Design

Qiupu Shen, Wencheng Zhang, Xiaohang Tang, Yixing Lu, Yongyin Yang, Yunfan Shi

Contents

1 System Architecture 1
2 Data 3
3 Processes & Algorithms 5
4 Interface 12
5 Design of Evaluation Process.....15
6 Reference.....17

1 System Architecture

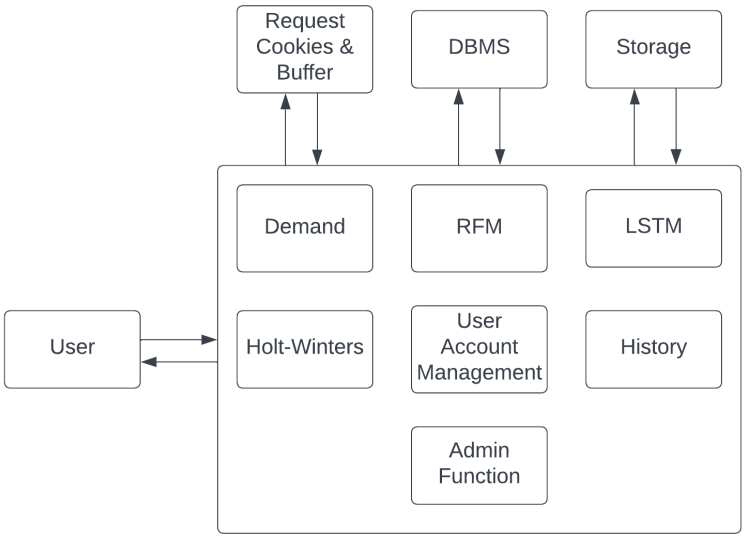


Figure 1: An overview of our system architecture

As is shown in Figure 1, the system consists of seven components: Demand/RFM/LSTM(AI)/Holt-Winters (Statistical); each includes related upload/download as well as user account related systems. The guest could not access LSTM, Holt-Winters and admin functions. A registered user could access

all except admin functions. The system employs Request POST, cookies, browser buffer, DBMS and local file storage for data communication/persistence and file I/O.

After choosing the function, users must first upload a valid formatted file. This file upload functionality is identical for each of the four core functions. After the file upload is finished, results will be sent to frontend and plotted by Function Plot (Demand), Treemap (RFM) and Time Axis Line Chart (LSTM/Holt-Winters) via Apache ECharts framework (ECharts.apache.org, n.d.), while an ECharts boxplot will visualise the statistical features of raw data. According to Lima (2021), the backend LSTM (TensorFlow pre-trained model inference) /Holt-Winters will make the prediction, and they will be updated in the plot.

1.1 Customer persona analysis

After choosing the RFM function (CleverTap, 2018) and the completion of upload, our backend implementation based on Pandas and data aggregation operations will segment customers into groups. The JSON data delivered by backend will be divided into random groups by an algorithm developed by us and fit into the Treemap on the user interface. Box plot of statistical features shares similar logic as above.

1.2 Demand curve pricing analysis

As for demand curve fitting and pricing strategy (insightr, 2018), the backend will use scipy (GeeksforGeeks, 2019) curve fitting library to fit the dataset using the pre-defined mathematic model for demand curve. After optimising the objective function which determines how well the curve fits the data, the parameters of the best candidate will be passed to frontend. The frontend will design corresponding functions based on backend parameters and draw the data points outputted from this function using function plot in ECharts.

1.3 Time-series business metric prediction

A pre-trained LSTM model and a statistical algorithm are designed to realise the forecast function. To realise the requirement, we decided to focus on time series (we chose LSTM). However, we discovered a bottleneck when applying the model. From our search of dataset use examples (ADAM, 2021) and code examples (Lima, 2021), we found that in most cases, the research focus is on aspects like accuracy evaluation of the model and general LSTM introduction rather than practical use. Thus, two problems were found:

1. Many proposed methods are too shallow to be useful or applicable.
2. From further reading of conference papers (Rizal, Soraya and Tajuddin, 2019; Hamami and Dahlan, 2020; Selvin et al., 2017), most LSTM

practices introduce a method named "Sliding Window". It is a method that uses more than a one-time variable to predict the next step value. For example, use the value at t and t+1 to predict the value at t+2. Nevertheless, it can only predict the next step value (or only one value).

In our project, however, we try to achieve the forecast of metrics (like profit and sales amount) in a long continuous period specified by the user, which requires feeding the values of the previous period. However, these values are unknown to us based on the principle mentioned below.

For example, suppose it is 1 Apr now, and we have profit data for previous months. The situation is that if we want to forecast the profit of 30 Apr, the model requires us to feed, for example, data of 27 Apr to 29 Apr. However, the fact is we do not have the data since future data is unavailable now.

Therefore, according to our investigation, there are no existing LSTM practices adaptable to our project. This would be the rationale for our implementation.

Design thoughts:

We devised an algorithm to solve the second problem mentioned above, which will work during the inference stage after training. The principle can be briefly described as whenever the model predicts a row from historical data; the result will be added to historical data and used for later predictions. However, the propagated error rate which we could not effectively contain is considered a major limitation.

2 Data

2.1 Datasets

Demand:

This dataset contains beef quarterly sales data (Quantity & Price) from 1977 to 1999 (Li, 2018).

RFM:

This dataset is the actual customer sales data (invoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, Country) in Sephora (FABIENDANIEL, 2019).

Time-Series:

The Gold-Price Dataset is a two-column time-series dataset (Date, Value) that

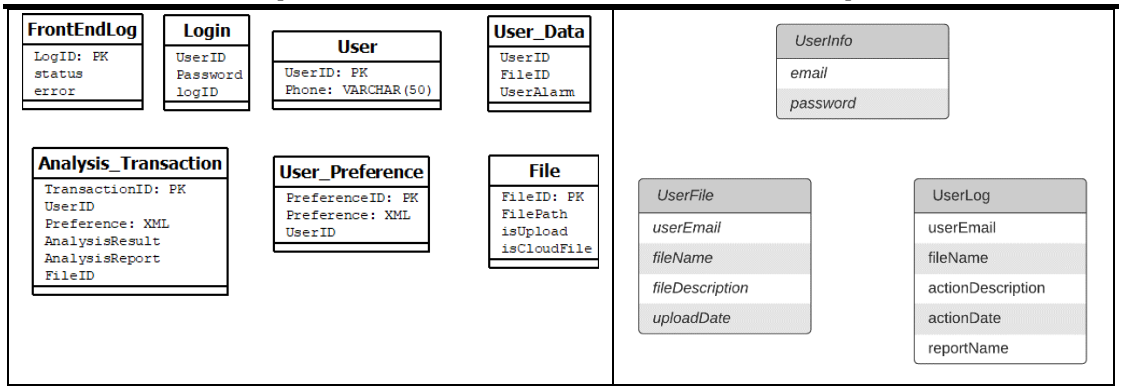
contains data from 1970-01-01 to 2020-03-13 (MÖBIUS, 2021).

All these datasets are obtained from Kaggle's public ones with licenses, permissions and legal statements provided. Hence, we confirm that they are used with permission and that personal privacy is respected.

2.2 Database design

DB ER diagram:

initial plan **actual implementation**



Data dictionary:

a) Descriptions of relationships:

Entity	Multiplicity	Relationship	Multiplicity	Entity
User	1..*	Uploads	1..1	file
File	1..1	Is in	1..1	Report
User	1..*	Possesses	1..1	Report

Logical Table Structure:

UserInfo(email, password) **Primary Key** email	**UserFile**(userEmail, filename, fileDescription, uploadDate) **Foreign Key** userEmail **references** UserInfo(email)
UserLog(userEmail, filename, actionDescription, actionDate, reportName) **Foreign Key** userEmail **references** UserInfo(email)	

Transaction Matrix:

Table	register				login				logout				upload				Result generate				history				admin			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
UserInfo	X	X				X																				X		
UserFile													X		X							X						
UserLog																	X		X			X						

I=Insert; R=Read; U=Update; D=Delete

Business Policies:

1.Demand and RFM: open to guests
2.Time series (LSTM/Holt-Winters): registered user only
3.Download analysis image: open to guests
4.Download PDF report/check history: registered user only

Business Rules:

1.One user could upload any number of files where each upload file could only be related to one analysis result/report.
2.Upload file format constraint:

a) All upload files must be in .csv format
b) Demand:

```
Year,Quarter,Quantity,Price
1977,1,22.9976,142.1667
```

c) RFM:

```
InvoiceNo,StockCode,Description,Quantity,InvoiceDate,UnitPrice,CustomerID,Country
536365,85123A,WHITE HANGING HEART T-LIGHT HOLDER,6,12/1/2010 8:26,2.55,17850,United Kingdom
536365,71053,WHITE METAL LANTERN,6,12/1/2010 8:26,3.39,17850,United Kingdom
536365,84406B,CREAM CUPID HEARTS COAT HANGER,8,12/1/2010 8:26,2.75,17850,United Kingdom
```

d) Time series:

Time	Any metric
1970-01-01	35.2

3 Processes & Algorithms

3.1 Processes

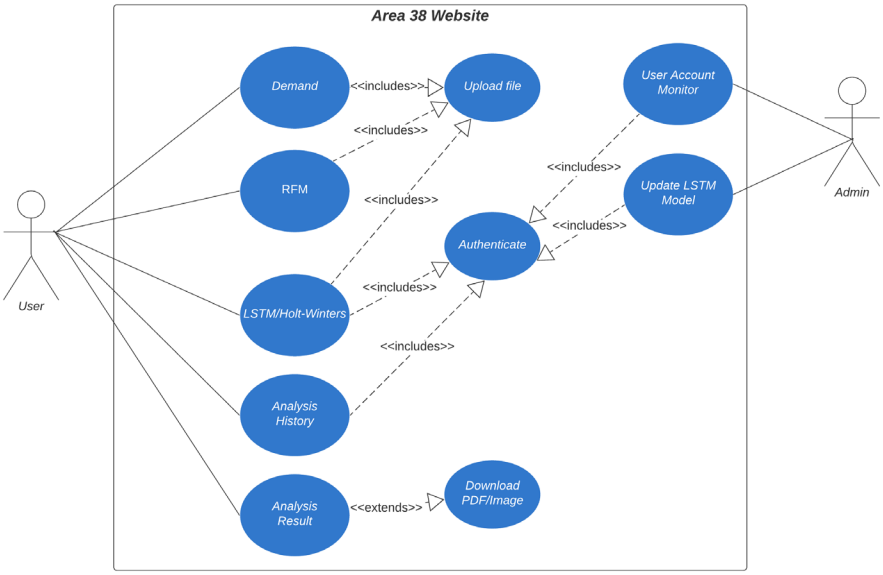


Figure 2: System use case diagram. Note some contents are simplified for the clarity of the above diagram

As is shown in Figure 2, guests can use Demand and RFM while LSTM/Holt-Winters are only accessible after login. The analysis result could be downloaded as PDF (need login) or image. Finally, users can check analysis history after login.
Admin can check user account information and update LSTM model after Authentication.

Functional Descriptions:

Demand function	
Description	The function fits the most suitable demand curve according to uploaded data, providing live revenue calculation based on target price with basic statistics and result downloading provided.
Goals	Live revenue calculation, advice on pricing strategy
Actions	Display curve, update live revenue, store result in DB
Triggers	The user chose the function and uploaded data in correct format.
Information used	User data, pre-defined mathematical

	model
Information produced	Demand Curve, live revenue given target price, raw data statistics

RFM function	
Description	The function segments customers into seven groups of different value levels according to uploaded data, providing straightforward plots visualising those groups of which the label is the suggested action with basic statistics and result downloading provided.
Goals	Live customer group visualisation and interaction, advice on each group
Actions	Display interaction Treemap, store result in DB
Triggers	User chose the function and uploaded data in correct format.
Information used	User data, pre-defined aggregation algorithm
Information produced	Customer segment Treemap, Live information and suggestion as mouse pointer goes, raw data statistics

LSTM function	
Description	The registered user only function predicts short term (specific days input by user) business metric according to uploaded data, providing straightforward line plots visualising business metric of which the x-axis can be adjusted to allow users to view details of the curve with basic statistics and result downloading provided.
Goals	Custom number of days of prediction of custom business metrics
Actions	Display interactive line plot, store result in DB
Triggers	User already login, User chose the function and uploaded data in correct

	format
Information used	User data, pre-trained LSTM model.
Information produced	Customised business metric line plot, Live curve update as user changes visible x-axis range, raw data statistics

Holt-Winters function	
Description	The registered user only function predicts short/long term (specific days input by user) business metric according to uploaded data, providing straightforward line plots visualising business metric of which the x axis can be adjusted to allow users to view details of the curve with basic statistics and result downloading provided.
Goals	Custom number of days of prediction of custom business metrics
Actions	Display interactive line plot, store result in DB
Triggers	User already login, User chose the function and uploaded data in correct format
Information used	User data, Holt-Winters library function
Information produced	Customised business metric line plot, Live curve update as user changes visible x-axis range, raw data statistics

PDF Download function	
Description	This registered user only function fetches user analysis result and generates PDF report for users to download.
Goals	Allow users to download their analysis PDF report
Actions	Check Login status, fetch user analysis result, generate PDF report file response

Triggers	User already login, has valid analysis result and clicked 'download PDF report'.
Information used	User analysis data
Information produced	User analysis result PDF report

Upload function	
Description	This upload function receives user local file upload and stores the file at server local storage. In addition, it stores which function user chose and user login status.
Goals	To allow users to upload their local files to webserver
Actions	Store upload file at server local storage, which function user chose and user login status
Triggers	User chose valid upload file and clicked 'submit'.
Information used	User upload file
Information produced	

History function	
Description	This registered-user-only function fetches user analysis history actions in DB and displays them.
Goals	To allow users to check their history actions.
Actions	Display history action date, related file and actions
Triggers	User already login, clicked 'History'
Information used	User history action record in DB
Information produced	

Admin function	
Description	This admin only function displays user account information in DB and allows upgrading LSTM pre-trained model by uploading latest version to replace the previous one.
Goals	To allow admin to conduct user

	management and core service maintenance
Actions	Display user accounts in DB, receive upload pre-trained LSTM model and replace the old one in local server storage.
Triggers	Valid admin login
Information used	User account record in DB
Information produced	

Authenticate function	
Description	The utility function recognises user type, changes user login status and updates cookies for other modules' reference.
Goals	Authentication and login status update of 2 different user groups
Actions	Change UI (guest/user/admin dashboard), update user login status.
Triggers	User clicked 'login/logout'/Login attempt with valid admin credential
Information used	User/admin account data in DB
Information produced	Updated login status in cookies

Data-flow diagram:

The Data-flow diagram below (see Figure 3 in the next page) illustrates the data flow between different processes of the system based on actions of both normal user and admin user. All analysis processes start with user uploading dataset files. Once the result is generated, it could be downloaded in two forms and will be archived into history. During maintenance, admin user could check user account information and update LSTM model after authentication.

3.2 Algorithms

Frontend

We use JSON data structure to carry data from backend to frontend where a customised algorithm fits the data into the architecture of charts.

Backend

Demand curve fitting and LSTM

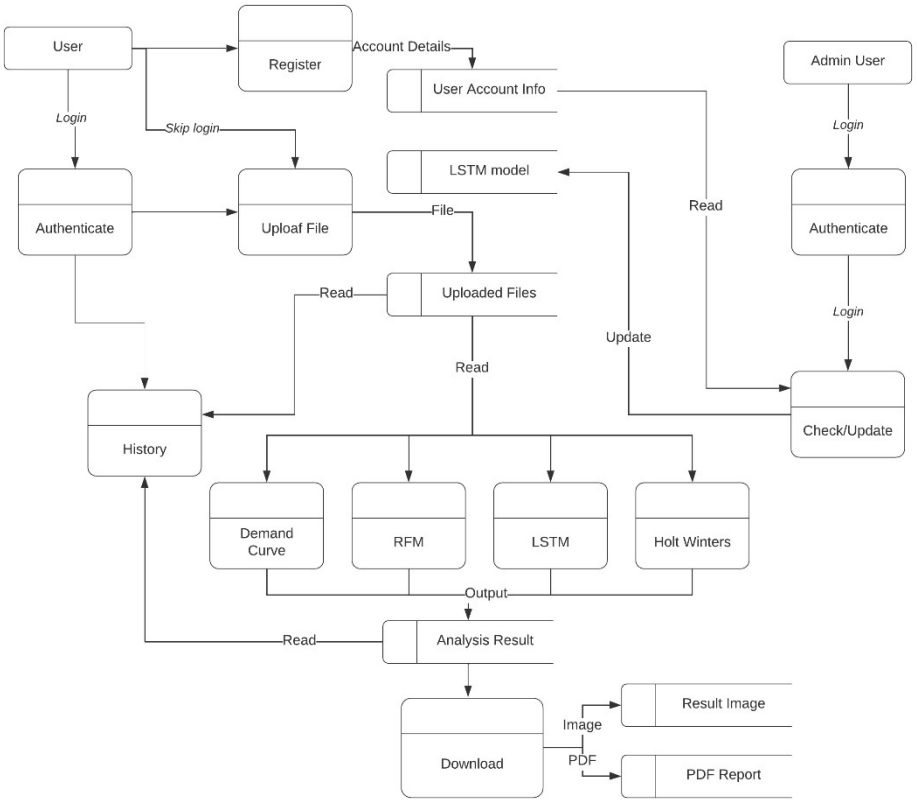


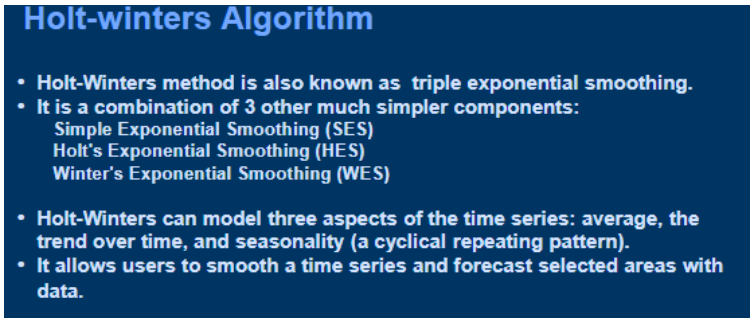
Figure 3: System data-flow diagram

Demand:	LSTM:
Data structure: Pandas DataFrame, Numpy ndarray	Data structure: Pandas DataFrame, Numpy ndarray, python list
Pseudocode **Objective(x, alpha, beta):** return exp(alpha * log(x) + beta) **Main():** x, y = read_csv() param, param_cov = curve_fit(Objective, x, y) optimal_alpha = param[0] optimal_beta = param[1] ans = exp(optimal_alpha * log(x) + optimal_beta) plot(original_data) plot(fitted_curve)	**LSTM Model Algorithm** We predict a "future value" with "historical data" and assume it as "historical" for next round. It means to predict & assume iteratively "towards the future". Pseudocode Input: Upload Data, Future (int) create an empty List named Prediction (List) load pre-trained model For k=1 to future do: predict next value with the last row of Uploaded Data add the predicted value to Prediction (List) append predicted value to the end of Upload Data Output: Prediction (List)

RFM: (Yuan, 2021)
Data structure: Pandas DataFrame

Holt-Winters (Makridakis, Wheelwright and Hyndman, 1997):
Data structure: Pandas DataFrame, Numpy ndarray
For time series prediction, statistical models are reported to have played a significant role while currently machine learning models are employed more

frequently. Given the already implemented deep-learning LSTM model requiring significant computational power, a complementary statistical model is selected to make the prediction process more efficient and informative. After reviewing literature on time series prediction (Kurniasih et al., 2018), we choose Holt-Winters model, which is an extension of Holt's linear model. This model takes level, trends and seasonality into account and could perform short-, medium- and long-term time series prediction (Rahman and Ahmar, 2017).



4 Interface

4.1 System interface

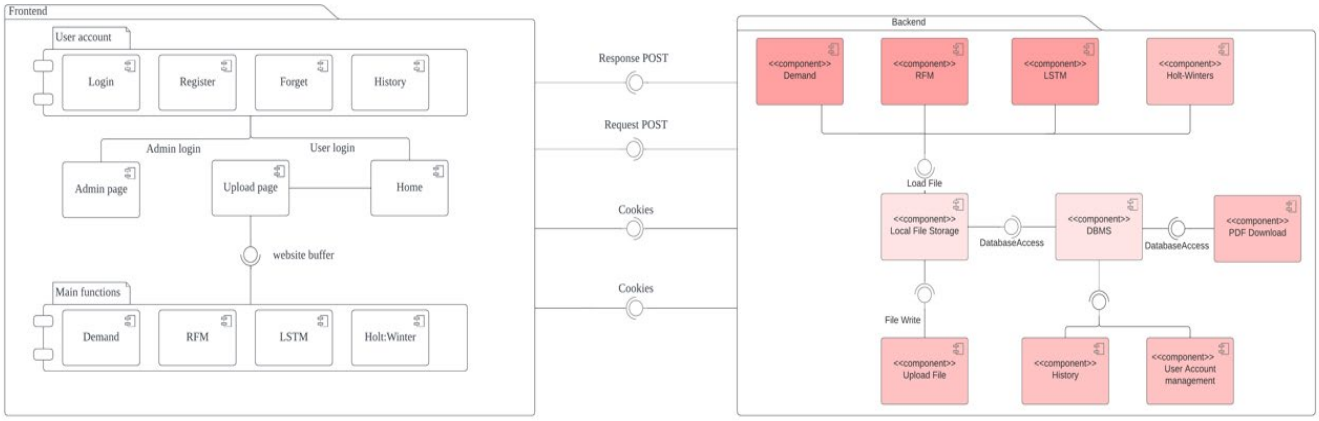


Figure 4: System interface

Figure 4 describes the internal and external interfaces between frontend and backend and their system component. External interfaces including Request POST and cookies, frontend internal interfaces is browser buffer and backend is function API call. Frontend system includes user account, core function display, admin and homepage components while backend system consists of 4 core functional components, four utility components, local file storage and DBMS.

4.1.1 System architecture interface

Frontend:
• During user account related operations and file upload, firstly data is packed into the format FormData, then use Axios to transmit data to and from backend.
• Use cookies to store intermediate data of smaller size, for example, user login status.
• Encode parameters inside URL to transmit data between different web pages.
• Use local browser buffer to store a large number of data points (at most 5 MB).

Backend:
Core functions, History, Download PDF:
• Check login status stored in cookies and the upload file path corresponding to each username
• All results of Demand, RFM, LSTM are returned in JSON format via Request POST

4.1.2 Algorithm interface

LSTM prediction

In this algorithm, there are four parameters which act as an interface.
"future": the range of forecast input by users. (e.g. 30 days)
"model": pre-trained model
"DataFrame": user upload file
"look_back": a hyper parameter related to the model (not exposed to user)

From users' view, they first need to upload their data file which will be passed to the algorithm via parameter "DataFrame". Then they need to input how many days they want to predict. This is passed to the algorithm via parameter "future".

Holt-winters prediction

The interface definition of Holt-Winters algorithm is similar to that of LSTM:
"future": how many days they want to predict.
"DataFrame": user upload file

4.2 User Interface

In this section, we will briefly explain our user interface design.

Our prototype UI and planned navigation flow are shown in Figure 5 (see the next page).
The final decision for UI/UX design is shown below in Figure 6 (see the next page).

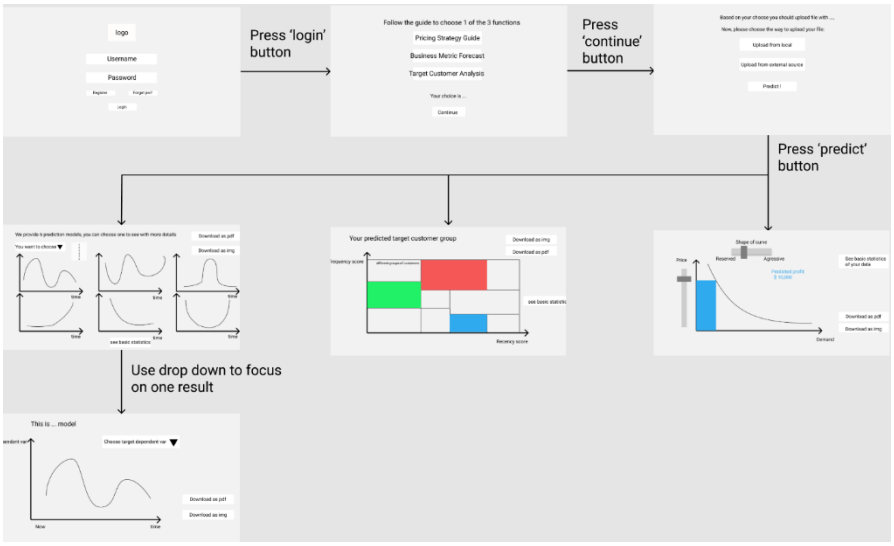


Figure 5: Prototype UI and planned navigation flow

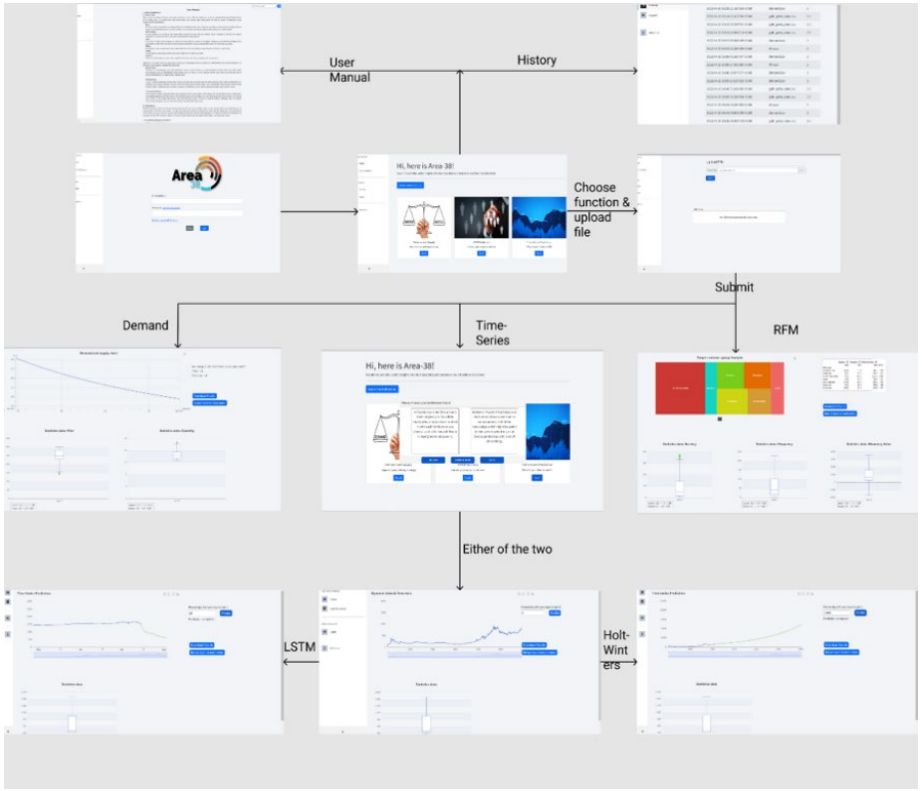


Figure 6: Final UI/UX design

According to Figure 7, there can be three major logic flows. The first one is user account and admin related state changes. In addition, separate dashboards for user and guest would grant access to different set of functionalities which all start with file upload. Result presentation and related downloading is encapsulated in each functionality state.

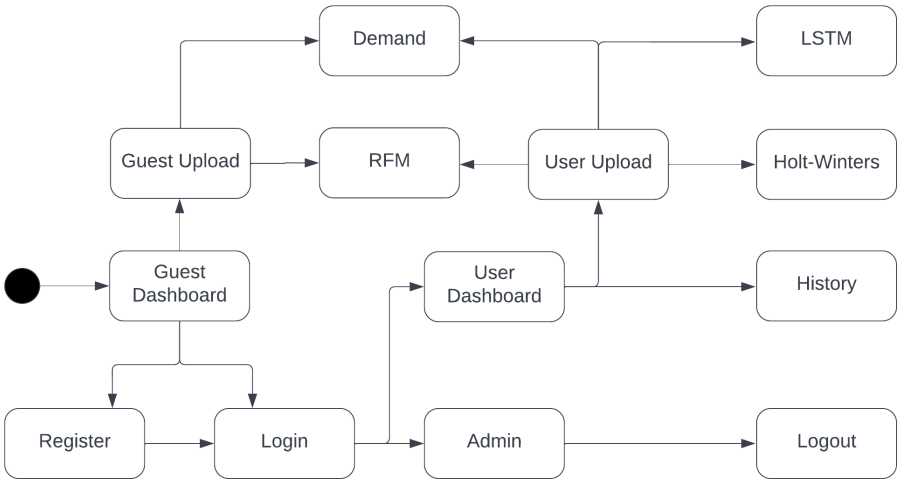


Figure 7: System state diagram

5 Design of Evaluation Process (Based on COMP201 definition)

5.1 Verification:

Software inspections: Group members of the same sub-team would inspect others' code on GitHub.

Dynamic verification: unit/regression test is done by the same developer of the function:

Frontend unit test:
(1). Test the display of pages directly; (2). Use the website console to print information about the data structure and functional logic. (3). Use pseudo backend to send requests to and receive the response data from. (4). Compile demo data sets to fit into different charts to see if displayed correctly.
Backend unit test:
(1) logic check (2) intermediate data check (3) processed result check (4) data type check (5) return check (6) runtime error check
Integration test: by front & backend: Mainly use browser console to print key information.
In each milestone, the integration test will check if the system satisfies functional and non-functional requirements. For example, user account service: the user information should be successfully sent to backend through form; the information regulation will be preliminarily checked by frontend. Main functions: the data computed by backend should be returned to frontend and

displayed on charts in a correct manner; users can interact with this system using designated buttons.

When it comes to **system testing**, our testing team intend to divide our testing process into three parts which are functional testing, stress testing and compatibility testing. In the initial stage of functional testing, we will use the smoke testing to quickly have verification whether the basic functionalities of the website are defective. Subsequently, we are going to use black-box testing method to further check some extra functionalities of our website without understanding its internal structure and processing procedure. For stress testing, we will use boundary value method to test the stability of website when it is beyond normal operational capacity. Finally, with the assistance of the automatic tool which called PowerMapper, we will test the compatibility of our website with different browsers like Firefox, Google Chrome, Internet Explorer etc. to ensure our website works under different configurations.

5.2 Validation: We plan to invite 5-10 beta users to fill the UEQ-S (User Experience Questionnaire Short version) (Schrepp, Thomaschewski and Hinderks, 2017) to evaluate if our project meets users' requirements.

5.3 Evaluation for our project highlight: LSTM model

Our evaluation focuses on model and algorithm. All these evaluations mean to compare and help find a relatively suitable time-series model.

Model:
Use simple cross-validation with 20% test set and 80% train set.
Use K-fold Cross-Validation (noted as K-CV) with 1:9 ratio (Refaeilzadeh, Tang and Liu, 2016).
Compute and print loss and val_loss, figure out the trend as training progresses (baeldung, 2022).

Algorithm:
Compute the time complexity.
Make predictions with test set and compare the result with the ground truth inside the same set.
According to a series of evaluations, the loss curve (Figure 8) and the computational time complexity are relatively acceptable. Thus, we decide to employ it.

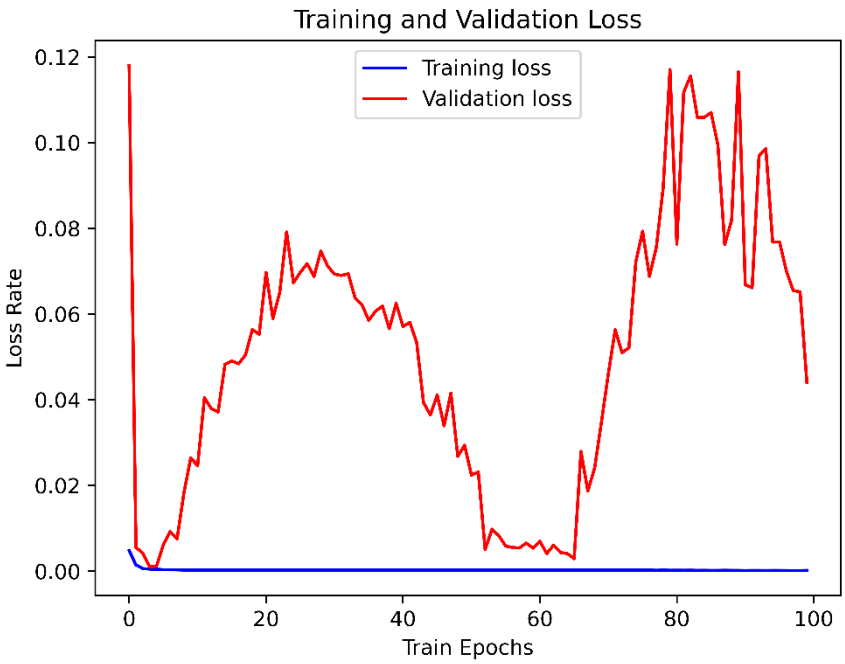


Figure 8: Training and validation loss of LSTM model

5.4 Evaluation of Holt-Winters Model

We plan to implement the Holt-Winters model in our project. In that case, we designed an experiment to evaluate the model's performance.

To evaluate our Holt-Winters Model, we split the dataset into 90% training and 10% test set. The proportion is different from the classic since, in practice, the Holt-Winters model is usually used for short-term prediction (while it is suitable for long-term prediction as well). We follow the previous work (Colin Cameron and Windmeijer, 1997) and choose R-Squared (R^2) as the evaluation criteria. R-Square is a statical measure that reflects the goodness-of-fit for regression results. Our model achieved $R^2 = 0.2099$ (higher is better, maximum is 1, and can be negative). We also visualise the forecasting results in Fig 1. The experiment's results meet our expectations, and we decide to choose the Holt-Winters model as our statistical model.

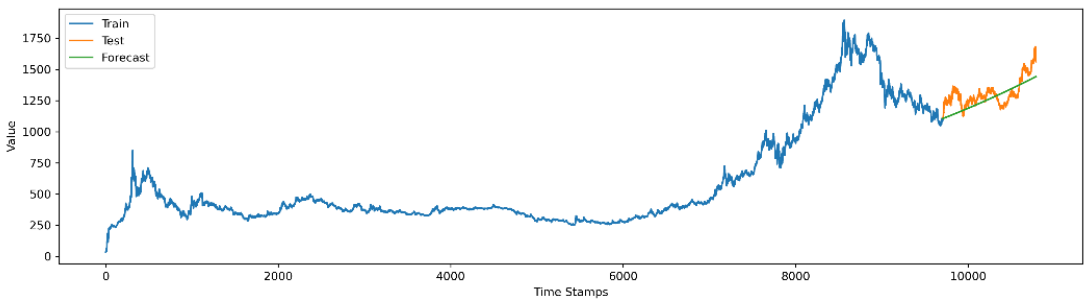


Figure 9: Prediction results of Holt-Winters model on the Gold-Price dataset

6 Reference

ADAM, A. (2021). *Starter: Digital currency - Time series SAR & USD*. [online] kaggle.com. Available at: https://www.kaggle.com/code/ahmedadam415/starter-digital-currency-time-series-sar-usd [Accessed 10 Apr. 2022].

baeldung (2022). *Training and Validation Loss in Deep Learning | Baeldung on Computer Science*. [online] www.baeldung.com. Available at: https://www.baeldung.com/cs/training-validation-loss-deep-learning [Accessed 9 May 2022].

CleverTap. (2018). *RFM Analysis for Customer Segmentation | CleverTap*. [online] Available at: https://clevertap.com/blog/rfm-analysis/.

Colin Cameron, A. and Windmeijer, F.A.G. (1997). An R-squared measure of goodness of fit for some common nonlinear regression models. *Journal of Econometrics*, [online] 77(2), pp.329–342. doi:10.1016/S0304-4076(96)01818-0.

echarts.apache.org. (n.d.). *Apache ECharts*. [online] Available at: https://echarts.apache.org [Accessed 9 May 2022].

FABIENDANIEL (2019). *Customer Segmentation*. [online] kaggle.com. Available at: https://www.kaggle.com/code/fabiendaniel/customer-segmentation/data [Accessed 4 Apr. 2022].

GeeksforGeeks. (2019). *SciPy | Curve Fitting*. [online] Available at: https://www.geeksforgeeks.org/scipy-curve-fitting/ [Accessed 9 May 2022].

Hamami, F. and Dahlan, I.A. (2020). Univariate Time Series Data Forecasting of Air Pollution using LSTM Neural Network. In: *IEEE Xplore*. [online] pp.1–5. doi:10.1109/ICADEIS49811.2020.9277393.

insightr (2018). *Different demand functions and optimal price estimation in R | R-*

bloggers. [online] R-bloggers. Available at: https://www.r-bloggers.com/2018/06/different-demand-functions-and-optimal-price-estimation-in-r/ [Accessed 9 May 2022].

Kurniasih, N., Ahmar, A.S., Hidayat, D.R., Agustin, H. and Rizal, E. (2018). Forecasting Infant Mortality Rate for China: A Comparison Between α-Sutte Indicator, ARIMA, and Holt-Winters. *Journal of Physics: Conference Series*, 1028(012195), p.012195. doi:10.1088/1742-6596/1028/1/012195.

Li, S. (2018). *Machine-Learning-with-Python*. [online] GitHub. Available at: https://github.com/susanli2016/Machine-Learning-with-Python [Accessed 2 Apr. 2022].

Lima, A. (2021). *Bitcoin Price Prediction Using Recurrent Neural Networks and LSTM*. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/05/bitcoin-price-prediction-using-recurrent-neural-networks-and-lstm/ [Accessed 10 Apr. 2022].

Makridakis, S., Wheelwright, S. and Hyndman, R.J. (1997). *Forecasting: Methods and Applications, 3rd Ed*. [online] *research.monash.edu*. John Wiley & Sons. Available at: https://research.monash.edu/en/publications/forecasting-methods-and-applications-3rd-ed [Accessed 15 Apr. 2022].

MÖBIUS (2021). *Learn Time Series Forecasting From Gold Price*. [online] www.kaggle.com. Available at: https://www.kaggle.com/datasets/arashnic/learn-time-series-forecasting-from-gold-price?select=gold_price_data.csv [Accessed 12 Apr. 2022].

Rahman, A. and Ahmar, A.S. (2017). Forecasting of primary energy consumption data in the United States: A comparison between ARIMA and Holter-Winters models. *AIP*

Conference Proceedings, [online] 1885(020163). doi:10.1063/1.5002357.

Refaeilzadeh, P., Tang, L. and Liu, H. (2016). Encyclopedia of Database Systems. In: *Encyclopedia of Database Systems*. [online] New York: Springer. Available at: https://link.springer.com/referenceworkentry/10.1007/978-1-4899-7993-3_565-2 [Accessed Apr. 2022].

Rizal, A.A., Soraya, S. and Tajuddin, M. (2019). Sequence to sequence analysis with long short term memory for tourist arrivals prediction. In: *Journal of Physics: Conference Series*. [online] p.012024. doi:10.1088/1742-6596/1211/1/012024.

Schrepp, M., ThomaschewskiJ. and Hinderks, A. (2017). Design and Evaluation of a Short Version of the User Experience Questionnaire (UEQ-S). *International Journal of Interactive Multimedia and Artificial Intelligence*, [online] 4(Regular Issue). Available at: https://www.ijimai.org/journal/bibcite/reference/2634 [Accessed 9 May 2022].

Selvin, S., Vinayakumar, R., Gopalakrishnan, E.A., Menon, V.K. and Soman, K.P. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. In: *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. doi:10.1109/icacci.2017.8126078.

v3.cn.vuejs.org. (n.d.). *Vue.js*. [online] Available at: https://v3.cn.vuejs.org [Accessed 9 May 2022].

Yuan, Y. (2021). *Recency, Frequency, Monetary Model with Python — and how Sephora uses it to optimise their Google…*. [online] Medium. Available at: https://towardsdatascience.com/recency-frequency-monetary-model-with-python-and-how-sephora-uses-it-to-optimize-their-google-d6a0707c5f17 [Accessed 9 May 2022].