Comprehensive Projects

By Maggie Wu

Agenda

01 Text Analysis of Product Reviews on Twitter with Python

02 Automatic Workflow & Abnormal Analysis

03 Database Design & its Application Project with Oracle Apex

04 Fianncing Analysis with R

05 Supplychain Analysis with Python using Linear Programming

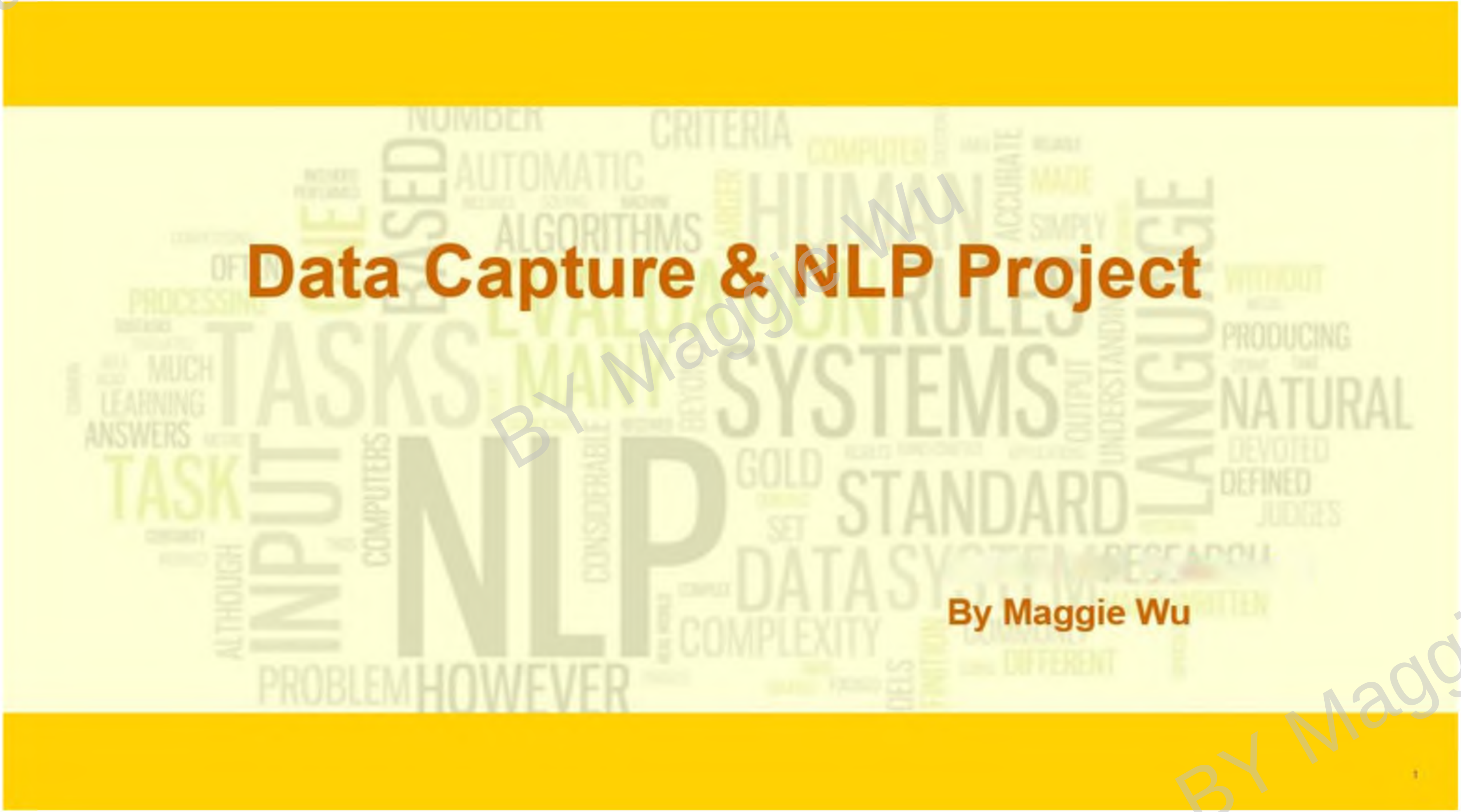
/01

Text Analysis of Product Reviews on Twitter with Python

BY Maggie Wu

BY Maggie Wu

Text analysis of product reviews on Twitter with Python：



Text analysis of product reviews on Twitter with Python

1. Introduction to the Project

This dataset comes from Twitter by using tweepy to capture 2000 lines of Tesla Model Y reviews with text cleaning processing.

Background:

Tesla is an electric vehicle and energy company in USA, with a market value of US $210billion, producing and selling electric vehicles; Tesla's vision is to accelerate the global transformation to sustainable energy. Tesla's market share in China is also growing. Tesla can often be seen on the road in recent two years. It has also driven the rapid development of the domestic electric vehicle industry and environmental sustainable development.

In this project, I analyzed the twitter reviews on Tesla Model Y to check for its popularity among English language countries, peoples' sentiments on it, interest in it and main topics they are talking about it.

Datasets:

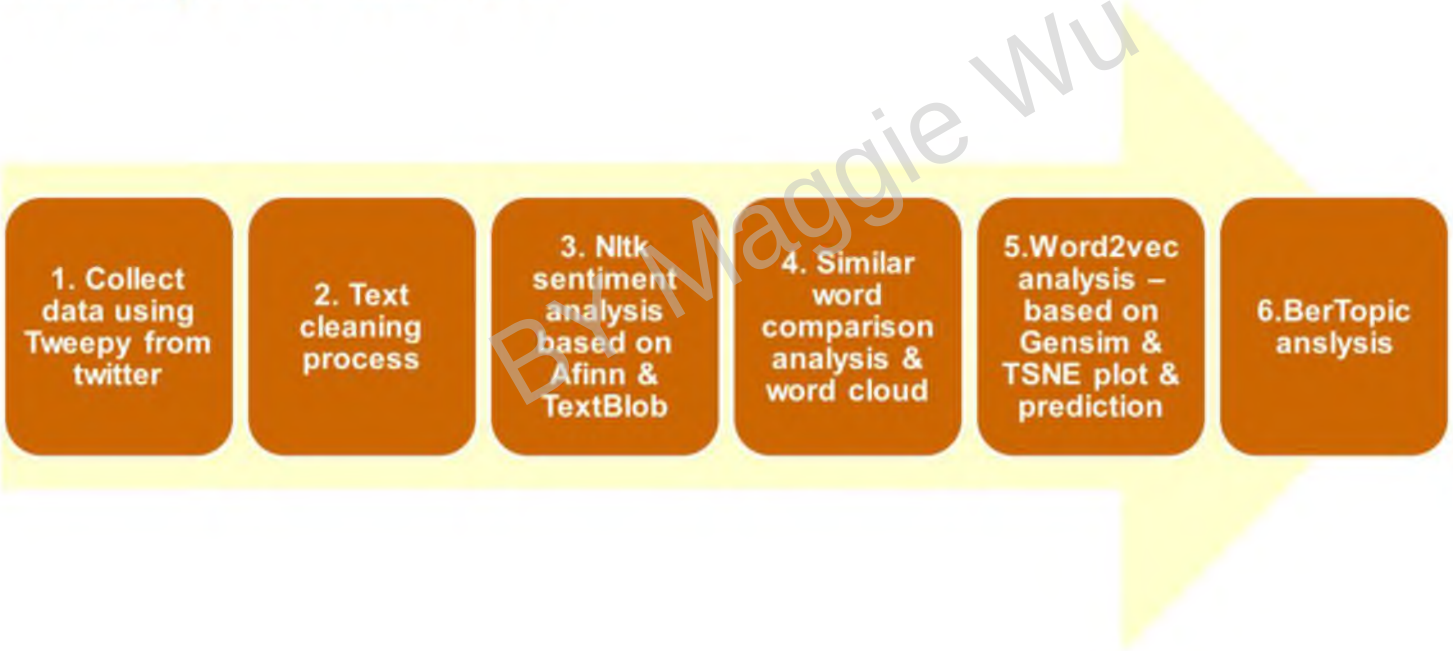
The dataset are mainly about peoples' 2000 recent reviews on Tesla Model Y on in English language countries.

There are 7 dimension as follows:

created_at, text, retweet_count, favorite_count, screenname, location, sentiment, cleantext

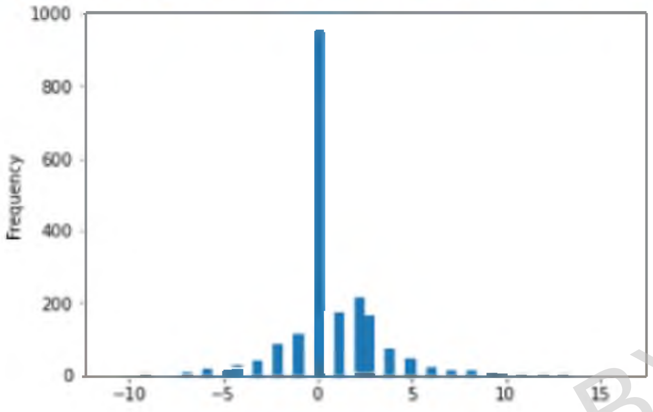
Text analysis of product reviews on Twitter with Python

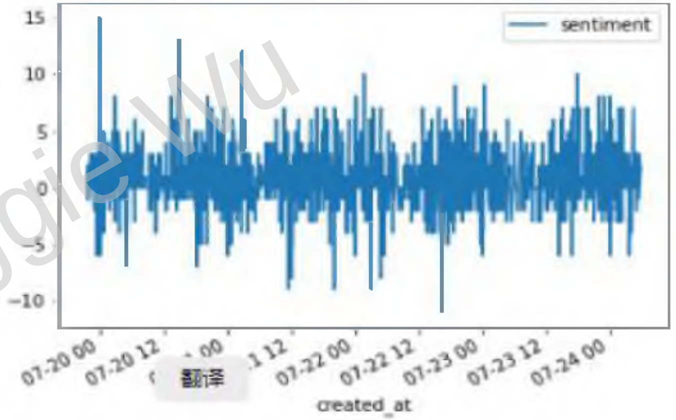
2. Data Manipulation Process



Text analysis of product reviews on Twitter with Python

3. Exploring the Unstructured Data – sentiment analysis





Sentiment analysis based on AFINN:

We can find out most sentiment is distributed nearby 0, the positive sentiment is not too high, so as the negative sentiment is not too low. People's sentiment is neutral.

Analyze the sentiment according to time line, the sentiment fluctuation is evenly distributed.

Sentiment analysis based on TextBlob:

Using Tweetblob we can also get the sentiment polarity (-1 ~ +1) is only 0.14 (not high) and objectivity is 0.5.

```
tweetblob.sentiment
```

```
Sentiment(polarity=0.1416118580308575, subjectivity=0.5031542267732269)
```

Text analysis of product reviews on Twitter with Python

3. Exploring the Unstructured Data - common word analysis

```
stopwords = nltk.corpus.stopwords.words('english')
stopwords.extend(['https','’ ','i','the','…','s','it','get','one','my','car',
wordfreqs = nltk.probability.FreqDist(w.lower() for w in tweetblob.words if w
mostcommon = wordfreqs.most_common(30)
mostcommon
```

```
[('tesla', 470),
 ('like', 139),
 ('new', 120),
 ('2022', 78),
 ('would', 74),
 ('battery', 74),
 ('first', 73),
 ('know', 72),
 ('4680', 71),
 ('performance', 66),
 ('giga', 60),
 ('range', 59),
 ('got', 58),
 ('love', 54),
 ('want', 54),
 ('berlin', 52),
 ('think', 52),
 ('really', 50),
 ('made', 49),
 ('production', 47),
```

Using Nltk package to count the most common words

- According to left side, we can see the most common words discussed by people, I think every word indicates one hot topic. We can find people concern about the battery, performance and production. People's sentiment is like, love, want, etc.
- Besides, if someone plan to buy a product and want to be familiar with it, they can use this way to explore:
- For example, the word 4680, when I searched for news about it, I learned it's a kind of battery which has production bottleneck which will lead to production shortage.
- The word giga and berlin, it indicates a Tesla factory Giga Berlin where a Tesla can be manufactured in 45 seconds.

Text analysis of product reviews on Twitter with Python

3. Exploring the Unstructured Data - common word analysis

```
#ngrams = nltk.bigrams(tweetblob.words)
ngrams = nltk.bigrams(w.lower() for w in tweetblob.words if w n


ngramfreqs = nltk.probability.FreqDist(ngrams)
mostcommon = ngramfreqs.most_common(20)
mostcommon
```

```
[(('giga', 'berlin'), 35),
 (('dual', 'motor'), 33),
 (('4680', 'battery'), 28),
 (('tesla', 'dual'), 28),
 (('2022', 'tesla'), 25),
 (('long', 'range'), 18),
 (('battery', 'pack'), 17),
 (('order', 'taiwan'), 17),
 (('tesla', 'giga'), 17),
 (('giga', 'texas'), 17),
 (('chevrolet', 'blazer'), 16),
 (('wan', 'na'), 15),
 (('gon', 'na'), 15),
 (('pay', 'per'), 15),
 (('per', 'minute'), 15),
 (('mustang', 'mach'), 15),
```

Using Nltk package to count the most common 2 related words:

- The left side analysis confirmed my finding on last page. we can find out giga is related to berlin, 4680 is related to battery.
- Other most common 2 related words bring about more new information:
- Per & minute: indicates Tesla's production in high speed;
- Chevolet & blazer: Chevrolet Blazer EV which is the second best seller in USA SUV markets, it's Tesla's strong competitor.

BY Maggie Wu

BY Maggie Wu

Text analysis of product reviews on Twitter with Python

3. Exploring the Unstructured Data – location using word cloud

Word cloud analysis for twitter location:

- In consideration the location is mixed with cities and countries, so I use word cloud to show the distribution.
- Since we use English speaking countries to collect the twitter reviews, so we can see USA, Canada, New York, Australia. However, we can also find out Austria and India people is very interested in Tesla.

Text analysis of product reviews on Twitter with Python

3. Exploring the Unstructured Data - similar word analysis

Similar words analysis compared by using different analysis package:

Nltk package:

```
text2 = nltk.Text(w.lower() for w in tokens if w not in stopwords)
text2.similar('battery')
```

```
the performance structural from
```

This analysis is very impressive. Two package show different results:

- When I asked Nltk about battery, they let me pay attention to its performance.
- When I asked Word2vec about battery, they imply 2023 and hope. I think maybe the battery problem can be solved in 2023.

Word2vec package:

```
tesla_w2v.wv.most_similar('battery')
```

```
[('2023', 0.3212713599205017),
 ('hope', 0.2639414072036743),
 ('driven', 0.2599856853485107 4),
 ('used', 0.2574970722198486 3),
 ('musk', 0.2332437187433242 8),
 ('ordered', 0.2327056229114532 5),
 ('yet', 0.2218781858682632 4),
 ('every', 0.2214201986789703 4),
 ('tonight', 0.2170273959636688 2),
 ('buy', 0.2025878280401229 9)]
```

Text analysis of product reviews on Twitter with Python

3. Exploring the Unstructured Data - similar word analysis

Similar words compared by using different analysis package:

Nltk package:

```
text = nltk.Text(w.lower() for w in tweetblob.words if w not in stopwords)
text.similar('berlin')
```

```
shanghai perfect differently
```

- When I asked Nltk about Berlin, they show its competitor Shanghai factory.
- When I asked Word2vec about Berlin, they imply carbon, performance and industrialization. In above mentioned analysis, we have known Giga Berlin's output is 45s/p. Now we got more information that Giga Berlin has adopted more solar energy to reduce carbon emission. Besides, Giga Berlin is trying to reduce the battery cost.

Word2vec package:

```
tesla_w2v.wv.most_similar('berlin') ##2
```

```
[('carbon', 0.3022310733795166),
 ('make', 0.27051132917404175),
 ('said', 0.2607069611549377 4),
 ('different', 0.25569725036621094),
 ('coming', 0.22725674510002136),
 ('high', 0.22516673803329468),
 ('making', 0.22131550312042236),
 ('save', 0.21456027030944824),
 ('performance', 0.209019273519516),
 ('industrialization', 0.20726546645164490)]
```

3. Exploring the Unstructured Data - similar word analysis

Similar words compared by using different analysis package:

Nltk package:

```
[72]: text = nltk.Text(w.lower() for w in tweetblob.words if w not in stopwords)
      text.similar('problem')
```

- When I asked Nltk about problem, they gave no answer.
- When I asked Word2vec about problem, they show skype, this answer is very interesting. I checked for related news, and found out that Tesla cars will soon support video conferencing (e.g. Skype, and there is a rear view camera in Tesla, which may be used as detecting driver distraction when Autopilot is turned on in future to reduce accidents. Besides, it also mentioned test, which is important for reducing problems.

Word2vec package:

```
tesla_w2v.wv.most_similar('problem')
```

```
[('test', 0.9976677298545837),
 ('made', 0.9976261854171753),
 ('car', 0.9975498914718628),
 ('skype', 0.9975183010101318),
 ('first', 0.9975097179412842),
 ('days', 0.997495174407959),
 ('two', 0.9974789619445801),
 ('model', 0.997477650642395),
 ('really', 0.9974758625030518),
 ('would', 0.9974592924118042)]
```

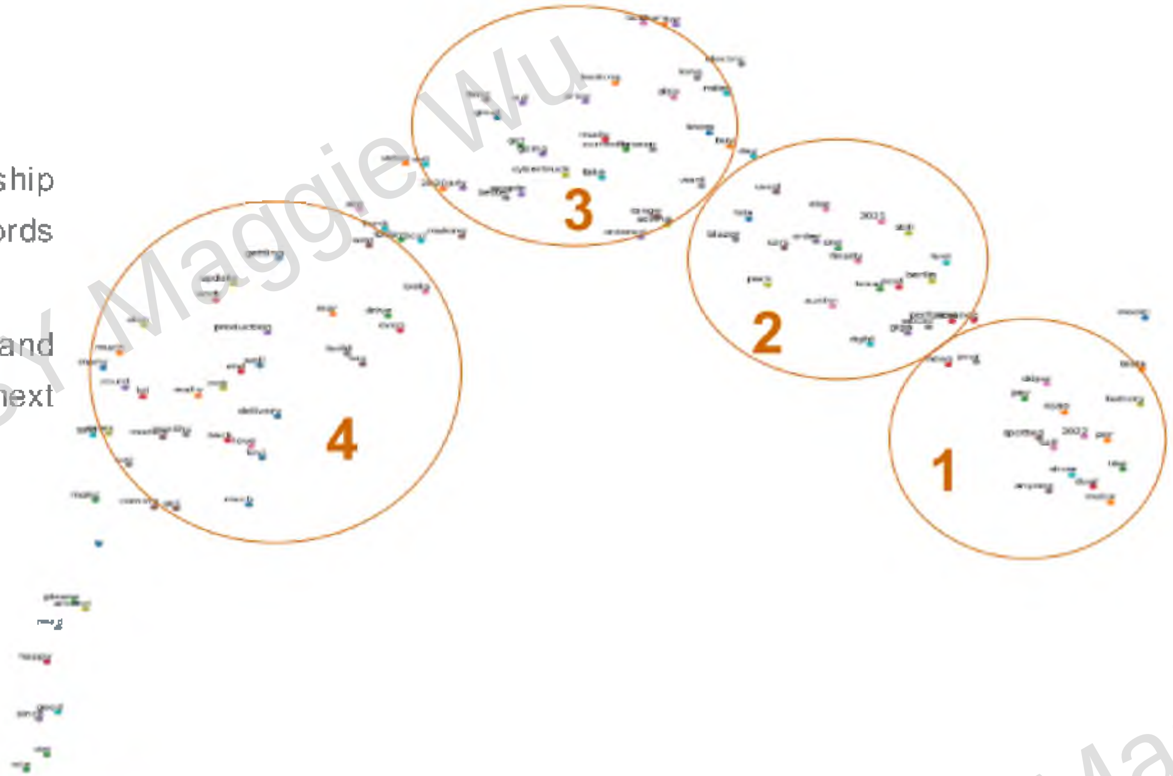
```
: tweetblob.words.count('problem')
```

```
: 12
```

3. Exploring the Unstructured Data –TSNE plot

TSNE Plot Analysis:

- I use TSNE plot to show the relationship between each words (I only show the words which are counted more than 50 times)
- Here I plan to divide them into 4 groups, and made detailed analysis one by one in next pages.



BY Maggie Wu

3. Exploring the Unstructured Data –TSNE plot

TSNE Plot Analysis:

Group 1

In this group we can see motor, ddpai, battery, 4680, dual, etc. So I think this group is most related to vehicle components and their structure.

Text analysis of product reviews on Twitter with Python

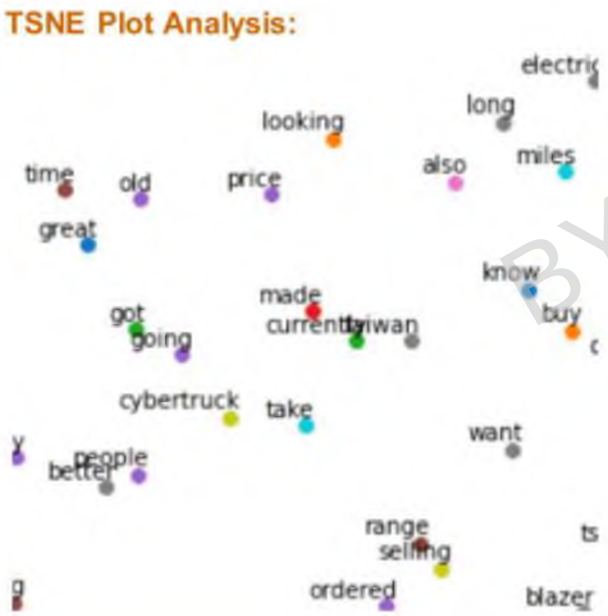
3. Exploring the Unstructured Data –TSNE plot

TSNE Plot Analysis:

Group 2

In this group we can see blazer (competitor's electric car type), texas berlin, Austin, giga, etc. So I think this group is most related to Tesla's different manufacturing location and its competitors.

3. Exploring the Unstructured Data –TSNE plot



Group 3

In this group we can see looking, miles, long, price, electric, time, cybertruck, range, selling, etc. So I think this group is most related to people's cost performance requirement towards Tesla's.

Cybertruck

3. Exploring the Unstructured Data –TSNE plot

TSNE Plot Analysis:

Group 4

In this group we can see well, really, love, many, much, production, delivery, months, test, match, etc. So I think this group is most related to Tesla's production and delivery capacity, and people's sentiment (purchasing enthusiasm) and maybe concern about test.

4. Summary and Recommendation

- According to sentiment analysis, out of my expectation, people are not very excited about Tesla, they didn't discuss actively every day. People are calm about it. I think if we can compare with its competitor Chevrolet Blazer EV, the sentiment result would be more clear.

- When I made common words and TSNE analysis, I find it's very helpful for a potential customer, in this way:

  1) The potential customer can try to know its competitor as alternative;
  2) Pay more attention to its battery problem before purchasing;
  3) Giga Berlin is trying to reduce the battery cost, they maybe can wait till 2023 to buy;
  4) There may be late delivery problem;
  5) They need to take consideration to cost performance.

- According to my analysis, I suggest the potential customer it would be better to wait and buy later.

/02

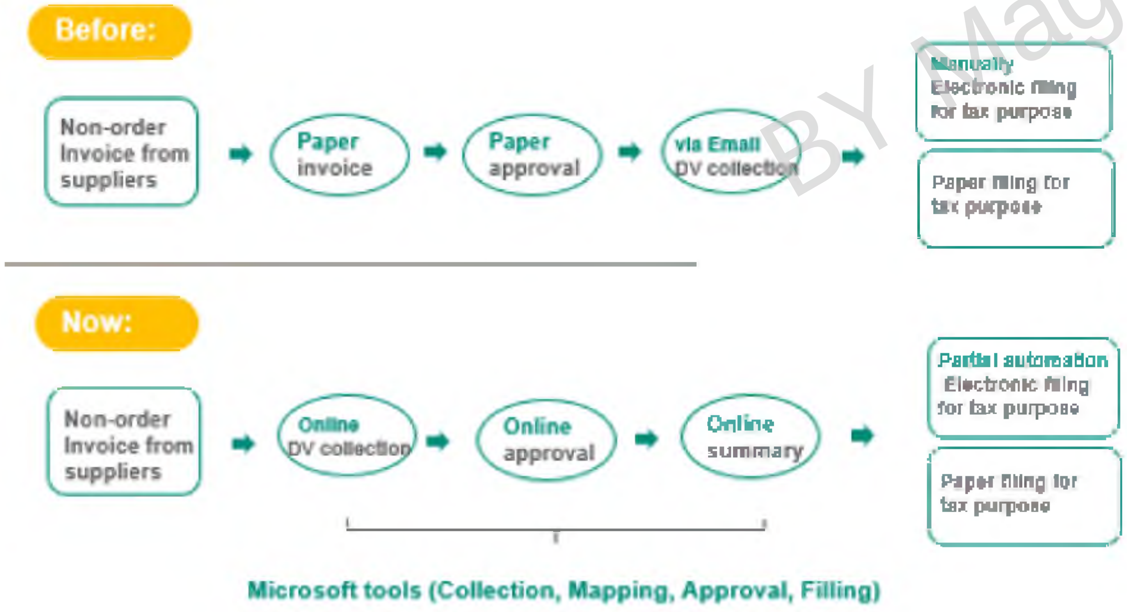
Automatic Workflow & Abnormal Analysis

Automatic workflow and abnormal analysis

1. Project overview

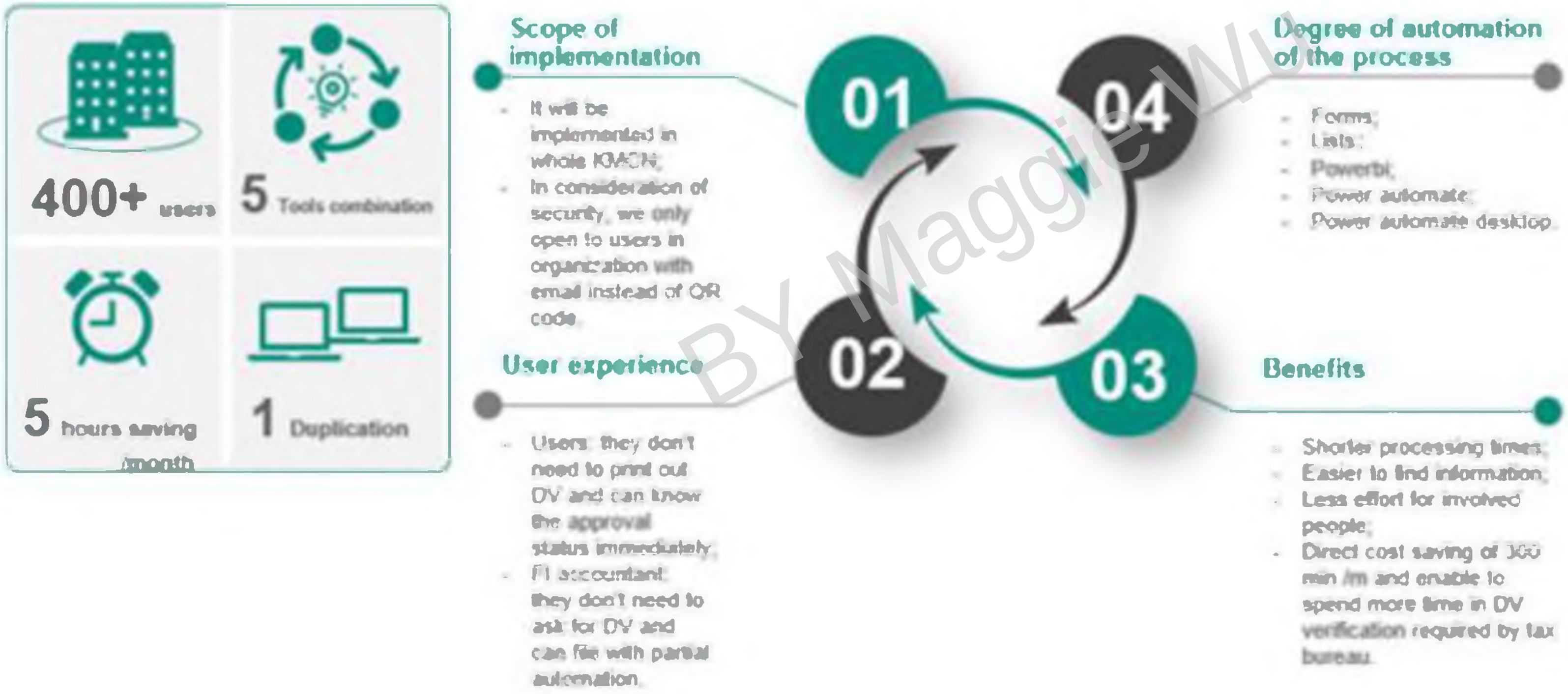
Since China tax bureau began to encourage the usage of digital invoice (**DV**). Now KMCN receive DV from suppliers who are qualified to issue DV, soon the authority plan to realize the complete DV all over China.

During the current transition period, KMCN need to file DV electronically, and meanwhile file the paper invoice of them, which leads to double workload then before. In order to improve work efficiency, we use Microsoft tools to optimize the workflow as follows.



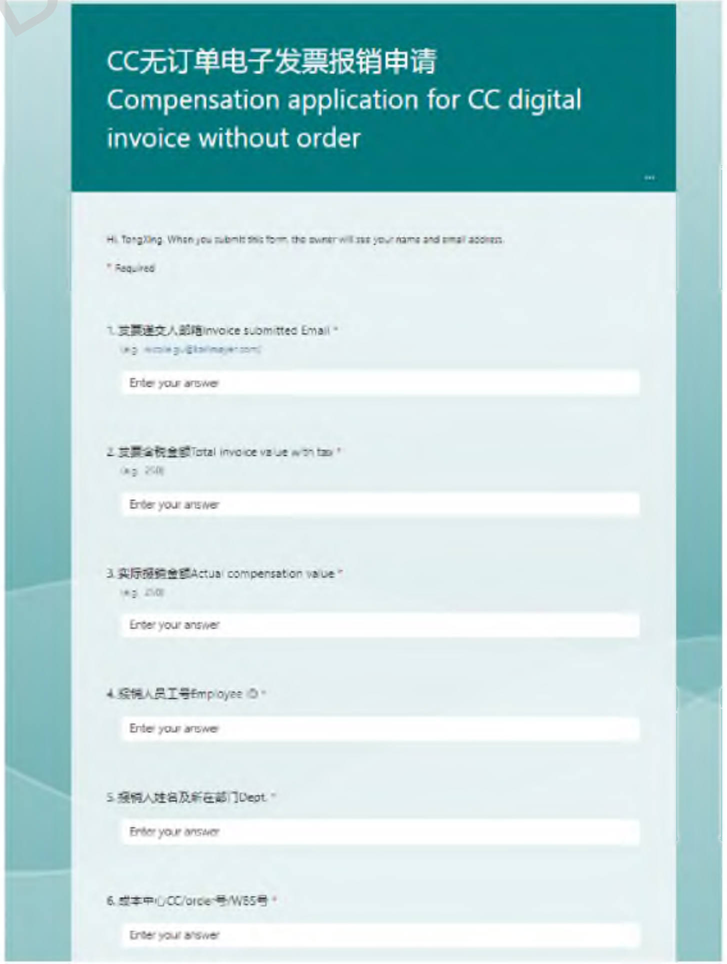
Automatic workflow and abnormal analysis

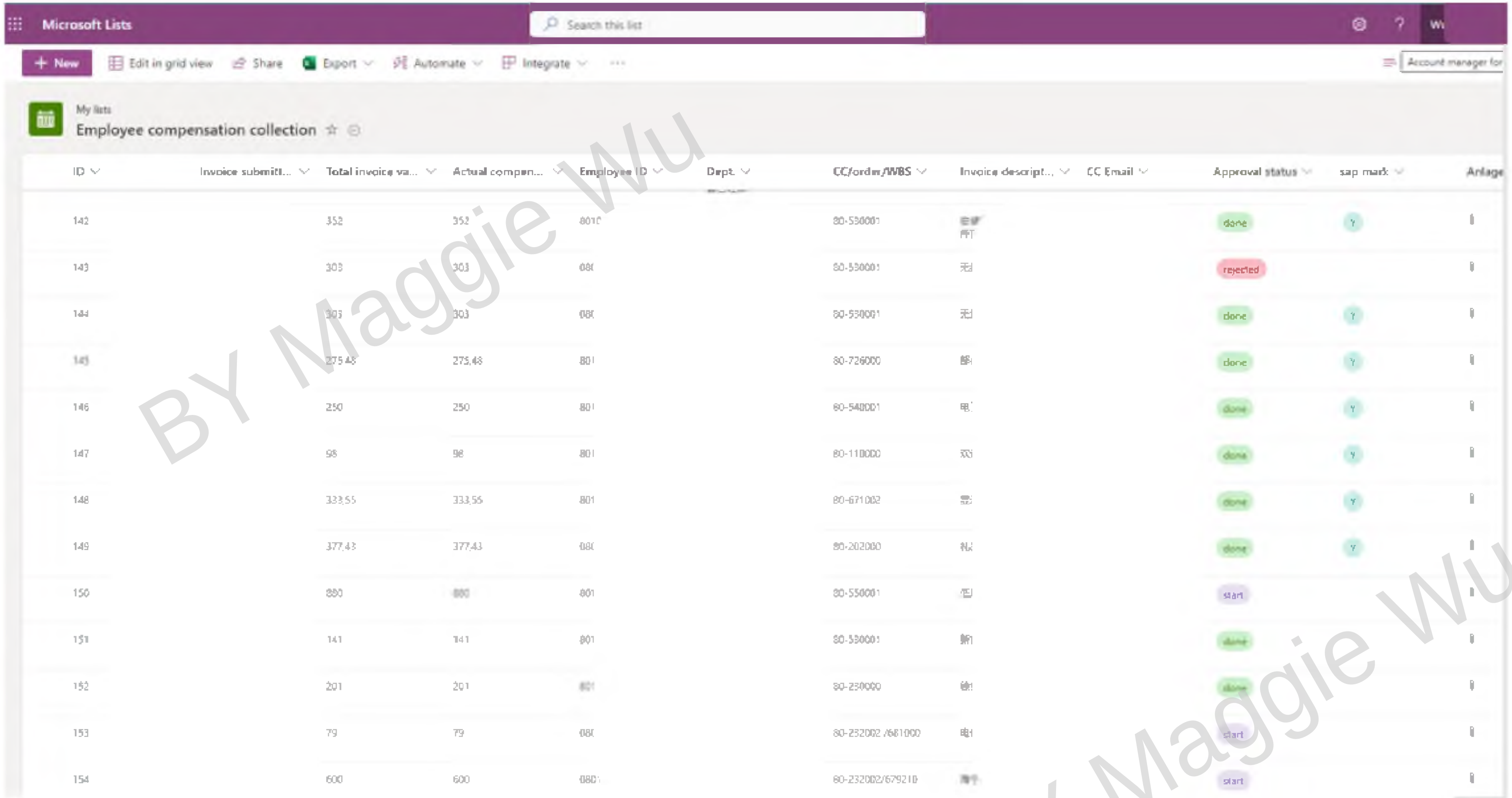
2. Project details



Automatic workflow and abnormal analysis

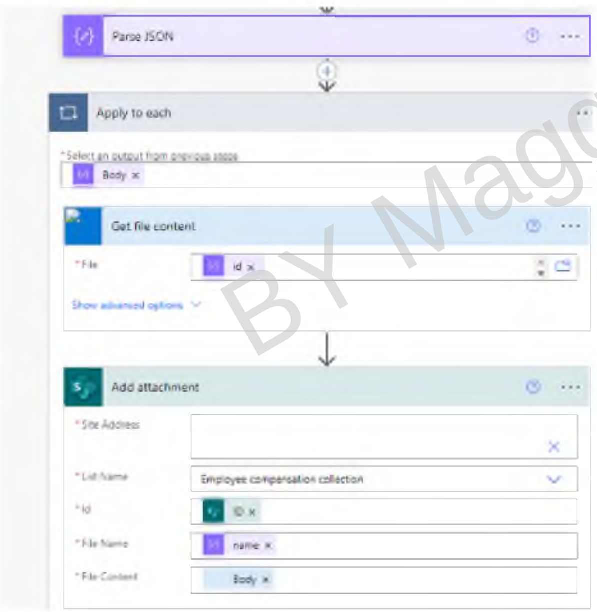
Microsoft Form & List

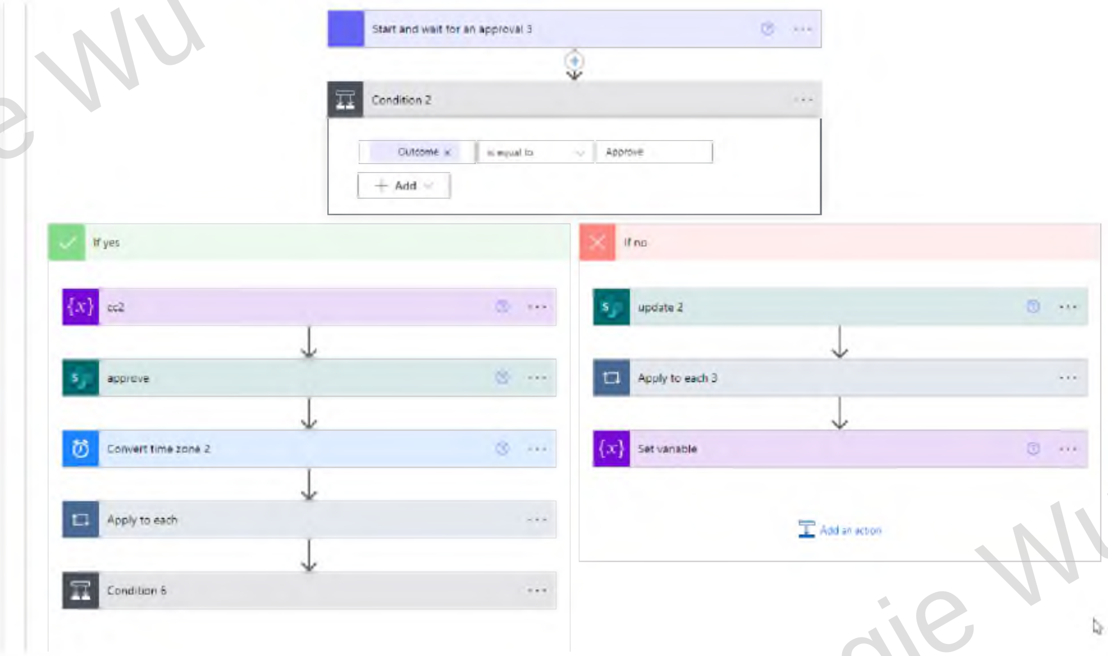




Automatic workflow and abnormal analysis

Workflow





/03

Database Design & its Application Project with Oracle Apex

BY Maggie Wu

BY Maggie Wu

Project overview:

In today's fast-paced business world, learning and development play a crucial role in the success of any organization. In large organizations, learning management systems (LMS) are used to manage the training and development of employees. However, for many medium and small-sized companies, the cost of such systems can be a barrier to realize this for their employees.

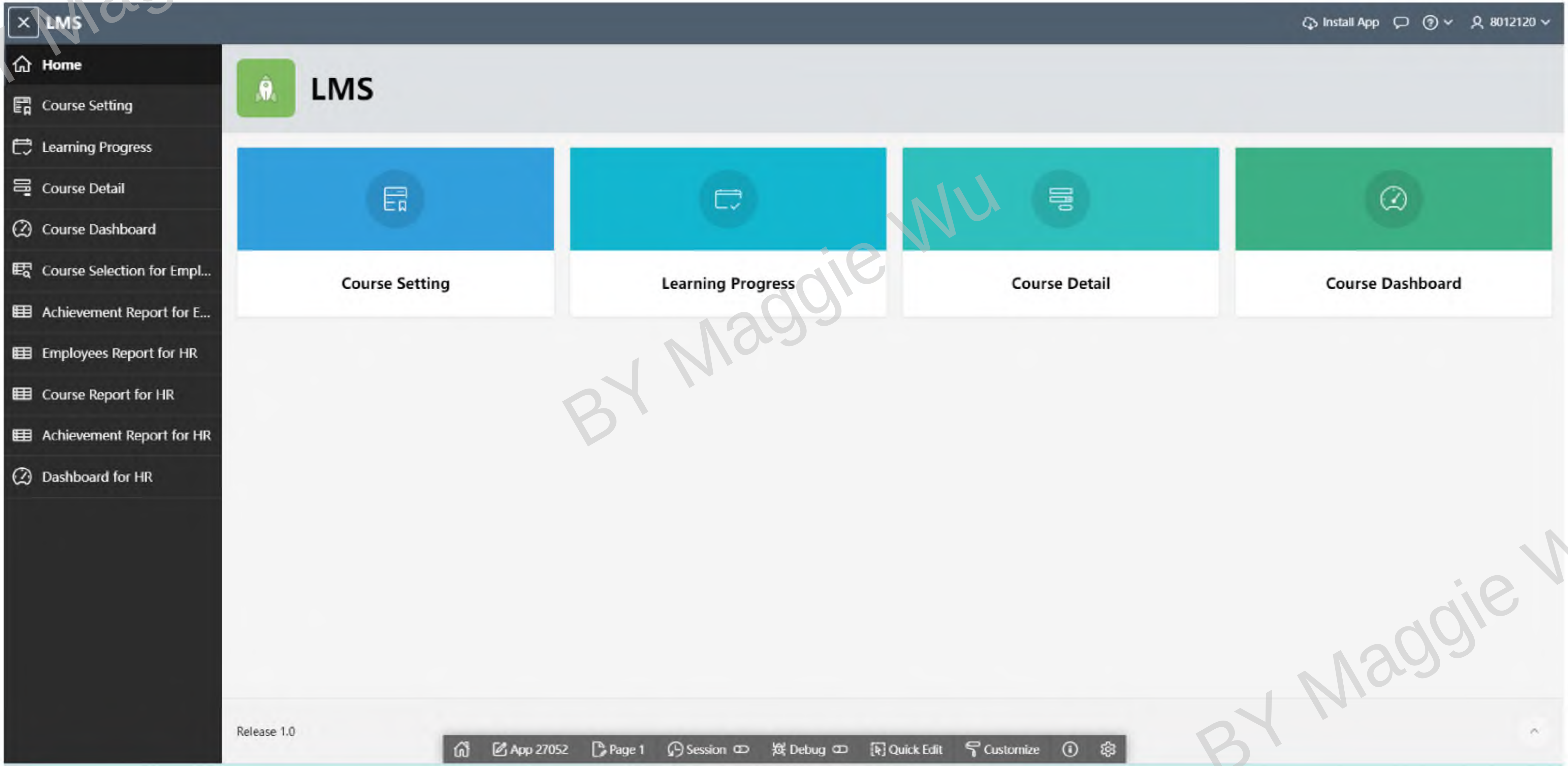
To address this problem, I plan to design a simple learning management application (LMS) that will enable small and medium-sized companies to manage their employee learning progress more effectively and to track the progress to ensure they are meeting company's learning objectives.

To develop this app, I use Oracle SQL to design a database to store the sample data, Oracle PL/SQL to make the trigger for automatically updating and new records ID creation, and Oracle APEX to design the LMS APP by means of the forms, reports, and dashboard function with SQL queries. Besides, I set 2 types of app user role for different authorization. Finally, I made integration test to ensure the app function correctly.

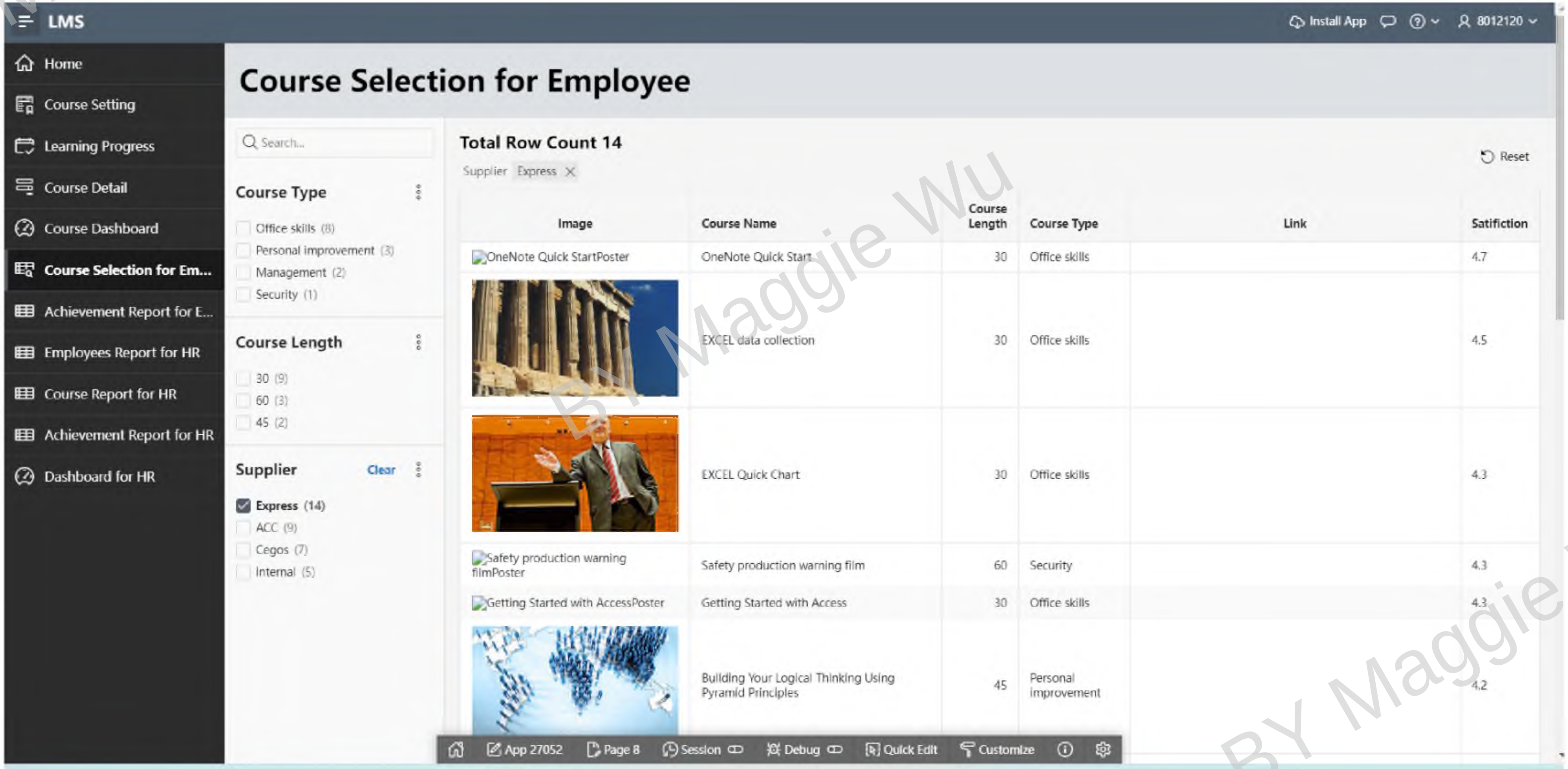
To ensure that the app is user-friendly, I have designed it to be easy to use and navigate. Users will be able to access the necessary forms, reports, and dashboards with just a few clicks. I have also included helpful features, such as search, filter, dashboard functions, to make it easier to find the information needed by users.


LMS System
LMS
Username
Password
Remember me
Sign In
App 24871
Page 9999
Session
Debug
Quick Edit
Customize

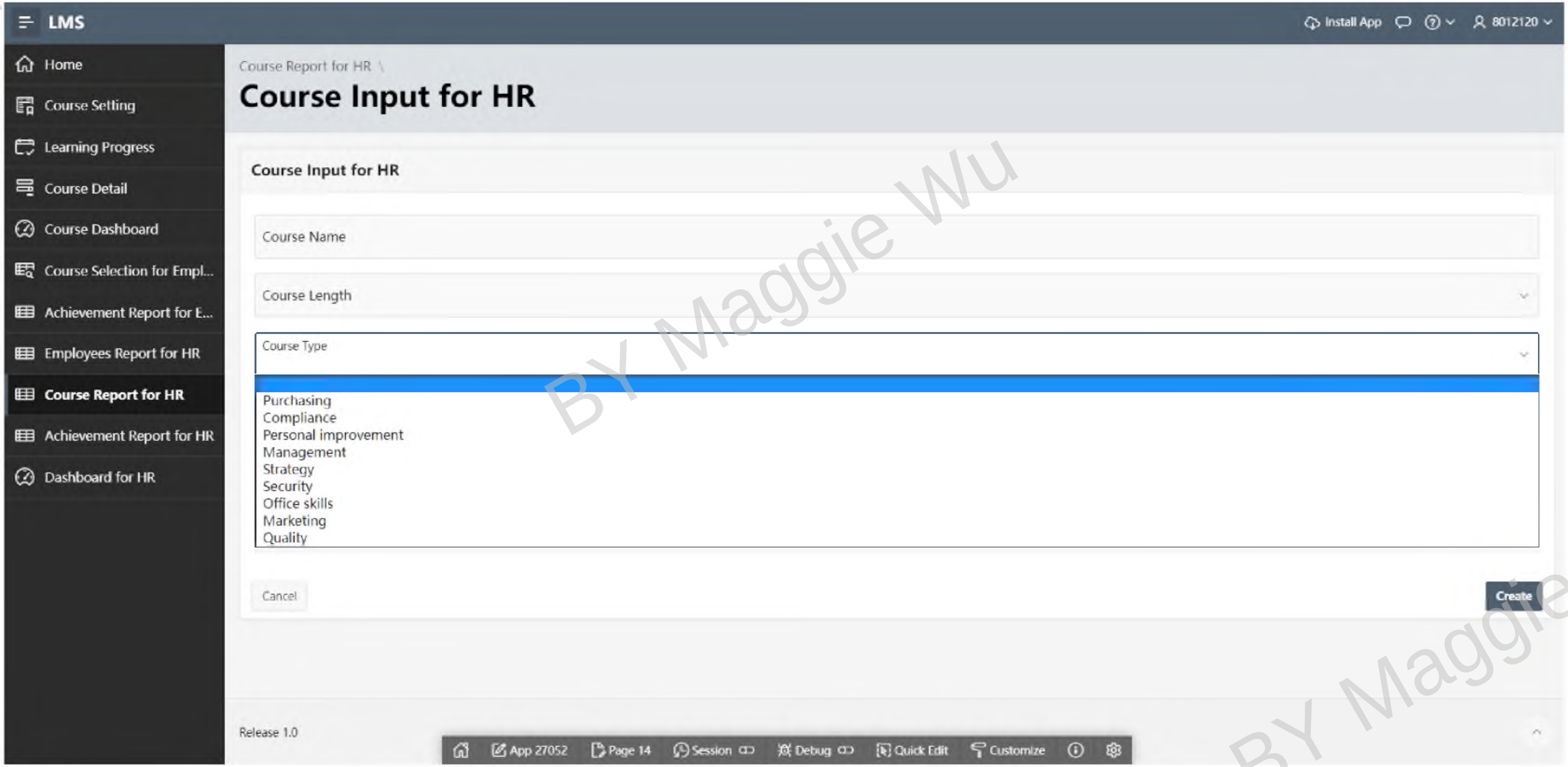
Database Design & its Application Project with Oracle Apex



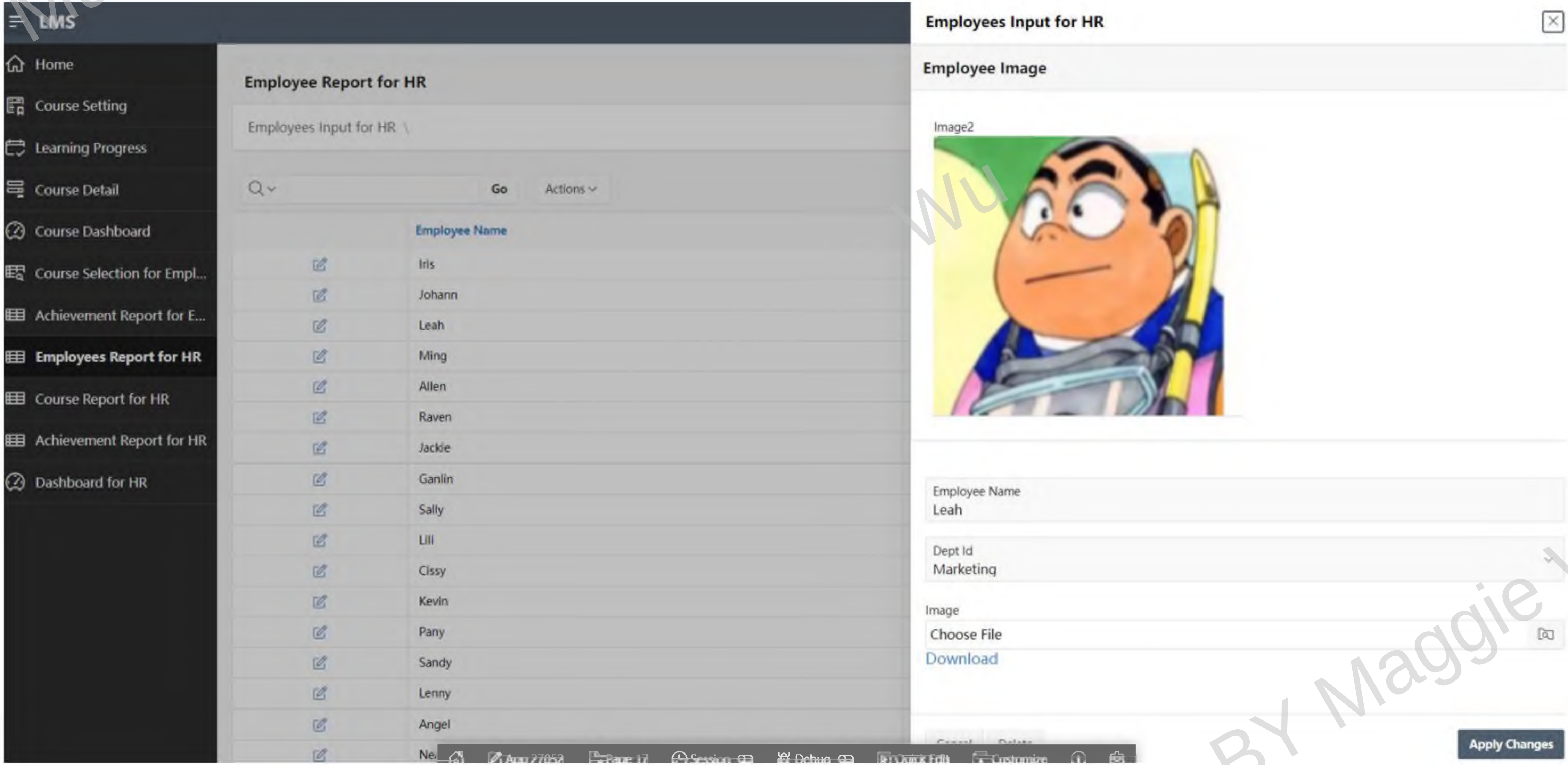
Database Design & its Application Project with Oracle Apex



Database Design & its Application Project with Oracle Apex



Database Design & its Application Project with Oracle Apex



Database Design & its Application Project with Oracle Apex

LMS

Install App | 8012120

- Home
- Course Setting
- Learning Progress
- Course Detail
- Course Dashboard
- Course Selection for Empl...
- Achievement Report for E...
- Employees Report for HR
- Course Report for HR
- **Achievement Report for ...**
- Dashboard for HR

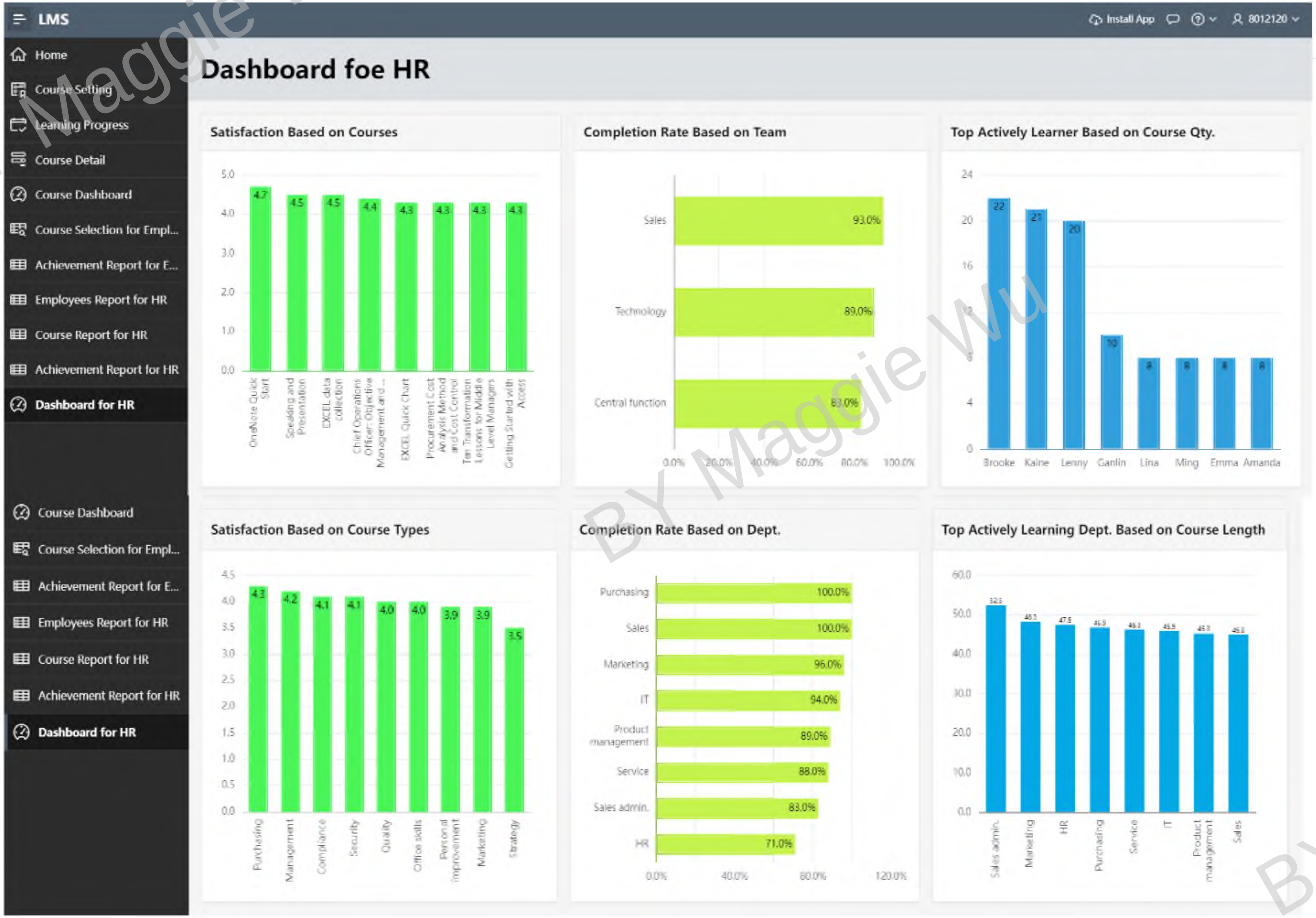
Achievement Report for HR

Go | Actions | Create

	Course	Employee	Course Status	Record Date	Satisfaction	Course Selection Type
	Safety production warning film	Iris	Completed	10/11/2022	4	HR administrator
	Safety production warning film	Jackie	Completed	4/19/2022	5	HR administrator
	Safety production warning film	Lili	Completed	10/13/2022	3	HR administrator
	Safety production warning film	Sandy	Completed	4/10/2022	5	HR administrator
	Export Control Compliance Training	Lenny	Completed	2/21/2022	4	HR administrator
	Export Control Compliance Training	Brooke	Completed	3/8/2022	4	HR administrator
	Export Control Compliance Training	Angel	On going	7/11/2022	5	HR administrator
	Compliance training	Neal	Completed	3/2/2022	3	HR administrator
	Compliance training	Cissy	Completed	3/2/2022	5	HR administrator
	Compliance training	Alberto	Completed	3/2/2022	3	HR administrator
	Compliance training	Miya	Completed	3/2/2022	5	HR administrator
	Compliance training	Bella	Completed	3/2/2022	5	HR administrator
	Compliance training	Charles	Completed	3/2/2022	4	HR administrator
	Compliance training	Fiore	On going	4/15/2022	4	HR administrator
	Compliance training	Lina	On going	7/7/2022	4	HR administrator
	Compliance training	Jerry	Completed	4/22/2022	5	HR administrator
	Compliance training	Anna	Completed	4/22/2022	5	HR administrator
	Compliance training			22	3	HR administrator

App 27052 | Page 15 | Session | Debug | Quick Edit | Customize

Database Design & its Application Project with Oracle Apex



/04

Fianncing Analysis with R

BY Maggie Wu

BY Maggie Wu

Project overview: Portfolio Analyzing & Optimizing

As the global economic trend is affected by multiple factors such as politics, global trade, and climate, the financial investment industry is facing enormous challenges. In recent years, quantitative trading based on financial big data and data science has been widely used in the investment field. Quantitative investment is an investment technology that integrates big data science, statistics, economics, finance and information science. Through this project, we master how to use R language to calculate the optimal investment portfolio.

1) By Sharpe Ratio

The Sharpe ratio is a measure of return often used to compare the performance of investment managers by making an adjustment for risk. People can compare investments and assess the amount of risk that each one has per percentage point of return. This helps people better control their risk exposure. The higher the rate, the

more returns the investment offers relative to the risks involved. Firstly, calculating mean(profit) and variance(risk), and then using a random sampling of portfolio weights, finally by find the point of tangency (the risk free line tangent to the efficient frontier), which is also with the highest Sharpe Ratio;

2) By Sortino Ratio

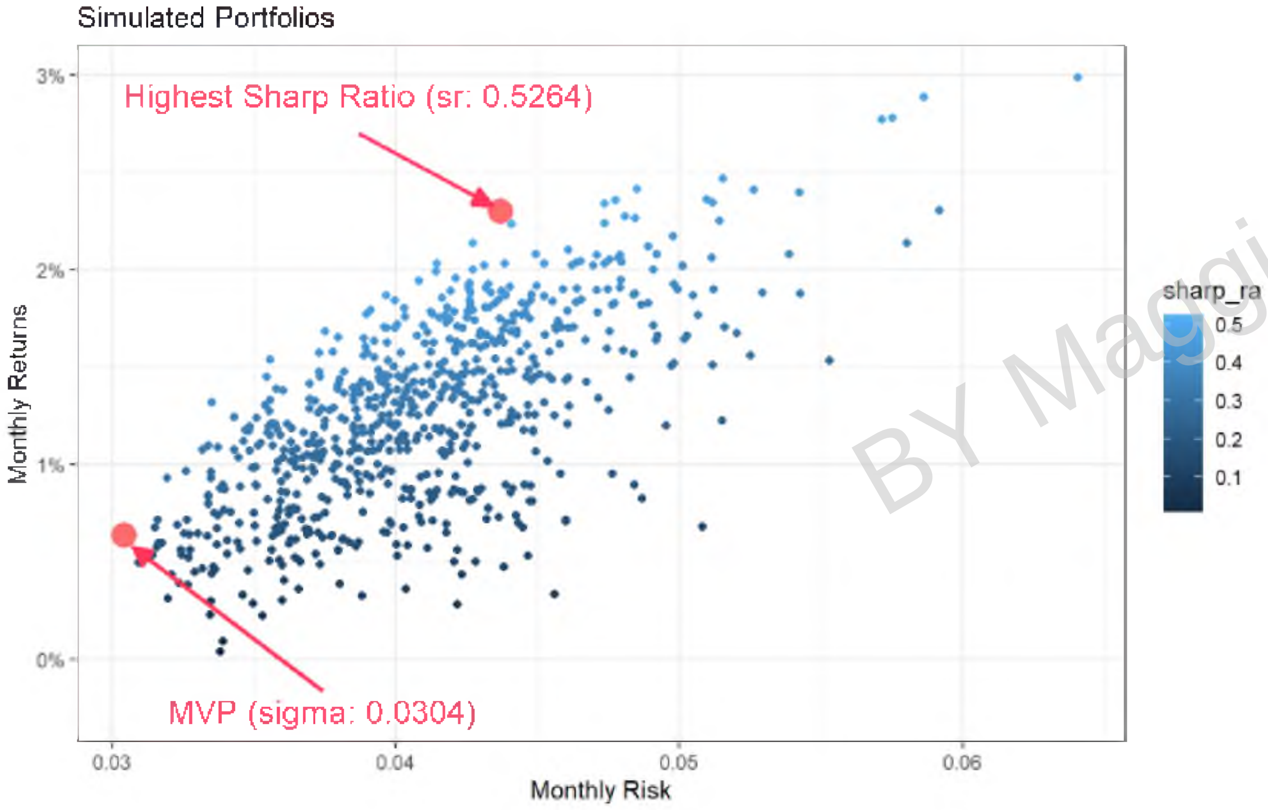
It uses downside deviation instead of standard deviation in Sharpe Ratio, that is measure the risk only returns falling below a specified target, while above target are set to zero.

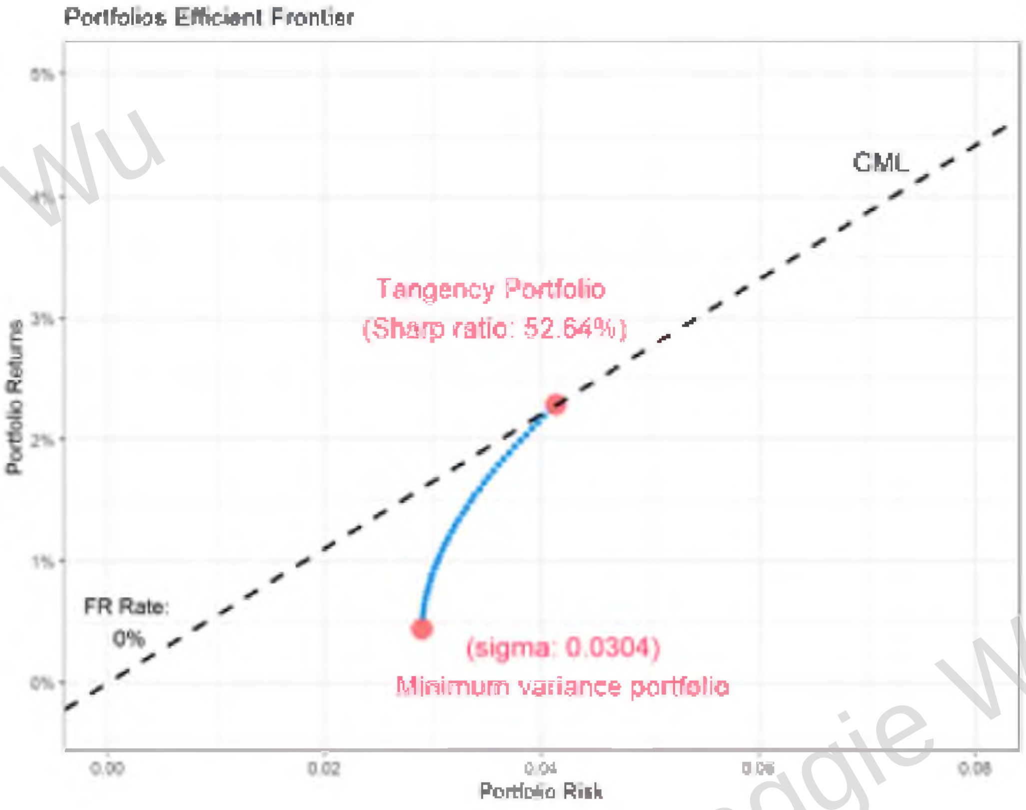
3) Quadratic Optimization

By using package PortfolioAnalytics we can design the optimal portfolio according our demand, e.g. to let weight add to 1, set the portfolio return to certain target and set the individual weight limit.

Fianncing Analysis with R

分析结果：Portfolio Analyzing & Optimizing





/05

Supplychain Analysis with Python using Linear Programming

Supplychain Analysis with Python using Linear Programming

Project overview:

- Where to set up the Plant to realize the minimum cost?
- Which plant needs to be closed due to higher cost?

Modeling

- Production at regional facilities
  - Two plant sizes (low / high)
- Exporting production to other regions
- Production facilities open / close

Supplychain Analysis with Python using Linear Programming

Coding Result:

In [76]:

```
## ppt 案例 case study
import pulp
# Initialize Class
model = pulp.LpProblem("Capacitated Plant Location Model", pulp.LpMinimiz
# Define Decision Variables
locc = ['USA','Germany','Japan','Brazil','India']
size = ['Low_Cap','High_Cap']


# Supply_Region
# USA            2719.6
# Germany          84.1
# Japan          1676.8
# Brazil          145.4
# India           156.4


# demand={'USA': 2719.6,
#  'Germany': 84.1,
#  'Japan': 1676.8,
#  'Brazil':145.4,
#  'India':156.4}


demand={('USA','Dmd'): 2719.6,
 ('Germany','Dmd'): 84.1,
 ('Japan','Dmd'): 1676.8,
 ('Brazil','Dmd'):145.4,
 ('India','Dmd'):156.4}


fix_cost={('USA', 'Low_Cap'): 6500,
 ('Germany', 'Low_Cap'): 4980,
 ('Japan', 'Low_Cap'): 6230,
 ('Brazil', 'Low_Cap'): 3230,
 ('India', 'Low_Cap'): 2110,
 ('USA', 'High_Cap'):9500,
 ('Germany', 'High_Cap'): 7270,
 ('Japan', 'High_Cap'): 9100,
 ('Brazil', 'High_Cap'): 4730,
 ('India', 'High_Cap'): 3080}
```

```
 ('Japan', 'India'): 9,
 ('Brazil', 'USA'): 12,
 ('Brazil', 'Germany'): 14,
 ('Brazil', 'Japan'): 21,
 ('Brazil', 'Brazil'): 8,
 ('Brazil', 'India'): 21,
 ('India', 'USA'): 22,
 ('India', 'Germany'): 13,
 ('India', 'Japan'): 10,
 ('India', 'Brazil'):23,
 ('India', 'India'): 8}

cap={('USA', 'Low_Cap'): 500,
 ('Germany', 'Low_Cap'): 500,
 ('Japan', 'Low_Cap'): 500,
 ('Brazil', 'Low_Cap'): 500,
 ('India', 'Low_Cap'): 500,
 ('USA', 'High_Cap'): 1500,
 ('Germany', 'High_Cap'):1500,
 ('Japan', 'High_Cap'): 1500,
 ('Brazil', 'High_Cap'): 1500,
 ('India', 'High_Cap'): 1500}



x =  pulp.LpVariable.dicts("production",[(i,j) for i in locc for j in locc],
                           lowBound=0, upBound=None, cat='Continuous') #Integer
y = pulp.LpVariable.dicts("plant",[(i,s) for s in size for i in locc], cat='Bina
# Define objective function
model += (pulp.lpSum([fix_cost[(i,s)]*y[(i,s)] for s in size for i in locc])
          + pulp.lpSum([var_cost[(i,j)]*x[(i,j)] for i in locc for j in locc]))

# for c in locc:
#     model += pulp.lpSum([cap[(i, s)] for s in size for i in locc ]) == demand[c,

# Define the Constraints
for j in locc:
    model += pulp.lpSum([x[(i, j)] for i in locc]) ==  demand[j,'Dmd']  #demand[j
for i in locc:
    model += pulp.lpSum([x[(i, j)] for j in locc]) <= pulp.lpSum([cap[(i,s)]*y[(i
                                                       for s in size])
```

```
# Solve Model
model.solve()
# 6. 打印结果
print(model)  # 输出问题设定参数和条件
print("优化状态:", pulp.LpStatus[model.status])
for v in model.variables():
    print(v.name, "=", v.varValue)
print("最优总成本 = ", pulp.value(model.objective))
```

```
production_('USA',_'Japan') Continuous
production_('USA',_'USA') Continuous

优化状态: Optimal
plant_('Brazil',_'High_Cap') = 1.0
plant_('Brazil',_'Low_Cap') = 0.0
plant_('Germany',_'High_Cap') = 0.0
plant_('Germany',_'Low_Cap') = 0.0
plant_('India',_'High_Cap') = 0.0
plant_('India',_'Low_Cap') = 1.0
plant_('Japan',_'High_Cap') = 1.0
plant_('Japan',_'Low_Cap') = 0.0
plant_('USA',_'High_Cap') = 1.0
plant_('USA',_'Low_Cap') = 0.0
production_('Brazil',_'Brazil') = 145.4
production_('Brazil',_'Germany') = 0.0
production_('Brazil',_'India') = 0.0
production_('Brazil',_'Japan') = 0.0
production_('Brazil',_'USA') = 1219.6
```

Thank you!