Ministry of Science and Higher Education of the Republic of Kazakhstan

Suleyman Demirel University

Anefiyayev Nurdaulet

Application for predicting the business orientation based on analysis of user desires

THESIS

Presented in Partial Fulfilment for the

Master of Technical Sciences Degree in Computer Science

(degree code: 7M06102)

Department of Computer Science

Faculty of Engineering and Natural Sciences

Supervisor: **Assistant Professor, PhD Assem Talasbek**

Kaskelen 2023

Suleyman Demirel University
Faculty of Engineering and Natural Sciences
Department of Computer Science

Dean of Faculty

Associate Professor

PhD Zhamanov A.

06 06 2023

Topic of the thesis:

Application for predicting the business orientation based on analysis of user desires

Thesis submitted as part of the requirements for the award of the MSc in "7M06102 - Computer Science" SDU, 2021-2023

Head of Department Assistant Professor, PhD Mukash Zh.

Academic Supervisor Assistant Professor, PhD Assem T.

Master student Nurdaulet Anefiyayev

Kaskelen 2023

Abstract

In today's competitive business landscape, understanding customer desires and preferences is crucial for the success of any organization. Customer churn is a significant problem for companies and describes moving out of customers to competitors. Predicting such behavior in advance offers companies valuable insights, empowering them to take measures to retain their customer base and potentially expand it. One key decision informed by data analysis involves identifying the most promising areas for business development. Machine learning algorithms can be leveraged to analyze customer data and predict business direction. This helps organizations make data-driven decisions and tailor their products, services, and marketing strategies to meet customer expectations. The thesis work describes how customer preferences were studied using their feedback, and analyzed partners and transactions to predict income for the company. To solve the problem, several stages and machine learning algorithms from different areas were used to compare and select for further use. Neural networks showed the best result in determining the direction of business and linear regression to find out a profit. After that, a website and a mobile application were built where the current situation of the company was visualized, and a page where the saved model was used to predict a new partner and his income.

Keywords: Business direction, customer feedback, churn rate, behavior prediction, machine learning techniques, visualization tool

Аңдатпа

Бүгінгі бәсекеге қабілетті бизнес саласында тұтынушылардың тілектері мен қалауларын түсіну кез келген ұйымның табысты болуы үшін өте маңызды. Клиенттердің іркілуі компаниялар үшін маңызды мәселе болып табылады және тұтынушылардың бәсекелестерге ауысуын сипаттайды. Мұндай мінез-құлықты алдын ала болжау компанияларға құнды түсініктер береді, олардың тұтынушылар базасын сақтап қалу және оны кеңейту үшін шаралар қабылдауға мүмкіндік береді. Деректерді талдау нәтижесінде алынған негізгі шешімдердің бірі бизнесті дамытудың ең перспективалы бағыттарын анықтауды қамтиды. Машиналық оқыту алгоритмдерін тұтынушы деректерін талдау және бизнес бағытын болжау үшін пайдалануға болады. Бұл ұйымдарға деректерге негізделген шешімдер қабылдауға және тұтынушылардың үміттерін қанағаттандыру үшін өнімдерін, қызметтерін және маркетингтік стратегияларын бейімдеуге көмектеседі. Диссертациялық жұмыста тұтынушылардың қалауы олардың кері байланысы арқылы қалай зерттелгенін сипаттайды және компанияның кірісін болжау үшін серіктестер мен транзакцияларды талдайды. Мәселені шешу үшін салыстыру және одан әрі пайдалану үшін таңдау үшін әртүрлі облыстардың бірнеше кезеңдері мен машиналық оқыту алгоритмдері пайдаланылды. Нейрондық желілер бизнестің бағытын анықтауда және пайда табу үшін сызықтық регрессия ең жақсы нәтиже көрсетті. Осыдан кейін компанияның ағымдағы жағдайы визуалды түрде көрсетілетін веб-сайт пен мобильді қосымша және жаңа серіктес пен оның кірісін болжау үшін сақталған модель пайдаланылған бет жасалды.

Түйін сөздер:Бизнес бағыты, тұтынушылардың кері байланысы, мінез - құлықты болжау, машинаны оқыту әдістері, визуализация құралы

Аннотация

В сегодняшней конкурентной среде бизнеса понимание желаний и предпочтений клиентов имеет решающее значение для успеха любой организации. Отток клиентов — серьезная проблема для компаний, связанная с уходом клиентов к конкурентам. Заблаговременное прогнозирование такого поведения дает компаниям ценную информацию, позволяя им принимать меры для сохранения своей клиентской базы и потенциального ее расширения. Одним из ключевых решений, основанных на анализе данных, является определение наиболее перспективных областей для развития бизнеса. Алгоритмы машинного обучения можно использовать для анализа данных о клиентах и прогнозирования направления бизнеса. Это помогает организациям принимать решения на основе данных и адаптировать свои продукты, услуги и маркетинговые стратегии в соответствии с ожиданиями клиентов. В дипломной работе описывается, как изучались предпочтения клиентов с использованием их отзывов, а также анализировались партнеры и сделки для прогнозирования доходов компании. Для решения задачи было использовано несколько этапов и алгоритмов машинного обучения из разных областей для сравнения и выбора для дальнейшего использования. Нейронные сети показали лучший результат в определении направления бизнеса и линейной регрессии для выяснения прибыли. После этого были построены сайт и мобильное приложение, где визуализировалось текущее положение компании, а также страница, где по сохраненной модели прогнозировался новый партнер и его доход.

Ключевые слова: Направление бизнеса, отзывы клиентов, коэффициент оттока, прогнозирование поведения, машинное обучения, инструмент визуализации.

Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
ARIMA	AutoRegressive Integrated Moving Average
AUC	Area under the ROC Curve
CI/CD	Continuous Integration and Continuous Delivery
CRF	Conditional Random Fields
CSV	Comma-separated values
GARCH	Generalized AutoRegressive Conditional Heteroskedasticity
GRU	Gated Recurrent Units
HTML	HyperText Markup Language
JS	JavaScript
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
KNN	K-Nearest Neighbors algorithm
LASSO	Least Absolute Shrinkage and Selection Operator
LLP	Limited Liability Partnership
LR	Linear Regression
LSTM	Long Short-Term Memor
ML	Machine Learning

MLP	Multilayer Perceptron
NASDAQ	National Association of Securities Dealers Automated Quotations
NN	Neural Network
ROC	Receiver Operating Characteristic curve
QR	Quick Response code
REST	REpresentational State Transfer
RNN	Recurrent neural network
SMOTE	Synthetic Minority Oversampling TEchnique
SVM	Support Vector Machines
TF-IDF	Term Frequency - Inverse Document Frequency

Table of Contents

Chapter 1

Introduction

1.1 Motivation

People have always sought advice and help while challenged with a choice, whether it's about choosing a restaurant, book, or movie. In the past, they relied on traditional methods such as asking friends. However, obtaining recommendations has become much easier with the improvement of websites and social media. This has led to the emergence of companies that offer services based on customer flow. For businesses like these, forecasting customer turnover, or the speed at which consumers depart from the company, is among the most vital responsibilities. It's crucial for companies to retain their customers because it's costlier to attract new ones than to keep the existing clientele [1]. Hence, companies are on the hunt for superior technology that can help pinpoint potential customer departures. The key to maintaining and growing the user base lies in the analysis of customer behavior and understanding their needs. Analyzing customer actions and considering their preferences is essential for retaining and increasing the number of users. Failing to do so would lead to increased operational expenses and ineffectual marketing initiatives. A firm that offers services like easy payment methods and other handy features geared towards making life simpler for people, strategically plans each year to improve customer satisfaction and stimulate growth. To achieve this, it is important to make accurate assumptions and analyze user actions, and machine learning prediction algorithms can aid in this process. However, pre-existing

algorithms and tools in the market may not yield the best results as each company is unique, and it is crucial to rely on their own data collected over the years. By setting goals and choosing a path based on data analysis, the business may also run kick-starts and cultivate new projects in a focused direction with a single product, driven by user demand. Typically, startups face risks while entering a competitive market, but in this scenario, the parent company can provide a safety net to the new offshoot, thus lowering the risk factor.

1.2 About company

In a business where every sale generates substantial revenue, it makes sense to invest heavily in customer acquisition. But you need to choose the right target audience in order to effectively spend your marketing budget, you should know who the company's customers are and where to find them. Without relevant data, you will have to act at random. And even if data is collected, it still needs to be analyzed, and people have a poor understanding of the relationship between terabytes of statistical and demographic numbers. What is customer acquisition - marketers analyze the buyer pool, look for other people with similar profiles, and invest in targeted ads. Companies now have to use multiple channels to find and retain customers, from old-fashioned emails to social media and chatbots. Multi-component and rapidly changing data sets are a difficult task.

Rahmet allows you to make purchases using your smartphone and receive cashback, and also allows partners to digitize user traffic and generate sales and service analytics. The company was founded in 2018 and has been actively developing ever since. Profitable - users receive cashback from every phone payment. Customers save part of the funds spent on daily expenses by paying with the application. Cashback returns to the wallet almost immediately. Users can add any bank card, link kcell cellular balance, or use google pay or apple pay services. The accumulated bonuses can be spent at your favorite establishments. Convenient - all discounts, promotions and bonuses from partners are collected in one application. Users become part of the loyalty system without wasting time filling out a questionnaire. Also, only an application is needed to navigate by places, no

need to carry a map of each business company. There is no need to carry a bank card or cash with you, payment can be made using a mobile application. Safe - no need to give a bank card to the waiter or enter personal details while shopping online. Now, you can have confidence that your payment information is securely safeguarded. The transfer of payment data carried out using the SSL protocol, meets all international security standards.

To use Rahmet, you just need to install the application on your smartphone, register by phone number, link your bank card, scan a QR code at partner establishments;

Our company did not have its own team of analysts, so the managers themselves made decisions related to development, often without confirmation by numbers. I offered my help and AI solution, automation, visualization and forecasting in return for giving me data. They agreed to first test for a part of the data and do it outside of working hours, since I worked as a mobile developer in this company.

1.3 Aims and Objectives

The purpose of the thesis is to predict the best business orientation for the company and create a software website and mobile platform in order to be capable of allowing company members to visualize the income. Such an application would allow the company to respond promptly to changes in customer churn rate, ensuring the sustained interest of the customer base. Consequently, this would strengthen customer relationships and loyalty by reducing the risk of customer churn.

The initial important approach is to choose the right dataset, determine the desired parameters, discard the noisy data, and normalize the data. Next, work with preprocessed data and pass it to different models for training. First, we choose a methodology for analyzing user ratings, then we analyze partners in different categories, and finally, we calculate the profit. After that, need to save the trained model with the best results for further use. At this moment, a web and mobile platform will be designed and created.

This gives the company several benefits: managers will not collect data and analyze it every time they need; all tables and metrics are collected in one place and are available both from the website and from the mobile application; the company does not spend extra money on signing unprofitable partners and marketing wastes; managers see how much you can earn annually and choose the right partners to make more profits.

1.4 Thesis outline

The chapter Introduction provides an overview of the research topic, its context, and the significance of the study, including the aims and objectives, explaining what the study intends to achieve.

The second chapter Literature Review provides an analysis of what this thesis cover, briefly summarizing the main topics and studies that need to be reviewed. This section examined relevant 15 research works, each of them discussed highlighting its strengths and weaknesses, as well as its relevance to research. A table was constructed comparing their aims, methods, and results.

The next chapter lists components, algorithms, and tools that have been used to carry out the research question. The methodology section shows data collection procedures, its analysis and normalization process, model construction, and hyper-tuning. This stage then presents the findings from dataset and evaluation of models that have been used,

In the conclusion section, summarized the main findings of the research, discussed the significance of the results, and listed the limitations of the work. Based on results and limitations, suggested areas where further research could be valuable, future development points.

In the end, The References section lists all the sources that are cited throughout the thesis: journal articles, and online resources. The Appendices section includes additional materials including code, images, and tables that can be useful to understand more.

Chapter 2

Literature Review

Various studies have explored ways to improve customer engagement and retention in businesses. Companies often hold strategic meetings to determine which areas of their business require more focus and attention. Since every business is unique, it is important to learn from the experiences of others, taking into account both their successes and failures. Prior research by other authors has revealed different approaches to addressing this issue, which will be discussed in further detail. One approach involves developing a software tool that allows for visualization of customer behavior and prediction of potential customer churn. The authors of this study used two data sets of clients, with a total of 500 unique customers and 16 attributes. The training data spanned eighteen month period (from the first month of 2015 until the second half of next year), while the testing data covered the second half of the same year. Authors explored different perspective ML algorithms, including RNN, regression, gradient boosting, and random forest with bootstrap aggregating. The optimal model was selected based on its predictive capabilities on the test set, measured using Area Under ROC Curve (AUC) metric. The outcomes indicated that the random forest model emerged as the most proficient [2].

The researchers at Creative Design LLP, a clothing company for children, were concerned about the quality and competitiveness of their products as they had not performed a thorough analysis of their competitors due to limited resources as a small business. To address this issue, they needed a tool that could

help them make informed decisions about the products they create. They decided to create a system capable to investigate the opinions of online users concerning various aspects related to their business. The author of the study evaluated the accuracy of text classification using two metrics: recall and precision, which were measured based on four classifications: true positive, true negative, false positive, and false negative. The author trained model classifying with techniques like k-nearest neighbors, Naive-Bayes, n-gram sequence, then conducted cross-validation checks to test the results [3].

It seems that the study aims to use consumer feedback and support cases to improve customer experience by identifying different attributes that contribute to positive or negative experiences. The authors used artificial neural networks and Naive Bayes to classify the data and compared their results. The set used for this purpose was sourced from the TripAdvisor webpage and consisted of more than 200 hotels and around 30 destinations. ANN showed an accuracy of 80% as indicated in the outcome, which was a 7% improvement over the proposed model [4].

To facilitate more informed evolutionary decisions were proposed a novel approach to capture evolving user requirements, focusing on real-time analysis of human intentions. Unlike traditional methods centered on historical defects and delayed feedback, one of the methods utilizes CRF to quantitatively assess users' emerging needs. This enables accurate goal inference from real-time observations of user actions and environmental changes. Another paper offers three key contributions: applying a CRF-based method to infer user goals from their actions and environmental contexts; proposing techniques to explore users' emerging intentions based on goal analysis results; suggesting a software evolution process driven by these intention detection methods, validated via a case study on an online service system with 120 users. The CRF model demonstrated impressive accuracy (91%) and confidence (0.85) when employed with suitable feature templates. In contrast, Hidden Markov Models performed sub-optimally with less than 74% accuracy, highlighting CRF's strength in sequential labeling [5].

The forthcoming study was leverage individual home appliance data obtained through load identification and employs a Markov chain to examine the temporal

behaviors of these appliances. This informs the development of a short-term load forecasting method, which significantly enhances prediction accuracy. Performed state analysis on key home appliances such as water heaters, air conditioners, and kitchen appliances in 100 households, utilizing data collected over a three-month period. The Markov model's predictive error rate is notably lower at 7-8% in comparison with the approximate 20% error rate of analogous day algorithms. Short-term load forecast based on short-interval household electricity consumption data involves: the introduction of non-intrusive load identification for monitoring current residential electricity consumption data in China, which enables data collection on appliance types, on/off moments, power, and energy; detailed analysis of household appliance operation characteristics allows for the determination of varying parameters and relevant boundary values of diverse states, leading to a state space set for specific appliances; using the Kolmogorov forward equation, formed a short-term load prediction process, case study data from 100 households collected between July and September showcase time-dependent probability curves for air conditioners and water heaters that align with daily household behaviors. The paper acknowledges the limitation of having broadly defined appliance states, which affects prediction accuracy. Future work will focus on refining the prediction model by detailing appliance states and incorporating a multiple-order Markov model [6].

Focusing on predicting customer churn in financial institutions, using a dataset retrieved from a bank database, develops a model working on the Multilayer Perceptron (MLP) Artificial Neural Network architecture, allowing to forecast potential churners and non-churners. The model's robustness is evaluated using performance metrics including precision, accuracy, recall, and f-score, and assessed the model's effectiveness against an analytical tool. The study's primary goals are: to devise a customer churn prediction model using MLP of ANN, then use it to identify possible churners and non-churners; soon after decide and contrast model's efficiency with that of investigating the device. The collection of data, containing around 50,000 customer data with 42 attributes, is drawn from a leading Nigerian financial institution's database. The implementation process involved two stages: MLP software development in Python, followed by utiliza-

tion of Neuro Solution Infinity software. A confusion matrix was used to display the MLP classifier's performance on a test dataset with verified true values. The matrix summarizes counts of correctly and incorrectly (misclassified) classified cases, divided by class. The model achieved a ROC curve value of 0.89, signifying excellent performance. It accurately classified 38 customers as churners, 4841 as non-churners, while misclassifying 9 clients as churners and 112 as non-churners. The insights derived from this predictive model can be leveraged to make decisions in customer retention management systems [7].

Following article addresses the challenges faced by Taiwan's banking sector in the competitive domestic credit card market and the subdued international market. The authors of the next research proposed a model that allows local banks to identify potential customer churn early and alert them to issues that could result in customer attrition. This model fuses a customer relationship management database with a time component and uses temporal abstraction to portray data over a specific period identified by experts. Association rule mining is employed to investigate and identify abnormal customer behavior. Customers are categorized into three groups: retained customers, potential churn customers, and volunteer churn customers. For retained one, significance was achieved when the keys like support, confidence, and importance levels exceed the threshold in training and confirming phases. The rate of detection adheres support surpasses almost three percent, by confidence exceeds 92.4%, and by importance was above zero according to a rule pattern. Principle for possible churn clients were filtered with support 0.45%, for confidence surpassing sixty percent and importance over zero in two stages. In a group of volunteer churn customers, rule requires support higher than 0.6%, confidence above sixty percent, and importance over zero, in two stages. Temporal abstraction is employed to track changes in client desire in the process of time, and association rule mining is used to unearth potential rules. Ultimately, rules created with domain experts' advice were used to delineate potential managerial implications, providing a reference for senior executives in future taking actions and making decisions [8].

People engaged in the stock market are constantly seeking ways to predict stock trends, patterns, and values. Prior to the advent of machine learning al-

gorithms, prediction methods were limited and often lacked accuracy. Machine learning, however, has revolutionized this field, offering novel approaches for stock market analysis. Given the stock market's unpredictable nature, its analysis remains challenging even for machines, as there are numerous factors that they cannot comprehensively analyze or provide reliable data sets for [9].

This research examines Fama's efficient market hypothesis, and develops an innovative trading strategy using both linear and nonlinear models. The study employs Support Vector Machines (SVM) and logistic regression for stock trading. The validity of Fama's efficient hypothesis is tested using the Wald-Wolfowitz test, which evaluates the randomness hypothesis on a two-state data series. After conducting the run test, logistic regression is applied to all data from the research period to assess the factors influencing price movements. Post hypothesis testing, logistic regression and SVM are used to forecast price movement direction in the stock market. The runs-test is again employed to test the hypothesis. Results generated from linear and nonlinear models are compared. By applying linear and deep learning models to predict price movements, a novel securities trading method is proposed. The authors found that using technical analysis in the research could yield non-standard returns in an inefficient stock market. After examining all the stock data, it was concluded that the price action expectation indicator was the most beneficial [9].

Fundamental analysis, which gauges a company's intrinsic value by evaluating internal and external financial aspects like macroeconomic factors, market conditions, and competitors, is essential in the study. In particular, the focus is on blue-chip stocks, considered the most valuable in the stock market. The proposed system applies the Stacked Long Short-Term Memory Network model to identify trading signals. This model processes both short-term and long-term memories, making it applicable to sequence data analysis. The researchers use technical indicators and a deep neural network to predict trading signals for ten selected blue-chip stocks, analyzing each stock individually. A limitation of this study is that it should have accounted for short sales in the Tehran Stock Exchange. Moreover, the study used only one weighting method, suggesting the inclusion of other methods in future research. The study also incorporated daily

risk control index calculations, leading to daily portfolio rebalancing. This move could escalate transaction costs, potentially affecting the portfolio's rebalance [9].

Stock price data, characterized by nonlinearity, non-stationarity, high noise, and strong time variance, poses challenges to price prediction. Previous research primarily used technical analysis, employing models like the ARIMA model, the GARCH model for financial data's volatility clustering effect, and the VAR model, an ARIMA variant. The NPMM tag model utilized XGBoost to design a transaction system and automate transactions. Upon testing the NPMM labeling method, a trading system was created based on the method. Empirical analysis of 92 stocks listed on the NASDAQ (using the proposed system) shows a decrease in learning data measure as the N value increases in the NPMM labeling, but an overall enhancement in trading performance. Overall, the system performs well [9].

This study focuses on predicting customer churn rates in subscription-based business models using diverse machine learning algorithms. It deeply explores 21 features, including demographic and billing data, usage patterns, and relationships, with the intention of thoroughly examining multiple algorithms before comparing the predictive outcomes to identify the most effective one. The dataset, acquired from Kaggle, comprises 7043 data items and 21 features, with a task of classifying customers as either churn or not churn. The data reveals a total of 1869 churn customers and 5174 non-churn customers, yielding an overall churn rate of 26.54%. For each algorithm, average k-fold cross-validated accuracy, roc-auc, and KS scores are computed. The Logistic Regression model, with a prediction accuracy of 79.6%, surpasses all other models based on performance indicators. Being one of the simplest models, it implies that for similar customer churn prediction challenges in the future, Logistic Regression is likely to deliver superior results. An aspect of the study that could be improved is the use of SMOTE during the model building phase. The success of subscription businesses heavily depends on recurring revenue generated by subscriptions, making customer experiences and relationships integral to these businesses [10].

This study addresses the uncertainty in consumer intent to purchase new energy vehicles, driven by various factors. The aim is to identify principal elements

influencing potential customers' willingness to buy these vehicles, informed by customer satisfaction and personal characteristics data. This should aid in formulating effective sales strategies. Two unique feature selection methods based on penalty terms and tree models are utilized, employing three models (LR, LASSO, SVM) for the former and two (RF, LightGBM) for the latter, resulting in five distinct models. These models perform machine learning to ascertain relevant features influencing sales across different brands. Features selected through a voting process are then examined. The research applies mathematical model knowledge to identify factors possibly affecting sales of different electric vehicle brands, based on target customers' purchases during experiential campaigns. Findings from previous studies are combined to create a customer mining model for different electric vehicle brands, evaluating model efficacy and determining the likelihood of a target customer purchasing an electric vehicle. Key influences on customers' purchasing decisions include battery technical performance, comfort, annual mortgage of the target customer, and the ratio of car loans to annual household income. The model's training results are satisfactory, showing no overfitting or underfitting, and converging to a good state. The model's accuracy on training and testing was 86.4%, 83.9%, and 84.5% respectively. This study offers several advantages. First, it employs highly interpretable ML algorithms. Second, it performs varying operations for different categories during feature screening, enhancing the credibility of the data. Lastly, the use of multiple ML algorithms for prediction boosts the accuracy of results. However, there are some drawbacks. The study overlooks the connections between different choices, leading to potential errors. Moreover, the application of a backpropagation neural network has weak generalization capabilities due to the limited amount of data [11].

Another research focuses on implementing machine learning to understand mall customer behavior based on income and purchasing power. A public dataset was used to form clusters related to customer purchasing habits using K-Means and Hierarchical clustering methods. By frequently adjusting centroids based on node alterations, the K-Means algorithm yielded the highest accuracy. A confusion matrix was used to compare the predicted and actual results, offering insights into the accuracy of the model. The research aims to determine if a predictable

purchasing pathway can be identified, acknowledging the need for judicious feature extraction. Excessive features might detriment the model's accuracy and optimization. Future research will seek to identify more optimal features to enhance the model's capability in predicting customer spending behaviors [12].

The review of literature suggests that machine learning algorithms can improve different aspects of business operations. Multiple authors concur that ML algorithms have shown their capability to perform insightful predictions. Yet, to effectively incorporate machine learning techniques into business practices, attentive thought must be given to several moments, including the data's quality and the accuracy of the model. By finding out these points and capitalizing on the latest advancements in research, this work could open another potential for market growth and competitiveness. In Table 2.1 below, a list of literature reviews can be seen briefly describing their aims, methods, efficiency metrics, limitations and future work.

Table 2.1: Analyze of literature works

No	Research	Goals and objectives	Strategy / Approach	Performance	Limitation and Future Work
1	Reznichenko E.V. and Matveev M.G.	create a software tool to visualise customer behaviour and predict his exit	Neural network, Logistic regression, Random Forest, Gradient boosting	Precision, Recall, F1, AUC	Random forest showed best results, but for further works were chosen linear
2	Meldebay M.A. and Sarbasova A.K	Need a system to analyse views of members of the Internet community on various topics of discussion.	Naive-Bayes, KNN, n-gramm	Cross validation algorithm to check classifiers .	show results of all classifier's
3	Amit Ganesh Upadhye and A.C Lomte	Process consumer feedback to learn about different attributes that contributed to positive or poor customer	Neural network, Naive Bayes	Compared to sentiments of customers, neural networks shows better results	Apply another data mining process on this kind of data which may or may not provide better results
4	Li Huang , Gan Zhou, Jie Li, Shihai Yang	realise excellent mechanisms of information interaction and economic, needed between power grid and users	BPNN, Markov, Similar day algorithm	error rate of Markov model is only-7%~ 8% compared to around 20% of similar day algorithm	prediction will be improved from two sides, including more detailed states of the appliance and the

Table 2.2: (continue) Analyze of literature works

5	Haihua Xie, Jingwei Yang, Carl K. Chang, Lin Liu	Propose CRF methodology to provide quantitative exploration of a system-user interactions	CRF methods: divergent behaviour, goal transition, erroneous behaviour)	accuracy of goal inference is high (> 90%)	only able to capture users' emerging functional requirements , not non-functional
6	Li Chen Cheng	New model to detect potential customer churn and provide an early warning indicator to loss of customers.	Temporal abstraction, Association Rules	Churn rate index	effect of cross selling and customer loyalty not considered; in the future deal with
7	Boyuan Zhang	Predict churn rate in subscription business models before taking actions	Logistic regression, Gradient Boosting (SMOTE), NN	k-fold cross-validated accuracy, roc-auc, KS scores	A more frequent survey could be constructed during the customers' use of the service
8	Kamorudeen Akindele Amuda Adesesan . B	Predict churn by eliminating manual feature engineering in financial institution	SVM, Linear Regression, Genetic algorithm, Decision Tree, PNN	ROC values, accuracy, F1 score	For future work implement other ANN architectures LSTM
9	Zhichong Li	find main factors affecting of willing to purchase new energy vehicles	SVM, Logistic Regression, LASSO, Light GBM	accuracy	ignoring connection existing between different choices will lead to errors

Chapter 3

Project's components

3.1 Machine Learning and Data Science

Data Science is a set of specific disciplines from different areas responsible for analyzing data and finding optimal solutions based on it. In the past, this domain was exclusively handled by mathematical statistics. However, with the advent of machine learning and artificial intelligence, optimization and computer science have also been incorporated as methods of data analysis alongside mathematical statistics. Data Science holds significant importance as it combines tools, methodologies, and technologies to derive meaningful insights from data. Modernized organizations are overloaded with data, owing to a plenty of devices capable of automatically gathering and storing information.

This part describes which platform tools are used to create ML models and which language is used for part of machine learning [13].

3.1.1 Anaconda/Jupyter

Anaconda is the platform most famous for data science, which fits the needs of ML, integrated with several data sources and tools. It is an open-source Python and R languages distributive for ML, data science, analytics, scientific computing, etc. which simplifies management and deployment. Anaconda is a data analysis

tool that is available for free. Depending on the operating system (Windows, Linux, or MacOS), installation is feasible. The Anaconda interface is shown in Figure 3.1. To run, many scientific packs require particular versions of other packs. Data scientists frequently utilize various versions of many software and isolate these versions using distinct environments. Navigator allows you to interact with packages and environments without having to input conda commands into a terminal window. You may use Navigator to identify the packages you need, install them in your environment, execute them, and update them [14].

Figure 3.1: Anaconda Navigator interface

Anaconda Enterprise is the commercial version of Anaconda. This allows businesses to create enterprise-grade, scalable, and secure apps. However, you may install python and then optionally install programs using pip to execute Data Science operations. Anaconda is an option that installs all of the required packages all at once. Users will find it more convenient as a result of this. Both strategies accomplish the same goal. Developers are free to select any of them based on their preferences. Anaconda is widely used in the data science field

since it solves many common difficulties early on in the development process as well as later on. Anaconda, in general, makes data science and machine learning activities easier.

Conda is a package and environment management system that runs on different operating systems, like Windows, macOS, and Linux. Conda helps packages and their dependencies install, run and update promptly. Conda easily constructs, keeps, loads, and switches between environments on the platform. Initially is intended for Python language, but with updates, any language can be packed and distributed. It is supported in all Anaconda versions. Conda is a free and open source programming framework and environment manager. Conda installs, executes, and updates packages and their dependencies in a matter of seconds. On your local system, Conda makes it simple to build, save, load, and switch between environments. It was designed to package and deliver Python applications, although it may package and distribute software written in any language.

Conda is a package manager that simplifies the process of finding and installing software packages. An additional benefit of using conda is its functionality as an environment manager. If there's a need for a package that requires a different Python version, do not have to switch to another environment manager. With only a few commands, conda allows you to create a completely new environment to operate this alternate Python version. This feature doesn't interfere with the running of your standard Python version in usual environment. All versions of Anaconda and Miniconda come with the conda package and environment manager. Anaconda Enterprise, which allows local administration of enterprise packages and environments for Python, R, Node.js, Java, and other application stacks, is also a component of Conda. Conda is also available on the community channel conda-forge. You can also acquire conda from PyPI, although that method isn't as up to date [14].

JupyterLab is a flexible, integrable, and easily extensible environment that supports simultaneous work with several Jupyter notebooks, text files, datasets, terminals, and other components, and can organize documents in the workspace in a convenient manner using tabs and separators. JupyterLab supports the visualization and editing of many data formats: CSV, JSON, IPYNB, etc. It also

provides an adjustable interface with the ability to move blocks and layouts in a convenient order, with a focus on the interactivity of the calculations.

3.1.2 Python

Python is a high-level programming language. One of the advantages is that can be easy to pick up both for first-time beginners and advanced programmers with other languages. Python is a multi-paradigm, object-oriented, structural, generic, functional, and meta-programming are fully supported. Basic support for aspect-oriented programming is provided through metaprogramming. Many other techniques, including contract and logic programming, can be implemented using extensions. Language licenses are written in a such way, that everything is open and free, everyone can use it, develop and spread it among other developers for commercial use. Owner company already presents thousands of open libraries which were developed not only by themselves but also by community participants. This is a great opportunity for developers with an infinity of possibilities [15].

Python is a popular high-level, general-purpose programming language that is interpreted, not compiled, logo shown in Figure 3.2. It emphasizes a design philosophy that values readability and simplicity, featuring a syntax that allows developers to convey ideas using fewer lines of code compared to languages such as C++ or Java. Python's language constructs facilitate writing clean, understandable code, whether you're working on small or large scale projects.

Figure 3.2: Python Logo

3.1.3 Server-side training

There are several advantages to using a server-side AI training algorithm compared to running it on a local PC:

- Scalability. Server-side infrastructure can easily scale to accommodate larger workloads and data sets.
- Cost-effectiveness. Running AI training algorithms on a local PC may require significant investment in hardware.
- Continuous training. Server-side environments enable continuous AI training, which allows models to learn and improve over time without interruption.
- Save time. You don't have to sit down and make a list of pros and cons.

GitLab is an open-source web-based platform that provides a comprehensive suite of tools for software development, project management, and collaboration; can be identified by logo in Figure 3.3. GitLab's core functionality revolves around Git, a distributed version control system, allowing developers to efficiently manage their codebase, collaborate on projects, and track changes [16]. Initially, for data analysis, training and building models, have been used to a MacBook notebook with a macOS operating system, with an M1 Pro processor, with 16gb RAM.

Docker is an open-source platform designed to facilitate the deployment, scaling, and management of applications through the use of containerization technology, Figure 3.4. A container is essentially a lightweight, self-sufficient, and executable software package that bundles everything required to run a software component, encompassing the code, a runtime, libraries, environment variables, and system tools. This ensures that the software operates consistently, irrespective of the environment it's deployed in. Docker adds another layer of abstraction and automation to operating-system-level virtualization, both on Windows and Linux systems. This way, it simplifies the process of creating, deploying, and running applications by using containers, making it easier to package an application with all the parts it needs, such as libraries and other dependencies, and ship it all out as a single package. In simpler terms, Docker allows you to "containerize" your applications, bundling them along with their environment and dependencies, so they can run anywhere Docker is installed, regardless of the underlying operating system or hardware. The main advantages of Docker are Portability, Performance, Isolation, Scalability & CI/CD [17].

Figure 3.3: Gitlab Logo

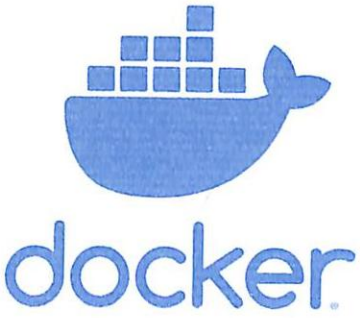

Figure 3.4: Docker Logo

Training a Machine Learning model involves multiple steps including preprocessing data, building a model, training the model, evaluating the model, and possibly iterating on the model. GitLab can support all of these steps. First of all, GitLab Repository is set up, then using code commit it to the GitLab repository. As Jupyter notebooks are used to create code, then convert them to scripts before pushing, as scripts are easier to version control. GitLab's continuous integration/continuous deployment (CI/CD) pipelines allow automation of the training and evaluation of machine learning models. A simple pipeline could include: an installation step that installs all necessary dependencies; a training step that runs training code on data; an evaluation step that evaluates the trained model; each step is run in a separate Docker container, so will need to specify an appropriate Docker image for each step. Running the Pipeline: Once the .gitlab-ci.yml file is set up, now can trigger the pipeline by pushing it to the repository. GitLab will automatically create a new pipeline and run each of the steps. Reviewing Results: The results of each step can be reviewed in the GitLab interface. If any step fails, can see the console output to help debug the issue.

3.1.4 Algorithms

3.1.4.1 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are a class of artificial neural networks designed to process sequential data by maintaining an internal state that captures information from previous time steps. They are particularly suited for tasks involving time series, natural language processing, and speech recognition, where

the order and context of the input data are crucial [18].

The primary purpose of an RNN is to model dependencies and patterns within sequences, allowing the network to make accurate predictions or generate new sequences based on the learned context. Unlike feedforward neural networks, which process inputs independently, RNNs contain loops that allow information to persist, enabling them to capture temporal dependencies.

An RNN's structure consists of input, hidden, and output layers. The hidden layer connects to itself, forming a loop that allows the network to store information from previous time steps. At each time step, the RNN takes the current input and the hidden state from the previous time step as inputs, updating the hidden state and generating an output. The hidden state acts as a memory, encoding information from the past [Appendix A.2].

However, RNNs suffer from the vanishing and exploding gradient problems, which make it difficult to learn long-range dependencies. To address these issues, advanced RNN architectures like Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) have been introduced. These architectures incorporate specialized gating mechanisms that help the network retain and access information over longer sequences, leading to improved performance in various sequence-to-sequence tasks [19].

3.1.4.2 XGBoost

XGBoost, short for eXtreme Gradient Boosting, is an open-source, high-performance machine learning library designed for efficiency, flexibility, and portability. Its primary purpose is to provide a scalable and accurate implementation of gradient-boosted decision trees, a popular ensemble learning method. XGBoost has gained widespread popularity for its exceptional performance in various machine learning competitions and practical applications, such as regression, classification, and ranking problems [20].

XGBoost works by iteratively combining weak learners, typically decision trees, into a strong learner using a gradient boosting framework. At each iteration, a new tree is added to the ensemble to minimize a given loss function. The model

learns from the residuals or errors of the previous trees, gradually improving its prediction accuracy. XGBoost incorporates various optimization techniques, including column block for parallel learning, regularization to prevent overfitting, and early stopping to save computation time [21]. These features make XGBoost a powerful and efficient tool for tackling complex machine learning tasks [Appendix A.3].

3.1.4.3 Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees to create a more accurate and robust model for various machine learning tasks, such as classification and regression. The primary purpose of a Random Forest is to improve the prediction accuracy and overcome the overfitting problem often encountered with individual decision trees.

Random Forest works by constructing multiple decision trees during the training phase. Each tree is trained on a random subset of the training data, drawn with replacement (bootstrap sampling). Furthermore, a random subset of features is selected at each node when splitting, introducing additional diversity among the trees. The final prediction is obtained by aggregating the predictions of all trees in the ensemble. For classification, this is done by taking a majority vote, while for regression, the average prediction is used [Appendix A.1].

By combining the output of multiple trees, Random Forest reduces the impact of noise and biases present in individual trees, leading to a more accurate and generalizable model. The method is known for its simplicity, versatility, and strong performance across a wide range of applications [22].

3.1.4.4 Naive Bayes

Naive Bayes is a family of probabilistic machine learning algorithms based on Bayes' theorem, which is widely used for classification tasks. The primary purpose of Naive Bayes is to predict the class of an instance by calculating the probability of each class, given the instance's features, and selecting the class with the highest probability.

The "naive" assumption in Naive Bayes is that the features are conditionally independent, given the class. This simplification allows for efficient computation of probabilities, despite the assumption not always being accurate in real-world scenarios. However, Naive Bayes often performs surprisingly well, even when the independence assumption is violated.

To train a Naive Bayes classifier, the algorithm computes the prior probabilities of each class and the likelihood of each feature given each class using the training data. During prediction, it calculates the posterior probabilities of each class for a given instance and selects the class with the highest probability. Naive Bayes is known for its simplicity, speed, and effectiveness, especially in text classification and spam filtering tasks [23].

3.1.4.5 Linear regression

Linear regression is a fundamental machine learning algorithm used to model the relationship between a dependent variable and one or more independent variables. It is primarily employed for predictive modeling, forecasting, and understanding the underlying relationships between variables. Linear regression can be categorized into two types: simple linear regression, which uses a single independent variable, and multiple linear regression, which uses multiple independent variables.

The primary purpose of linear regression is to find the best-fitting line (or hyperplane, in the case of multiple regression) that minimizes the sum of squared differences between the observed and predicted values, known as the residual sum of squares. This best-fitting line is used to make predictions or infer relationships between the variables [24].

Linear regression works by estimating the model parameters (coefficients) that define the line or hyperplane. This is typically achieved using the Ordinary Least Squares (OLS) method, which calculates the coefficients that minimize the residual sum of squares. Alternatively, gradient descent optimization can be used to iteratively update the coefficients until the optimal values are reached [Appendix A.4].

In summary, linear regression is a widely used algorithm for modeling and predicting continuous outcomes based on input features. It provides a simple and interpretable approach to understanding the relationships between variables and making predictions. However, its main limitation is that it assumes a linear relationship between the dependent and independent variables, which may not always hold in real-world scenarios.

3.2 Application

3.2.1 Mobile

3.2.1.1 Android studio

Having considered all the strengths, it was decided to write our application on a platform called Android Studio covering all users. Team IntelliJ IDEA by a well-known corporation released Android Studio IDE (Integrated Development Environment) as displayed in Figure 3.5, which originally was specially constructed for the Android platform to develop apps, but now Flutter is also available. The main reason is that both techniques were released and continue to be developed by Google. Since Android Studio has experienced a large number of updates, additions, and revisions, and now it is an enormous forceful code editor, it represents the best sandbox along with developer tools. To develop applications AS offers lots of features that improve efficiencies, such as a rapid and feature-rich emulator, an environment that can be developed not only for android devices, easy changes with GIT, code manipulation, and modification of resources during runtime with a hot restart. Last year, ide improved dramatically, including Jetpack Compose and Flutter frameworks. As we know Google has a bunch of technologies and methodologies in a Google Cloud Platform and Firebase, and of course, all of these are integrated to make it easy to usage and be user friendly [25].

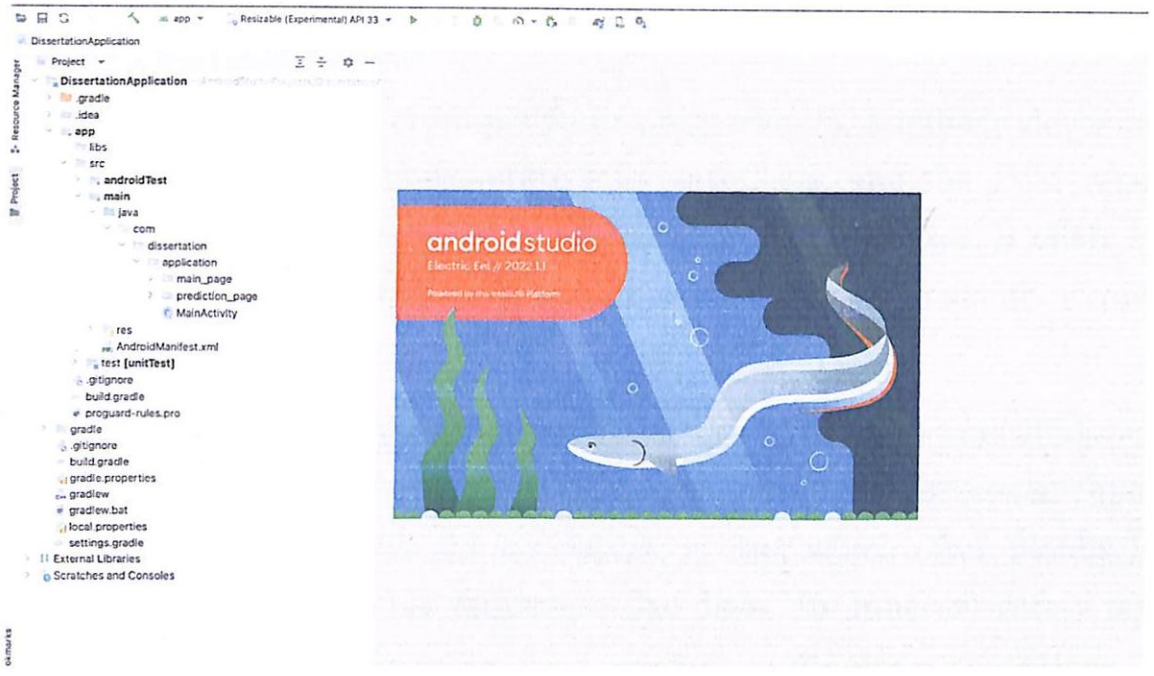

Figure 3.5: Android Studio interface

After creation of new project on Android Studio, it displays project files by default in the Android Project View, which can be changed by selection Project level view. These structures is divided into modules to allow quickly access the important source files for your project[16]. Figure 6 illustrates interface of Android Studio, this IDE was chosen to run Flutter app. In the picture structure of project also shown:

- pubspec.yaml is file where all libraries and environment settings is listed

- assets folder contains images, data and other raw files

- android, ios, web folders contains native settings of each component

- lib folder consist of main flutter files written in dart language for all components

- and other project build gen files [25].

3.2.1.2 Kotlin

Kotlin is a statically typed, cross-platform programming language developed by JetBrains. It's designed to interoperate fully with Java, and the JVM version of its standard library depends on the Java Class Library. However, it offers syntax and feature improvements over Java that make Kotlin more concise, expressive, and safer to use [26].

Kotlin, Figure 3.6, addresses many limitations of Java by providing features like null safety, coroutines for asynchronous programming, extension functions, and infix notation. It's known for its concise syntax which often results in less code written compared to other languages like Java. Its inherent safety features help prevent common programming errors such as null pointer exceptions.

In 2017, Google announced Kotlin as an official language for Android development, which has led to its wide adoption in the Android developer community. Besides Android, Kotlin is also used for backend development, and with the introduction of Kotlin/JS and Kotlin/Native, it's increasingly being used for cross-platform development including web and iOS applications.

Overall, Kotlin offers an excellent balance of object-oriented and functional programming features, enabling developers to write clean, maintainable, and performant code [26].

Figure 3.6: Kotlin Logo

3.2.2 Web

3.2.2.1 Django framework

Django, created by Adrian Holovaty and Simon Willison, was first released in 2005 as an open-source web framework for Python. Its primary goal is to enable

developers to build web applications quickly and efficiently by providing a high-level, reusable set of components. Python and Django are a powerful duo in the world of web development. Python's simplicity and versatility, combined with Django's high-level, reusable components, enable developers to build robust, scalable, and maintain [27].

Key Features of Django:

- Model-View-Template (MVT) Architecture: Django follows the MVT architectural pattern, which separates an application's data (model), presentation (template), and logic (view) layers, promoting modularity, maintainability, and code reusability.
- Built-In Admin Interface: Django comes with a powerful and customizable admin interface, which makes it easy to manage the data and content of a web application.
- Robust Security Features: Django offers built-in security features to protect applications from common web security threats, such as cross-site scripting (XSS), cross-site request forgery (CSRF), and SQL injection.
- Scalability: Django is designed to handle high levels of traffic and can be easily scaled to accommodate growing user bases and increasing demands.
- Extensibility: Django's modular design allows developers to extend its functionality by creating reusable apps or leveraging third-party packages available in the Django ecosystem.

To start developing web applications with Python and Django, developers need to install Python and Django on their local machines. After installation, they can create a new Django project using the command-line interface and begin building their application.

Django follows the "Don't Repeat Yourself" (DRY) principle, encouraging developers to write reusable code and maintain a clean and efficient codebase. This principle, coupled with Python's readability and simplicity, allows developers to create web applications with reduced development time and increased maintainability.

There are numerous resources available for learning Python and Django, including official documentation, online tutorials, books, and video courses. Developers can also benefit from community forums, such as Stack Overflow and Reddit, where they can ask questions, share knowledge, and connect with fellow developers [27].

Chapter 4

Implementation

4.1 Data analysis

Over the last five years, the company has collected an enormous data aimed at understanding various aspects of business operations. This collection of data, spanning half a decade, provides an invaluable resource for discerning patterns, predicting trends, and making informed strategic decisions. A dataset is composed of multiple tables, each tailored to capture distinct views of the business processes. These tables comprise several elements, including customer interactions, data about transactions and different kinds of partners, product performance, and operational efficiency metrics, among others. The five-year span of the data offers a comprehensive overview of the business cycle, capturing fluctuations in different periods and seasons. This timeline allows for accurate trend analysis and forecasting. All data entries have been accurately verified with managers of the company for consistency and credibility to avoid inaccuracies and biases, aiming to extract valuable insights that will guide business growth, enhance customer satisfaction, and ensure operational efficiency in the coming years. My goal was to show that data is not just numbers, but the future success of a business.

The first dataset contains user reviews and ratings gathered after paying for a service, as shown in Table 4.1. Each record comprises a user's subjective feedback and objective rating of a product.

Table 4.1: Reviews dataset first 5 rows

	id	user id	partner id	rating	comment	answer	created at	update at
0	1764432	12831174	659fa43	5	Отличное заведение! на высшем уровне : ассорти...	1	2022-01-11	2022-01-11
1	1117296	10210395	659fa43	5	Крутое впечатление! Очень понравились: Сервис,...	1	2020-01-14	2020-01-14
2	2001033	12325438	b736500	4	Все супер, но лимонады прям на любителя, особе...	0	2022-10-24	2022-10-24
3	1678903	12831174	19a85de	1	Ждали заказ ровно час, очень плохо	1	2021-04-03	2021-04-05
4	1136973	11188786	19a85de	5	Бахадур молодец. Вежливый сотрудник	1	2020-11-09	2020-11-09

In detail, a comprehensive dataset comprises 30,000 rows, each representing data points such as the user, partner, the purchased product, the review text, corresponding rating, and data indicating whether a partner has responded to the user review. This information is essential as it demonstrates the engagement level of partners, and offers insights into the dynamics of customer-partner interaction. The presence of a partner's response can significantly affect customer satisfaction and loyalty. The reviews, both positive and negative, give invaluable firsthand insights into product usability, quality, and potential shortcomings from the customer's perspective. Ratings offer a quantifiable measure of satisfaction that can be analyzed to ascertain overall product appeal, Figures 4.1 and 4.2. By mining this data, we can gain a deeper understanding of customers' experiences and preferences, aiding in tailoring our offerings more effectively.

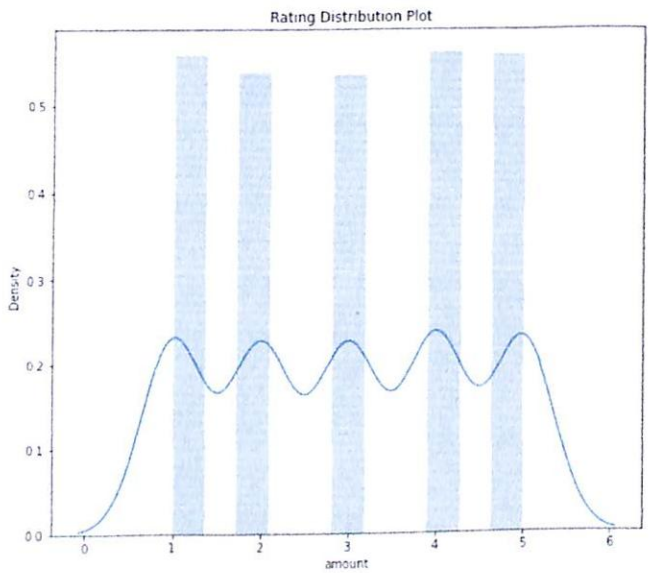

Figure 4.1: Rating Distribution Plot

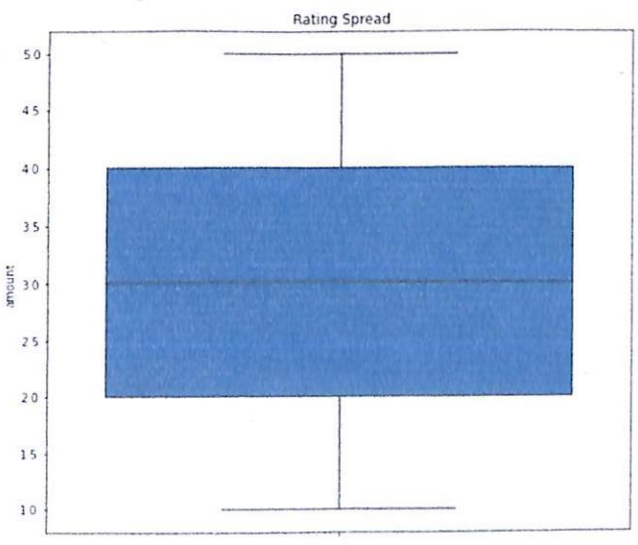

Figure 4.2: Rating Spread Plot

The second dataset is a collection of information representing partners. This includes the branch name, a brief description, the time they partnered with the

company, addresses, activeness status, and their overall rating. Additionally, this dataset also records the number of ratings given to each partner, the count of images they've uploaded, and their respective business categories, Figure 4.3. It provides insights into their performance, their engagement level with our platform, and customer feedback, while the overall rating and rating count serve as key performance indicators. Activeness status and image count could reflect their commitment to the platform. This rich, multi-faceted dataset thus provides a comprehensive picture of our partners' operations, their customer engagement levels, and the overall health of our partner ecosystem. With careful analysis, it can yield significant insights for optimizing our partner management strategies and driving mutual success. Indeed, the dataset also contains valuable information about whether our partners offer takeaways and promotions, Figure 4.4. This data is important as it reflects the partners' efforts to increase customer engagement and drive payments. The takeaway service indicates a partner's flexibility and adaptability, offering services that meet varying customer needs. Promotions are indicative of a partner's marketing strategies. These can be critical factors for customers choosing between different partners, especially in today's digital and convenience-oriented world. Altogether, this dataset, containing 7793 rows, plays a critical role in our partner relationship management strategy, facilitating our efforts to strengthen these partnerships and foster mutual growth.

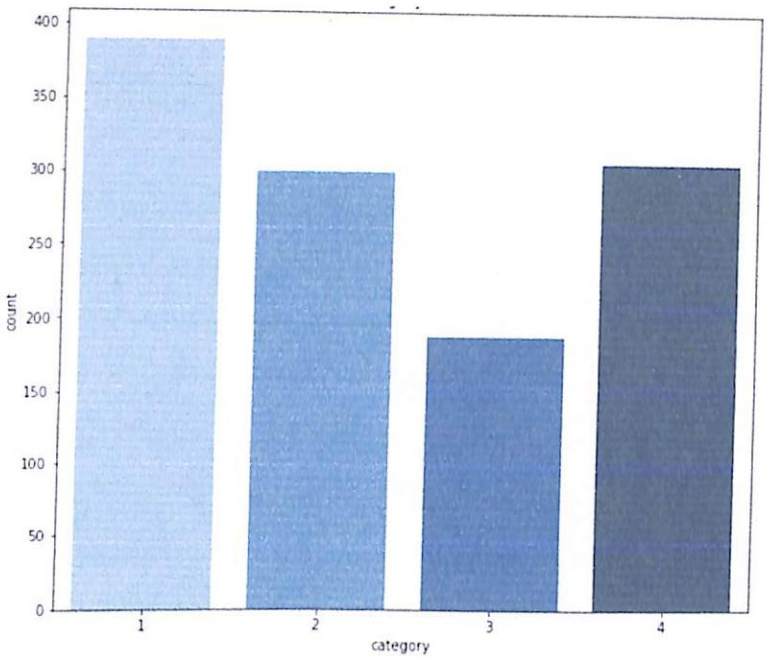

Figure 4.3: Partners count by category

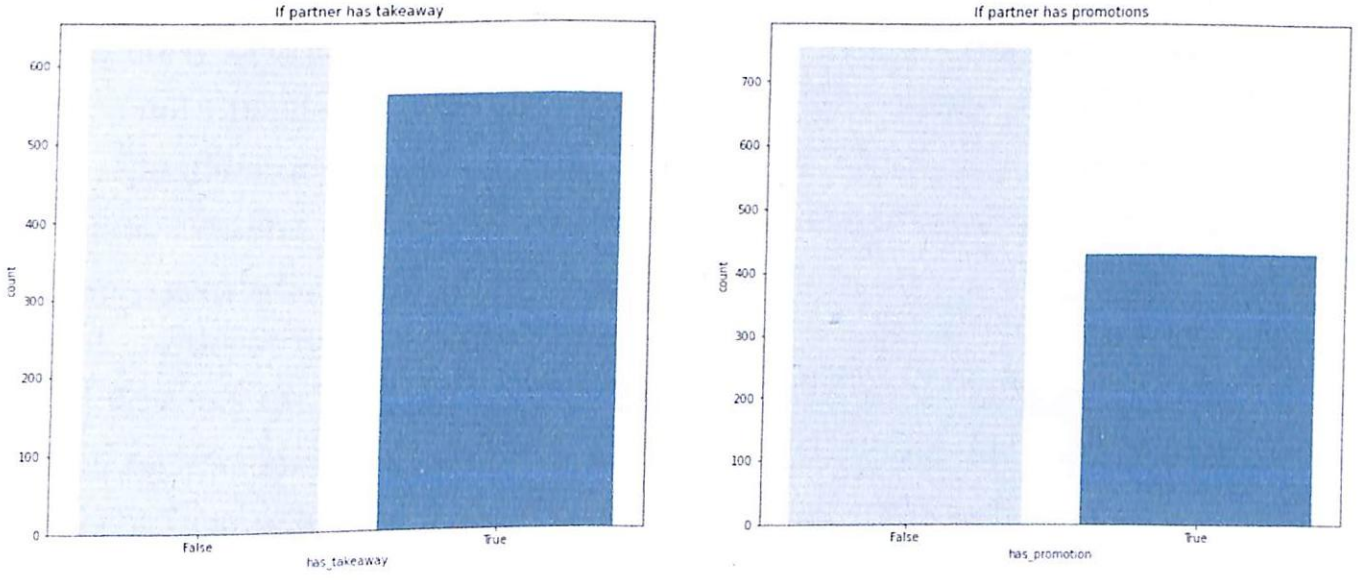

Figure 4.4: Partner additional offers

4.1.1 Data preprocessing

Data preprocessing is a critical step in the analysis, ensuring the data is clean, reliable, and suitable for further processing. This step entails a number of processes, including data cleaning, data transformation, and data integration, also normalizing and scaling numerical data, and encoding categorical data, preparing it for our analytical models.

A. Data Conversion. The initial step in the data preprocessing stage was to convert the data from JSON format, as provided by the API, into a more manageable CSV format. JSON is a widely-used data interchange format that allows for easy human readability and quick network transmission. However, for our data analysis, a tabular format like CSV is more convenient and compatible with numerous data analysis tools and libraries [28]. To accomplish this was created a method for JSON parsing to iterate over objects, extracting the required data fields [29]. Each object corresponds to a row in our CSV file, with the keys mapped to column headers and the corresponding values inserted under those headers. This conversion into CSV format makes the data more accessible, facilitates its manipulation, and readies it for the subsequent steps of data preprocessing.

B. Data Cleaning. The next step involved cleaning the dataset by eliminating noisy data and unnecessary information. Noisy data refers to irrelevant or false data points that can distort analysis. Unnecessary information may include redundant or irrelevant columns that do not contribute to intended analysis. The data cleaning process includes converting all text to a common lower case to ensure consistency and prevent duplication based on case differences. This is crucial, particularly in text-based columns like 'description' and 'reviews', where case differences can lead to the same word being treated as different entries. Text data in reviews and description were further preprocessed through 'stopword' removal and 'lemmatization'. Stopwords are common words such as 'is', 'the', 'and', etc., that add little semantic value to the text. Removing them reduces data size and focuses analysis on meaningful words. Lemmatization is the process of reducing words to their base or root form, which simplifies text data and aids in recognizing patterns. Together, these steps help streamline the data, reducing its complexity and size, while enhancing the quality and reliability of the subsequent analysis. Data cleaning also involved dealing with missing values. Depending on the nature and extent of the missing data, we either filled them in using suitable imputation methods or dropped the rows or columns with excessive missing values. Furthermore, any URLs, numbers, special characters, or HTML tags present in the reviews were removed as they usually do not contribute much value to

analysis. For numerical data, looked for any exceptions that could potentially skew analysis. Depending on their impact, exceptions could be limited, or treated appropriately to ensure they do not unduly influence our results.

C. Vectorizing. Having cleaned and preprocessed data, the next step is 'vectorization'. This process involves converting processed text data into a format that can be understood and used by machine learning algorithms to a numerical format. In this work used several approaches. One of the most common methods of text vectorization is the 'Bag of Words' approach, where each unique word in the text is represented by a number [30]. Another approach is 'TF-IDF', which not only counts word occurrence but also weighs the counts based on how frequent the word is in the entire dataset, helping highlight words that are more unique to each document [31]. For user reviews and partner descriptions applied methods to transform the textual data into a numerical matrix. Each row in the matrix corresponds to a document, and each column corresponds to a unique word from the entire text corpus.

Now, ensuring that the data is clean, consistent, and simplified by minimizing noise and redundancies, maximized the accuracy and efficiency of our subsequent data analyses. The output of this step is a streamlined dataset that's ready for exploration and modeling.

4.2 Model

Having preprocessed and vectorized our data, we are now ready to proceed to the model-building phase. This step involves selecting an appropriate machine learning algorithm and training it on the dataset. Once the model is chosen, we split the dataset into a training set and a test set. The training set is used to train the model – this is where the model learns to make predictions by identifying patterns in the data. The test set is used to evaluate the performance of the model, providing an unbiased assessment of how well the model can generalize to new, unseen data. During the training process, we also tune the hyperparameters of the model to optimize its performance. This could involve adjusting parameters

like the learning rate, the number of trees, or the kernel type. Upon training and tuning the model, we evaluate it using appropriate metrics, such as accuracy, precision, recall, or F1 score for a classification problem, or Mean Squared Error (MSE) or R-Squared for a regression problem. Model building is an iterative process. The aim is to build a robust, accurate model that can effectively answer business questions and provide valuable insights.

A. Review analysis. Given the nature of the data and the task at hand, sentiment analysis using the Naive Bayes classifier is indeed a fitting approach. This methodology will allow us to classify the reviews into positive or negative sentiments based on the text provided by the users. Naive Bayes is a probabilistic machine learning model based on Bayes' theorem with an assumption of independence among predictors. It's a popular choice for text classification due to its simplicity, efficiency, and effectiveness with high-dimensional datasets. Firstly, will use preprocessed and vectorized text data. Each review will be represented as a vector of features, based on the occurrence of words in the review. Next, will train the Naive Bayes classifier on the training subset of the data. During this process, the model will learn the probabilities of each word given a positive or negative sentiment. After the model has been trained, we can use it to predict the sentiment of the reviews in our test dataset. It does this by applying Bayes' theorem to calculate the probabilities of a review being positive or negative based on the words it contains, and classifying the review based on which probability is higher. By performing this sentiment analysis, we can gain valuable insights into the general sentiment towards specific products or partners, and use this information to inform business decisions. We can assess the quality of products, identify areas of improvement, gauge customer satisfaction, and more, all driven by data.

B. Classifying business partners. The next step was to construct a model to categorize partners into different categories based on their attributes such as location, description, ratings, etc. To accomplish this was employed a range of powerful machine learning models, namely Recurrent Neural Networks (RNNs), XGBoost, Naive Bayes, and Random Forest. The differences between these models lie in their handling of data types, their approach to learning, and their ten-

dencies towards overfitting or underfitting. To classify the business category of a partner, we need to consider all relevant information. Sequential data like text description can benefit from the sequential learning ability of RNNs, while other attributes like location or ratings can be well handled by tree-based models like XGBoost or Random Forest. Using these different models and comparing their performances help us build a robust and accurate business category classifier. For each of these models, data were split into training and testing sets, fitted the model on the training data, and then tested the model's predictions against the actual categories in the test set. By tuning hyperparameters and optimizers, we sought to maximize the accuracy of our partner classification models. Each model was evaluated based on its classification accuracy and the results were then compared to select the best-performing model. In the 'Algorithms' section, the specifics of each algorithm used are described deeper. We carefully discuss the parameters that were used for each model and the techniques applied to enhance the model's performance: for RNNs including the number of layers and nodes in each layer, the activation functions used, and the optimizer used; for XGBoost and Random Forest described the critical hyperparameters, like the number of decision trees, maximum depth of the trees, learning rate, and how these parameters were optimized. Overall, that section provided a comprehensive view of how each model was set up, tuned, and optimized for the task of classifying partners into different business categories based on various attributes.

C. Predicting business profit. The final step of analysis involved predicting business profits. For this purpose, employed Linear Regression, a simple yet powerful statistical and machine learning method used to understand the relationship between variables and make predictions. The dataset for this phase contained information about the partner, their annual amounts, and the weighted results of previous analyses - the review sentiment analysis and the partner category classification. The annual amount for a partner represents their total business revenues for the year. The Linear Regression model was trained with these variables as predictors to predict the profit of the business. Hypothesized that the annual amount, the sentiment around the business, and the business category (as encoded by their respective weights) would have an influence on the profit. Each partner's profits

were then predicted based on the trained Linear Regression model. In this way, work brings together a range of analyses to make practical, valuable predictions for businesses, allowing them to leverage their review and partner data in meaningful ways. This model provides a valuable tool for businesses to anticipate their profits based on data they already have or can acquire. These profit predictions can influence several critical decisions, including resource allocation, budgeting, marketing strategies, and more. However, as with any model, it's crucial to bear in mind that these are predictions and actual profits can differ due to various unforeseen factors. Thus, while these predictions provide valuable insights, they should be one of several tools used in decision-making processes.

4.3 Evaluation

A. Review analysis evaluation. A confusion matrix is a great tool for visualizing the performance of a classification model to the Naive Bayes classifier used in sentiment analysis. The matrix provides a summary of the correct and incorrect predictions broken down by each category. After building and applying Naive Bayes classifier on the test data, calculated values to fill in the confusion matrix. This provides a clear overview of how well the model performed, not just in terms of the overall accuracy, but also in other measures like precision, recall, and F1 score. For example, a model with high precision but lower recall accurately identifies positive reviews but misses a good number of them. Similarly, a model with high recall but lower precision is capturing most of the positive reviews but incorrectly labels many negative ones as positive. The confusion matrix, therefore, is an invaluable tool for evaluating and refining our sentiment analysis model.

$$
\begin{aligned}
Accuracy &= \frac{TP+TN}{Total} \\
Precision &= \frac{TP}{TP+FP} \\
Recall &= \frac{TP}{TP+FN} \\
F1 &= 2 * \frac{Precision*Recall}{Precision+Recall}
\end{aligned}
\tag{4.3.1}
$$

Table 3, the confusion matrix for reviews, provides a clear visual summary of the model's performance. The matrix categorizes the model's predictions into four segments: True Positives (TP), False Negatives (FN), False Positives (FP), and True Negatives (TN). The model correctly identified 14,705 reviews as positive and 7,504 reviews as negative. However, it incorrectly categorized 3740 positive reviews as negative and 4051 negative reviews as positive. Next, based on the confusion matrix, calculated crucial performance metrics using Equation 4.3.1, which results are presented in Table 4. The four metrics calculated are Accuracy, Precision, Recall, and F1 Score. Accuracy is the proportion of total correct predictions (both positive and negative) out of all predictions. Precision refers to the proportion of correct positive predictions out of all positive predictions. Recall is the proportion of correct positive predictions out of all actual positive instances. The F1 Score is the harmonic mean of Precision and Recall, providing a balance between the two. These metrics together help in understanding the model's performance beyond just accuracy, giving insights into its strengths and weaknesses and guiding us toward potential improvements. Summarizing this indicates that the model is performing quite, with a good balance between precision and recall.

B. Classifying business partners. Evolution of model accuracy across training iterations for three different algorithms illustrated in Table 4.4: for RNN it is epochs, for XGBoost and a Random Forest classifier it is tree count. An iteration corresponds to one cycle through the full training dataset. In this training, the model's accuracy is recorded every 50 iterations until it reaches 500. By observing the table, we can infer several key insights. All models show an upward trend in

Table 4.2: Confusion matrix for reviews

	Actual Positive (True)	Actual Negative (False)
Predicted Positive	14705	4051
Predicted Negative	3740	7504

Table 4.3: Evaluation of confusion matrix

Metrics	Result
Accuracy	68.33%
Precision	78.4%
Recall	66.2%
F1 Score	71.8%

accuracy as the number of iterations increases. This is typical during the training process, as models learn to better fit the data over time. The Recurrent Neural Network model starts with an accuracy of 0.68 at the 50th epoch and improves to 0.86 by the 500th epoch. The XGBoost model begins with an accuracy of 0.72 at the 50th epoch and shows a consistent improvement, reaching an accuracy of 0.82 by the 500th epoch. The Random Forest model starts with an accuracy of 0.69 at the 50th epoch and increases its accuracy to 0.81 by the 500th epoch. It's important to note that increasing repetitions often leads to better model performance on the training set.

Table 4.4 demonstrates how the model's accuracy evolves as the training progresses, visual representation of how the model's accuracy changes over iterations. Observing the trend in this table allows us to understand how quickly the model is learning and at what point the model might be overfitting. Also, it is easier to spot trends and inflection points in a graphical format, making this a useful tool for monitoring the model's learning process. Figures 4.5 and 4.6 provide a direct comparison of the final accuracies achieved by each model after training is complete. This is a valuable resource for determining which model performed best and comparing the final loss values for each model gives insight into which model has learned the task most effectively. The neural network showed the best results in terms of performance metrics, it suggests that the algorithm was most successful at learning the patterns in data and classifying the business categories.

Table 4.4: Accuracy per iteration

Iterations	RNN	XGBoost	RandomForest
50	0.68	0.72	0.69
100	0.72	0.76	0.72
150	0.76	0.78	0.72
200	0.78	0.77	0.73
250	0.79	0.79	0.75
300	0.81	0.78	0.76
350	0.82	0.80	0.77
400	0.83	0.81	0.79
450	0.85	0.80	0.80
500	0.86	0.82	0.81

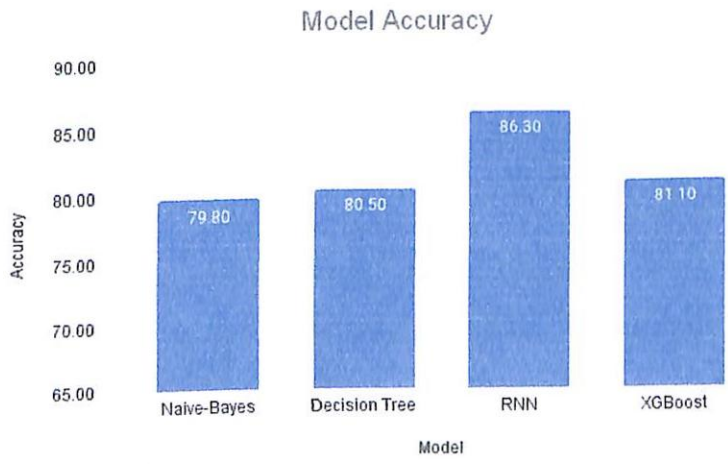

Figure 4.5: Overall accuracy comparison for different models

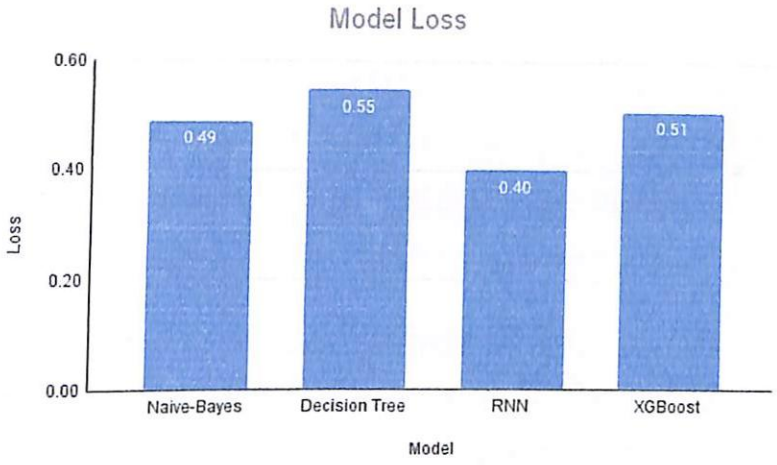

Figure 4.6: Overall model loss comparison for different models

4.4 Prediction Visualization

It's important to note the crucial role that user interactivity and engagement fulfilling one of the primary objectives of this research. It was decided to create applications on different platforms for visualization: website and android mobile versions. The primary challenge was the incompatibility between the python-based predictive model and the intended visualization platforms. To resolve this issue and achieve one of the main goals of the thesis, the model was saved after development. Upon saving the model, a REST API was built using Django, which is a robust and flexible web framework for Python. The primary function of this API was to facilitate the interaction between the saved predictive model and the web and mobile applications [32]. This API allowed the prediction results from the model to be retrieved and used in a compatible format for the applications.

The web application comprises two main pages. The first page presents diagrams that represent the current situation. This visual display of data enables managers to understand the prevailing circumstances in a simplified and user-friendly manner. The second page consists of an interactive fill-form about the new partner. When a manager or other co-worker completes the form, the responses are processed by the REST API, which communicates with the saved prediction model [32]. Based on this input, the model generates a prediction of potential profits. This prediction is then displayed to the user, both numerically and through a diagram, allowing for an easy understanding of future profit

scenarios.

Furthermore, a mobile application was also developed to increase accessibility and provide a convenient way for users to use the tool from anywhere. This comprehensive solution effectively demonstrates the fulfillment of the thesis' goal to develop an accessible, user-friendly platform for visualizing and predicting data.

Furthermore, concrete examples of visualizations, as depicted in Figures B.1 through B.5. These figures present snapshots of the actual visualizations that users see when they interact with the web and mobile applications, serving to illustrate the practical application of the predictive model and the user interface design. These figures serve as key evidence in the dissertation, effectively demonstrating the successful achievement of the research goal to create an accessible and user-friendly platform for data visualization and profit prediction.

Chapter 5

Conclusions and future work

This chapter summarizes the whole research with its matter findings related to aims and motivation. The goal of this work was to make an application to predict business orientation and profit of company. An intuitive, user-friendly, and adaptive application was designed and built to achieve the purpose, it was constructed on multiple platforms web-based browsers and android operating system mobile phones, so that a managers and other co-workers could use it.

In this study, we embarked on a comprehensive exploration of datasets collected over the past five years by a company: reviews, partners, payments dataset. The collected data, presented in a rich set of tables and spanning thousands of records, provided the basis for our analysis and model building. Sentiment analysis of user reviews was conducted using a Naive Bayes model, while partner categorization was performed using RNN, XGBoost, and RandomForest models. Predictions of business profits were generated using linear regression model. The sentiment analysis revealed crucial information regarding user perception of products and services. The RNN model proved to be the most effective at categorizing partners, underscoring its superior performance in handling sequential data. Linear regression predictions offered actionable insights for future resource allocation and investment strategies. The implications of these findings are by leveraging these technologies, businesses can gain an in-depth understanding of customer behavior, evaluate partner performance, and predict future profitability.The results illustrate the significant potential of machine learning in deriving

actionable insights from data. Accurate categorization of partners aids in the strategic planning of collaborations, and predicting profits empowers businesses to make informed financial decisions.

Despite the valuable insights, our study is not without limitations. The accuracy of the models heavily depends on the quality and quantity of data, and showed only 80%, although more needed. Any inconsistencies or biases in the data could influence the model's performance. Future research could explore the integration of other machine learning models to enhance prediction accuracy and to consider more diverse data inputs. Time-series analysis could be used to predict business profits taking into account seasonal variations. Regular reassessment of the chosen models is recommended to keep pace with advancements in the field.

In conclusion, this study demonstrates that even without a dedicated data science team and limited companies can leverage existing data to extract insightful patterns, make informed decisions, and forecast future trends. By utilizing accessible machine learning techniques, businesses can perform tasks such as sentiment analysis, partner categorization, and profit prediction. This not only makes their operations more data-driven but also empowers them to make strategic decisions based on robust, evidence-backed models. The insights derived from such an exercise underscore the immense potential of affordable AI and machine learning tools in transforming the way businesses operate, regardless of their size or resources.

Bibliography

[1] Churn rate: What it means, examples, and calculations. URL https://www.investopedia.com/terms/c/churnrate.asp. Accessed: 2022-04-05.

[2] Reznichenko E.V.; Matveev M.G. Development of a software tool to predict customer outflow using machine learning. Collection of student scientific works of the faculty of computer sciences vsu, 11:287–292, 2017.

[3] Meldeby M.A.; Sarbasova A.K. Analysis of opinions of buyers based on machine learning. Al-Farabi Kazakh National University, pages 360–364, 2017.

[4] Amit Ganesh Upadhye; A.C Lomte. Customer opinion assessment using artificial neural networks and machine learning. IJAR, pages 1829–1835, July 2020. doi: 10.21474/IJAR01/11451. URL http://dx.doi.org/10.21474/IJAR01/11451.

[5] Xie Haihua; Yang Jingwei; Carl K. Chang; Lin Liu. A statistical analysis approach to predict user's changing requirements for software service evolution. The Journal of Systems Softwaren, 132:147–164, 25 June 2017. doi: 10.1016/j.jss.2017.06.071. URL https://www.sciencedirect.com/science/article/abs/pii/S0164121217301358?via.

[6] Huang Li; Yang Shihai. Short-term load prediction based on user behaviors analysis. ICSGSC, pages 18–23, 2020. URL https://ieeexplore.ieee.org/document/9248540.

[7] Kamorudeen A. Amuda; Adesesan B. Adeyemo. Customers churn prediction in financial institution using artificial neural network. Inter-

national Multi-Conference on ICT Applications, December 2019. URL https://www.researchgate.net/publication/338096967_Customers_Churn_Prediction_in_Financial_Institution_Using_Artificial_Neural_Network.

[8] Li Chen Chengm; Chia-Chi Wu; Chih-Yi Chen. Behavior analysis of customer churn for a customer relationship system: an empirical case study. Journal of Global Information Management, 27(1):111–127, January 2019. doi: 10.4018/JGIM.2019010106. URL https://www.igi-global.com/gateway/article/215025.

[9] Chunjiang Li. Application of machine learning algorithms in the stock market analysis. FMIBM, 10:352–358, 2023. doi: 10.54097/hbem.v10i.8119. URL https://doi.org/10.54097/hbem.v10i.8119.

[10] Boyuan Zhang. Customer churn in subscription business model - predictive analytics on customer churn. FIBA, 44:870–876, April 2023. doi: 10.54691/bcpbm.v44i.4971. URL https://doi.org/10.54691/bcpbm.v44i.4971.

[11] Zhichong Li. Research on sales strategy of electric vehicle target customers based on machine learning algorithm. MMML, 22:270–278, December 2022. doi: 10.54097/hset.v22i.3388. URL https://doi.org/10.54097/hset.v22i.3388.

[12] Sriramakrishnan Chandrasekaran; Abhishek Kumar. A clustering approach for customer billing prediction in mall: A machine learning mechanism. Journal of Computer and Communications, 7:55–66, December 2022. doi: 10.4236/jcc.2019.73006. URL https://doi.org/10.4236/jcc.2019.73006.

[13] Thomas H. Cormen; Charles E. Leiserson; Ronald L. Rivest; Clifford Stein. Introduction to algorithms. 4, April 2022.

[14] Anaconda official documentation. URL https://docs.anaconda.com/. Accessed: 2022-04-15.

[15] What is python? executive summary. URL https://www.python.org/doc/essays/blurb/. Accessed: 2022-04-15.

[16] Gitlab is the most comprehensive ai-powered devsecops platform. URL https://docs.gitlab.com/. Accessed: 2022-04-15.

[17] Docker overview get started. URL https://docs.docker.com/get-started/overview/. Accessed: 2022-04-16.

[18] Debasish Kalita. A brief overview of recurrent neural networks (rnn). March 2022. URL https://www.analyticsvidhya.com/blog/2022/03/a-brief-overview-of-recurrent-neural-networks-rnn/.

[19] Recurrent neural network. URL https://en.wikipedia.org/wiki/Recurrent_neural_network.

[20] Xgboost documentation. URL https://xgboost.readthedocs.io/en/stable/.

[21] Tianqi Chen; Carlos Guestrin. Xgboost: A scalable tree boosting system. the 22nd ACM SIGKDD International Conference, August 2016. doi: 10.1145/2939672.2939785. URL https://dl.acm.org/doi/10.1145/2939672.2939785.

[22] Jehad Ali1; Rehanullah Khan; Nasir Ahmad; Imran Maqsood. Random forests and decision trees. IJCSI, 9(5):272–278, September 2012. URL https://ijcsi.org/papers/IJCSI-9-5-3-272-278.pdf.

[23] Feng-Jen Yang. An implementation of naive bayes classifier. 2018 CSCI, December 2018. doi: 10.1109/CSCI46756.2018.00065. URL https://ieeexplore.ieee.org/document/8947658.

[24] Khushbu Kumari; Suniti Yadav. Linear regression analysis study. Journal of the Practice of Cardiovascular Sciences, 4(1):33–36, January 2018. doi: 10.4103/jpcs.jpcs_8_18. URL https://www.j-pcs.org/article.asp?issn=2395-5414;year=2018;volume=4;issue=1;spage=33;epage=36;aulast=Kumari.

[25] Meet android studio. URL https://developer.android.com/studio/intro.

[26] Kotlin docs - get started with kotlin. URL https://kotlinlang.org/docs/home.html. Accessed: 2022-04-16.

[27] Django-the web framework for perfectionists with deadlines. URL https://www.djangoproject.com/. Accessed: 2022-04-15.

[28] Comma-separated values. URL https://en.wikipedia.org/wiki/Comma-separated_values.

[29] Introducing json. URL https://www.json.org/json-en.html.

[30] Wisam Abdulazeez Qader Musa M.Ameen Bilal I. Ahmed. An overview of bag of words;importance, implementation, applications, and challenges. 2019 International Engineering Conference (IEC), January 2020. doi: 10.1109/IEC47844.2019.8950616. URL https://ieeexplore.ieee.org/document/8950616.

[31] Stephen E. Robertson. Understanding inverse document frequency: On theoretical arguments for idf. Journal of Documentation, 60(5):503–520, January 2020. doi: 10.1108/00220410410560582. URL https://www.emerald.com/insight/content/doi/10.1108/00220410410560582/full/html.

[32] What is a rest api? URL https://www.redhat.com/en/topics/api/what-is-a-rest-api.

Appendix A

Apendex A

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV

classifier = RandomForestClassifier(n_estimators=100, random_state=42)
classifier.fit(X_train, y_train)

y_pred = classifier.predict(X_test)
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
print("Accuracy:", accuracy_score(y_test, y_pred))

param_grid = {
    'n_estimators': [50, 100, 200, 300],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'bootstrap': [True, False]
}

hp_grid_search = GridSearchCV(
    estimator = RandomForestClassifier(random_state=42),
    param_grid = param_grid,
    cv=5,
    n_jobs=-1,
    verbose=2
)

hp_grid_search.fit(X_train, y_train)
```

```
print("Best hyperparameters:", hp_grid_search.best_params_)

best_model = hp_grid_search.best_estimator_
y_pred = best_model.predict(X_test)

print('-------------------------------------------------------')
print(classification_report(y_test, y_pred))
print('-------------------------------------------------------')
print("Accuracy:", accuracy_score(y_test, y_pred))
```

Figure A.1: Building RandomForest Classifier

```
from sklearn.preprocessing import MinMaxScaler, LabelEncoder
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import LSTM

model = Sequential()
model.add(LSTM(units=256, activation='relu', return_sequences=True,
↪  input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(LSTM(units=64, activation='relu'))
model.add(Dense(units=32, activation='relu'))
model.add(Dense(units=len(np.unique(y_train)), activation='softmax'))

model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
↪  metrics=['accuracy'])

history = model.fit(X_train, y_train, epochs=100, batch_size=32, validation_split=0.1,
↪  verbose=1)

y_pred = model.predict_classes(X_test)

print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
print("Accuracy:", accuracy_score(y_test, y_pred))

print(model.summary())
```

Figure A.2: Building RNN model

```
import xgboost as xgb
from sklearn.model_selection import RandomizedSearchCV
```

```
classifier = xgb.XGBClassifier(
    use_label_encoder = False,
    objective = 'multi:softmax',
    num_class = len(np.unique(y_train)),
    seed=42
)
classifier.fit(X_train, y_train)

y_pred = classifier.predict(X_test)

print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
print("Accuracy:", accuracy_score(y_test, y_pred))

param_grid = {
    'learning_rate': [0.01, 0.05, 0.1, 0.2],
    'max_depth': [3, 5, 7, 10],
    'n_estimators': [100, 200, 300, 500],
    'subsample': [0.5, 0.75, 1],
    'colsample_bytree': [0.5, 0.75, 1],
    'gamma': [0, 0.1, 0.2, 0.3],
    'min_child_weight': [1, 3, 5]
}

random_search = RandomizedSearchCV(
    estimator = xgb.XGBClassifier(
        use_label_encoder=False,
        objective='multi:softmax',
        num_class=len(np.unique(y_train)),
        seed=42
    ),
    param_distributions=param_grid,
    n_iter=50,
    cv=5,
    n_jobs=-1,
    verbose=2,
    random_state=42
)

random_search.fit(X_train, y_train)

print("Best hyperparameters:", random_search.best_params_)

best_model = random_search.best_estimator_
y_pred = best_model.predict(X_test)
```

```
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
print("Accuracy:", accuracy_score(y_test, y_pred))
```

Figure A.3: XGBoost Classifier

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score,mean_squared_error
import numpy as np
from sklearn.linear_model import Ridge
from sklearn.model_selection import GridSearchCV

df = main_df
X = df.drop(['amount', 'name'], axis = 1)
y = df['amount']
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.75, test_size =
↪  0.25, random_state = 100)
print(X)
print(y[:5])
model = LinearRegression()
model.fit(X_train,y_train)
y_pred = model.predict(X_test)
print('mean squared error :',mse)
print('r square :',rsq)
#regularization
ridge = Ridge()
params = {'alpha': np.logspace(-4, 4, 100)}
grid_search = GridSearchCV(ridge, params, cv=5, scoring='neg_mean_squared_error')
grid_search.fit(X, Y)
print("Best alpha:", grid_search.best_params_['alpha'])
print("Coefficients:", grid_search.best_estimator_.coef_)
```

Figure A.4: Building Linear Regression

Appendix B

Apendex B

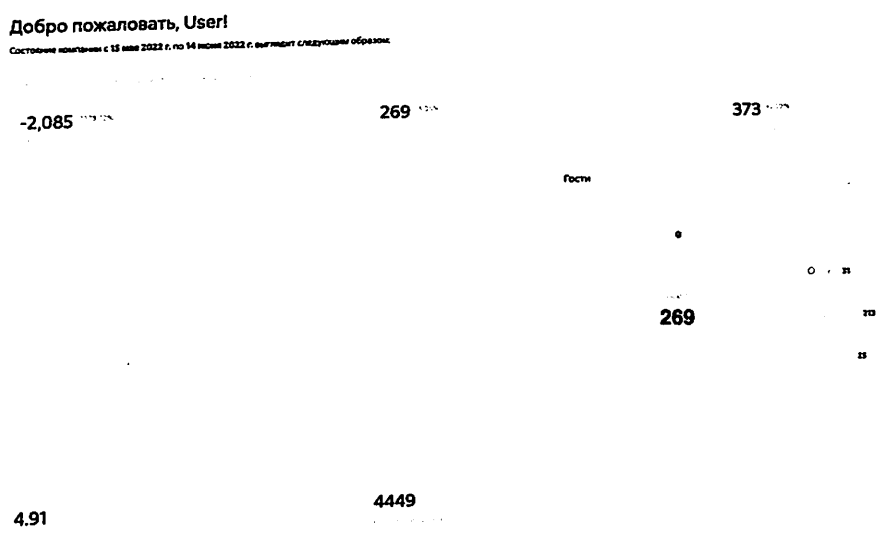

Figure B.1: Home page. Common analysis

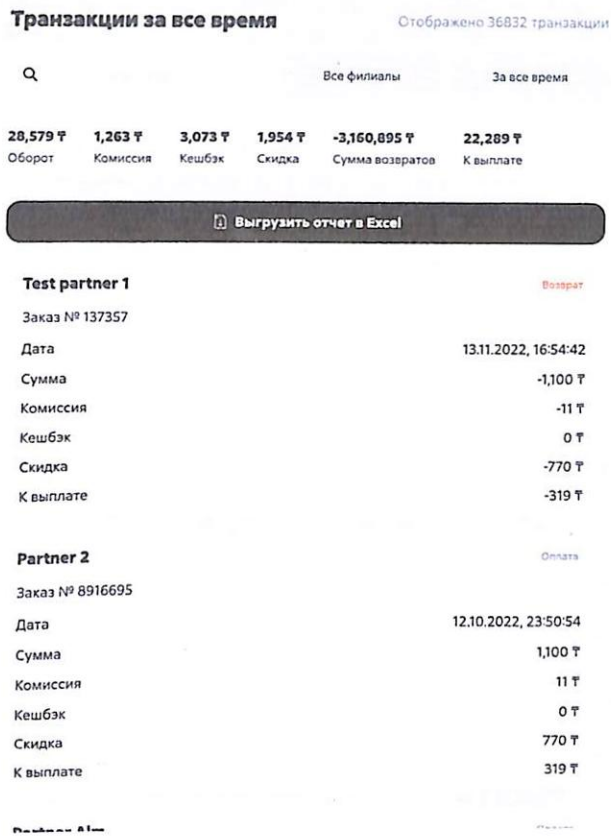

Figure B.2: Home page. Transactions analysis

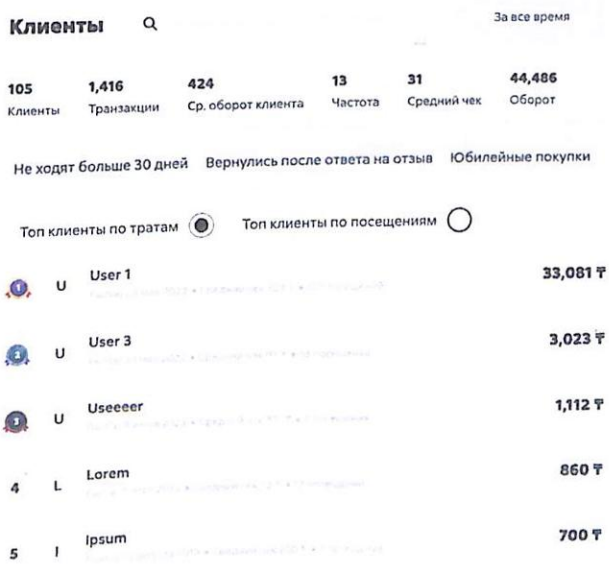

Figure B.3: Home page. Customers analysis

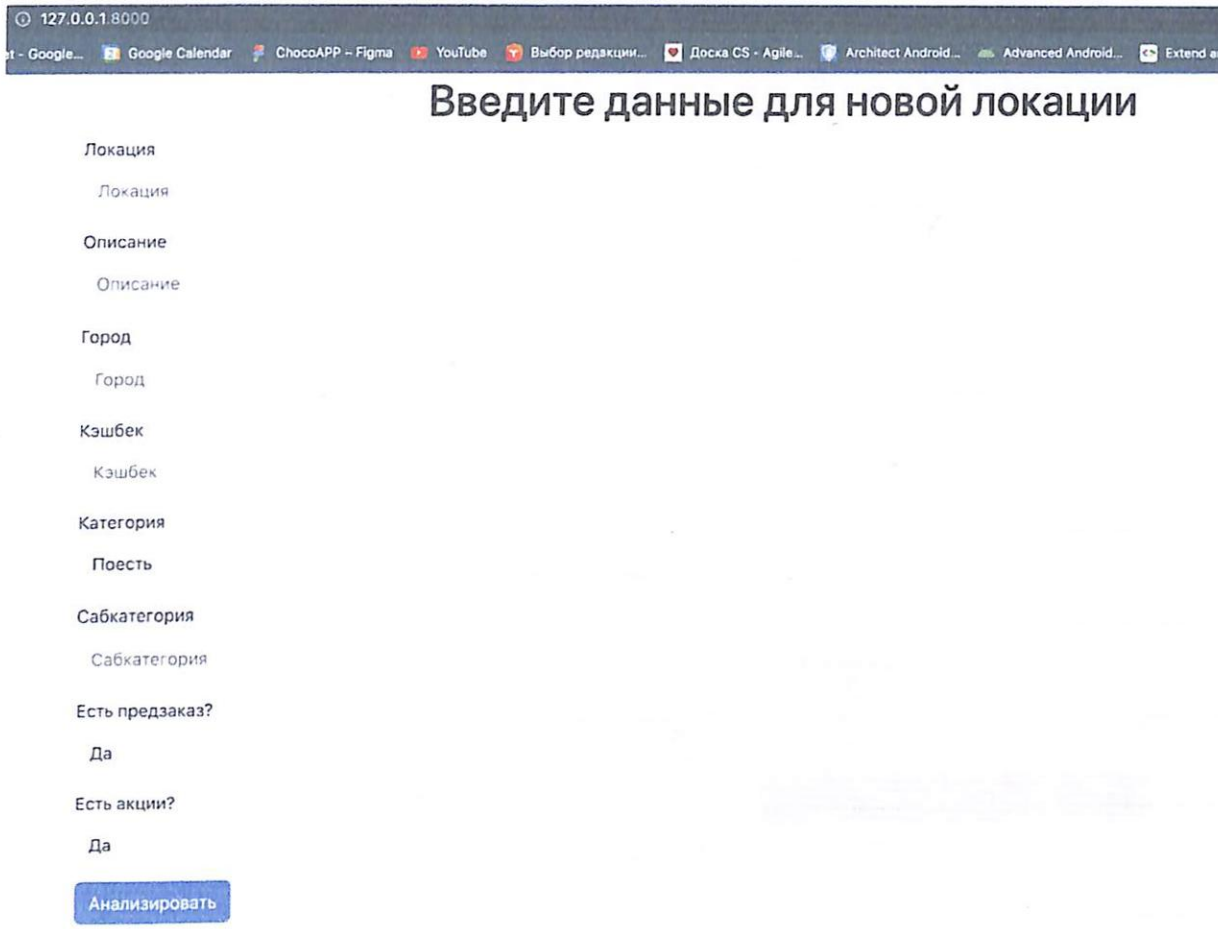

Figure B.4: Prediction fill-form page of web version

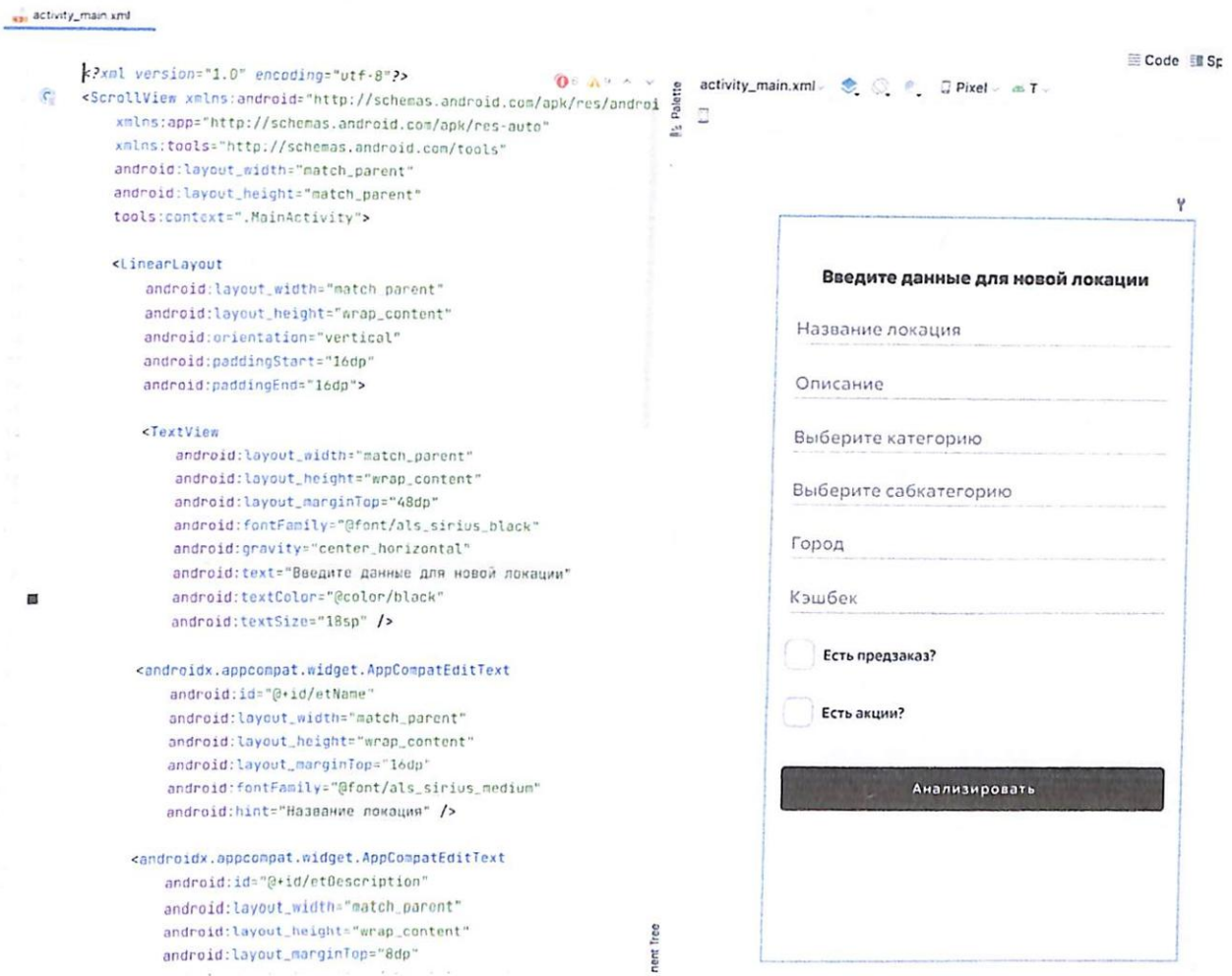

Figure B.5: Prediction fill-form page of mobile version