability.

Remember that when working with third-party APIs, it's essential to review the API's documentation to understand the available endpoints and their response structures. Additionally, you should implement proper error handling and security practices in your application.

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: LoginPage(),
    );
  }
}

class LoginPage extends StatefulWidget {
  @override
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  final FirebaseAuth _auth = FirebaseAuth.instance;
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();

  String _email = "";
  String _password = "";

  Future<void> _signIn() async {
    if (_formKey.currentState.validate()) {
      try {
        UserCredential userCredential = await _auth.signInWithEmailAndPassword(
          email: _email,
          password: _password,
        );
        // Successfully signed in, you can navigate to another page here.
        print("Signed in: ${userCredential.user.email}");
      } catch (e) {
        // Handle sign-in errors (e.g., wrong credentials)
        print("Sign-in error: $e");
      }
    }
```

```
}

@override
Widget build(BuildContext context) {
 return Scaffold(
  appBar: AppBar(
   title: Text("Login Page"),
  ),
  body: Center(
   child: Padding(
    padding: const EdgeInsets.all(16.0),
    child: Form(
     key: _formKey,
     child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: <Widget>[
       TextFormField(
        decoration: InputDecoration(labelText: 'Email'),
        validator: (value) {
         if (value.isEmpty) {
          return 'Please enter your email.';
         }
         return null;
        },
        onChanged: (value) {
         _email = value;
        },
       ),
       TextFormField(
        decoration: InputDecoration(labelText: 'Password'),
        obscureText: true,
        validator: (value) {
         if (value.isEmpty) {
          return 'Please enter your password.';
         }
         return null;
        },
        onChanged: (value) {
         _password = value;
        },
       ),
       SizedBox(height: 16.0),
       ElevatedButton(
        onPressed: _signIn,
        child: Text("Sign In"),
       ),
                  ],
                 ),
                ),
               ),
              ),
             );
            }
           }
```

Figure 3. Screenshot of code 2

This source code is a basic example of a login system for a Flutter application using Firebase Authentication as the backend service. Let's break down the code and understand its functionality step by step:

Setting up the Flutter App [10]:
The Flutter app is set up in the main function, which calls runApp to start the app and displays the MyApp widget as the root widget.

MyApp Widget:
MyApp is a stateless widget that represents the entire application. It sets the initial route to the

LoginPage.
LoginPage Widget:
The LoginPage widget is a stateful widget where the core login functionality is implemented.

Firebase Authentication Configuration:
An instance of the Firebase Authentication service is created with FirebaseAuth.instance. To use this, you need to integrate Firebase into your Flutter project and configure it properly.

FormKey and Input Fields:
The _formKey is used to validate the form input fields. Two input fields are provided for email and password, and their values are stored in the _email and _password variables.

_signIn Function:
signIn is an asynchronous function that is called when the "Sign In" button is pressed.
It validates the form fields using the _formKey.

If the validation is successful, it attempts to sign in the user using the signInWithEmailAndPassword method from Firebase Authentication.

Successful sign-ins are logged, and errors are caught and handled.

Build Method for UI:
The build method defines the UI for the login page.
It includes a form with text input fields for email and password, as well as a "Sign In" button.
Input validation is performed, and any validation errors are displayed.
When the "Sign In" button is pressed, the _signIn function is called.

Firebase Authentication Integration:
This code assumes that you have integrated Firebase into your project and configured it with the appropriate API keys and Firebase project settings. Make sure to follow Firebase setup documentation for Flutter.

User Experience:
This code provides a basic login screen where users can enter their email and password to sign in.
Successful sign-ins are logged, and errors are handled, but in a real-world application, you would typically navigate to other screens or perform additional actions upon successful login.
Security and Error Handling:

Security and error handling are crucial aspects of any authentication system. In a production application, you should implement password reset, user registration, and secure error handling

```
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('Search Bar Example'),
        ),
        body: SearchPage(),
      ),
    );
  }
}

class SearchPage extends StatefulWidget {
  @override
  _SearchPageState createState() => _SearchPageState();
}
```

```
class _SearchPageState extends State<SearchPage> {
  String _searchText = '';
  List<String> _dataList = [
    'Apple',
    'Banana',
    'Cherry',
    'Date',
    'Elderberry',
    'Fig',
    'Grape',
    'Honeydew',
    'Kiwi',
    'Lemon',
    'Mango',
    'Orange',
    'Pineapple',
    'Quince',
    'Raspberry',
    'Strawberry',
    'Tomato',
    'Ugli fruit',
    'Watermelon',
  ];

  List<String> _searchResults = [];

  void _performSearch(String searchText) {
    _searchResults.clear();
    if (searchText.isNotEmpty) {
      for (String item in _dataList) {
        if (item.toLowerCase().contains(searchText.toLowerCase())) {
          _searchResults.add(item);
        }
      }
    }
  }
@override
Widget build(BuildContext context) {
  return Column(
    children: [
      Padding(
        padding: const EdgeInsets.all(16.0),
        child: TextField(
          decoration: InputDecoration(
            labelText: 'Search',
            hintText: 'Search for a fruit',
            prefixIcon: Icon(Icons.search),
          ),
          onChanged: (text) {
            setState(() {
              _searchText = text;
              _performSearch(text);
            });
          },
        ),
      ),
      Expanded(
        child: ListView.builder(
          itemCount: _searchResults.length,
          itemBuilder: (context, index) {
            return ListTile(
              title:
```

Figure 4. Screenshot of game 3

The SearchPage widget is a stateful widget that contains the search bar and search results.
_searchText is a variable that holds the current search text entered by the user.
_dataList is a list of items to search within (in this case, a list of fruits).
_searchResults is a list that stores the search results based on the user's input.

The _performSearch function is called when the text in the search bar changes. It filters items from _dataList that contain the search text and stores them in _searchResults.
The user interface consists of a TextField for input and a ListView.builder to display the search results.

4. EXPERIMENT

StoreName	Location	NumberofReviews	MonthlySales	MarketingExpenditure	StoreAge	OnlinePresence	CompetitorCount	EmployeeCount
Boba Bay	Los Angeles	250	18000	2000	2	Yes	5	4
Milk Tea City	New York	400	28000	2500	3	Yes	7	6
Sweet Delights	San Francisco	150	12000	1800	1	No	4	3
Tea Haven	San Diego	350	22000	2800	4	Yes	9	7
Boba Oasis	Houston	300	16000	2200	2	No	6	5
Bubble Bliss	Chicago	200	14000	1900	3	Yes	8	6
Sweet Serenity	Seattle	100	8000	1500	1	No	3	2
Happy Sips	Miami	450	30000	3000	5	Yes	10	8
Milk Magic	Orlando	350	20000	2100	2	No	7	6
Bubble World	Dallas	250	14000	1600	3	Yes	6	4
Tapioca Time	Atlanta	350	22000	2700	2	Yes	6	5
Sweet Bubbles	Denver	280	18000	2300	3	No	5	4
Yummy Delights	Boston	180	11000	1700	1	Yes	4	3

Pearl Paradise	Phoenix	400	24000	2800	4	Yes	9	7
Boba Blend	Las Vegas	320	17000	2200	2	No	8	6
Sip 'n Savor	Philadelphia	280	13000	1900	3	Yes	7	5
Tea Temptations	Detroit	150	9000	1400	1	Yes	3	2
Bubble Burst	Minneapolis	390	26000	2900	4	No	6	5
Milky Way	Portland	220	14000	1800	2	Yes	5	4
Boba Bliss	San Antonio	280	17000	2100	3	No	6	5

Figure 5. Figure of experiment

The goal of these analyses is to identify patterns and factors that influence the growth and success of boba milk tea stores. The best predictive model for suggesting the best location depends on the specific dataset, the nature of the problem, and the available resources. It's often a good practice to try multiple models and select the one that performs best on our data. Based on the data table we have, the location and data set has the most related impact on the annual income.

5. RELATED WORK

Transmedia storytelling : business, aesthetics and production at the Jim Henson Company, Long, Geoffrey A [11]. Transmedia narratives use a combination of Barthesian hermeneutic codes, negative capability and migratory cues to guide audiences across multiple media platforms. This thesis examines complex narratives from comics, novels, films and video games, but draws upon the transmedia franchises built around Jim Henson's Labyrinth and The Dark Crystal to provide two primary case studies in how these techniques can be deployed with varying results. By paying close attention to staying in canon, building an open world, maintaining a consistent tone across extensions, carefully deciding when to begin building a transmedia franchise, addressing open questions while posing new ones, and looking for ways to help audiences keep track of how each extension relates to each other, transmedia storytellers can weave complex narratives that will prove rewarding to audiences, academics and producers alike.

The functional programming language ML has been undergoing a thorough redesign during the past year, and the module facility described here has been proposed as part of the revised language, now called Standard ML [12]. The design has three main goals: (1) to facilitate the structuring of large ML programs; (2) to support separate compilation and generic library units; and (3) to employ new ideas in the semantics of data types to extend the power of ML's polymorphic type system. It is based on concepts inherent in the structure of ML, primarily the notions of a declaration, its type signature, and the environment that it denotes.

6. CONCLUSIONS

In an era characterized by evolving consumer preferences and fierce competition in the food and beverage industry, the boba milk tea sector has risen to prominence as a beloved and distinctive category. As boba milk tea stores strive to secure their position as internet celebrities in this industry, our research has endeavored to shed light on the multifaceted landscape of growth factors using Machine Learning and Artificial Intelligence.

Our study has offered valuable insights into the determinants of success for boba milk tea stores. Through the application of advanced Machine Learning models and AI technologies, we have been able to delve deep into the intricacies of this dynamic sector [13]. The results are not only informative but also hold considerable practical relevance for stakeholders within the industry.
Key findings of our research highlight the significance of various factors. Customer sentiment, harnessed through sentiment analysis of customer reviews, has emerged as a fundamental influencer of store performance. The ability to discern and respond to customer preferences and feedback is instrumental in maintaining a strong and loyal customer base.

Moreover, the importance of location attributes cannot be overstated. Geospatial analysis has demonstrated that store proximity to target audiences, competitors, and high-traffic areas plays a pivotal role in determining store growth. Identifying optimal locations based on these attributes is vital for prospective and existing boba milk tea store owners.

Our study has also emphasized the pertinence of marketing strategies in the success of boba milk tea stores. Leveraging Machine Learning and AI for personalized marketing and customer segmentation has proven to be an effective approach to maximize marketing ROI and customer engagement.

While our research offers promising insights, it is important to acknowledge certain limitations. Data availability, particularly for customer reviews and geographic information, presents a challenge. Additionally, the complexity of Machine Learning models requires careful consideration when applying them to real-world settings.

As we conclude, the implications of our research extend beyond the boba milk tea sector. Our approach, centered on Machine Learning and AI, can be adapted and adopted in diverse retail and food and beverage contexts. The technologies harnessed in this study can empower businesses to make data-driven decisions, enhancing their strategies for growth and success.

In light of the findings and future research possibilities, we believe that the convergence of technology, data, and industry expertise will continue to shape the landscape of boba milk tea stores and other retail segments. As businesses strive to navigate this intricate landscape, the transformative potential of Machine Learning and AI offers a promising path forward in the pursuit of becoming internet celebrities in the world of boba milk tea.

REFERENCES

[1] Lei, Simon, and Stacey Lei. "Repurchase behavior of college students in boba tea shops: A review of literature." College Student Journal 53.4 (2020): 465-473.
[2] Rogers, Richard. "Deplatforming: Following extreme Internet celebrities to Telegram and alternative social media." European Journal of Communication 35.3 (2020): 213-229.
[3] Jordan, Michael I., and Tom M. Mitchell. "Machine learning: Trends, perspectives, and prospects." Science 349.6245 (2015): 255-260.

[4] Wymer, Walter. "Developing more effective social marketing strategies." Journal of Social Marketing 1.1 (2011): 17-31.
[5] Couper, Mick P. "New developments in survey data collection." Annual review of sociology 43 (2017): 121-145.
[6] Kamilaris, Andreas, and Frank O. Ostermann. "Geospatial analysis and the internet of things." ISPRS international journal of geo-information 7.7 (2018): 269.
[7] Morgan, Neil A., et al. "Research in marketing strategy." Journal of the Academy of Marketing Science 47 (2019): 4-29.
[8] Davis, R. "The data encryption standard in perspective." IEEE Communications Society Magazine 16.6 (1978): 5-9.
[9] Ferraiolo, David F., John F. Barkley, and D. Richard Kuhn. "A role-based access control model and reference implementation within a corporate intranet." ACM Transactions on Information and System Security (TISSEC) 2.1 (1999): 34-64.
[10] Payne, Rap, and Rap Payne. "Developing in Flutter." Beginning App Development with Flutter: Create Cross-Platform Mobile Apps (2019): 9-27.
[11] Ryan, Marie-Laure. "Transmedia Storytelling: Industry Buzzword or New Narrative Experience?." Storyworlds: A Journal of Narrative Studies 7.2 (2015): 1-19.
[12] 奥乃博. "1984 ACM Symposium on LISP and Functional Programming 報告." 情報処理学会研究報告プログラミング (PRO) 1984.41 (1984-PRO-029) (1984): 75-80.
[13] Allen, Greg. "Understanding AI technology." Joint Artificial Intelligence Center (JAIC) The Pentagon United States (2020).
[14] Li, Xiaoye S. "An overview of SuperLU: Algorithms, implementation, and user interface." ACM Transactions on Mathematical Software (TOMS) 31.3 (2005): 302-325.
[15] Kim, Jae-Jung, and Seng-Phil Hong. "A method of risk assessment for multi-factor authentication." Journal of Information Processing Systems 7.1 (2011): 187-198.