

2021 Woori Digital Academy **DAM**(Digital Analytic Master) 과정
프로젝트 결과보고서

(Financial Apps Review Analytics서비스 포함)



목차

 DAM 과정 소개

 과정 내용

02 실습 및 과제 수행 서비스 구현 프로젝트 수행



DAM 과정 소개

Digital Analytic Master 과정 내용

과정 내용(약식)

✔ Hadoop-빅데이터 분산처리 HDFS (파일시스템) MapReduce(병렬처리) Flume(로그수집) Sqoop(ETL, 데이터 조회) Hive(HDFS 데이터 조회)

✔ Spark-데이터 처리 엔진 Spark RDD(데이터 연산) Spark SQL(데이터 조회) Spark Streaming(실시간 처리) Spark ML Lib(머신러닝 라이브러리)

✔ Kafka(실시간 분산 데이터 스트리밍)



실습 및 과제 수행

과정 내용을 활용한 데이터 서비스 구현 Financial Apps Review Analytics



프로젝트 목표

금융 앱 서비스 강•약점 분석

- ✔ 국내,외 금융사 앱 리뷰의 긍/부정 분류를 통한 강/약점 분석
- ✓ 오픈소스 중심으로 저비용 조건하에 기 구축된 그룹 내 서비스를 오마주하여 개발
- ✓ Hadoop의 HDFS/MapReduce를 통한 Word Count 처리 및 Rank

Demand 요약 및 Notification/Report

- ✔ 앱 리뷰 내 언급 횟수를 기반으로 사용자 요구를 카테고리 분류하고 보고서를 제공하는 서비스
- ✓ 멀티플랫폼(Android,iOS,Windows,ChromeOS,Linux) Notification



프로젝트 주제 선정 배경

핀테크 앱, 금융뱅킹앱 추월...토스·카뱅 앱 접속건 수 1·2위

한국 소비자들은 지난해 금융 앱을 약 410억회 방문했고, 일주일 평균 은행 월렛 앱은 4.6회, <mark>핀테</mark>크 월렛 앱에는 11.7회 방문한 것으로 나타났다.

뱅크 월렛 앱은 1위가 KB스타뱅킹, 2위 NH스마트뱅킹, 3위 신한 쏠, 4위 우리은행 원터치개인뱅킹, 5위에는 신한페이판이 올랐다.

애플, 아마존에 이어 구글 등 빅테크 기업이 금융 분야로의 진출에 속도를 내고 있다.

23일 외신 등에 따르면 구글은 지난 18일(현지시간) 은행 계좌 개설과 간편송금, 생활비 관리 등 의 기능이 추가된 '구글 페이'를 출시했다. 핀테크, 인터넷은행의 시장점유 증가

시중은행의 비대면 서비스 확대

빅테크 기업의 금융 진출

우리금융의 MAU 증대를 위한 **비대면 서비스 활성화 전략 필요**



구축 Preview

개발 Workflow

개발환경

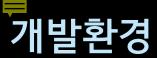
시스템 구성도

Data Flow

비고. 그룹 내 관련 서비스

개발 Workflow

데이터 수집	전처리	모델 학습	모델 평가	빈도 분 석
Play스토어, 앱스토어 크롤링 4대 은행 앱 인터넷은행 앱	자연어 분석 단어 토큰화 불용어 정의 불용어 처리 이상치 제거	궁/부정 분류기 SVM KNN RandomForest	Precision Recall F1 Score ROC Curve Accuracy	HDFS WordCount Spark Hadoop
핀테크앱(토스 외)	데이터 시각화	기능 상세 분류기	Loss	Service
8대 카드 앱 은행의 각종 앱 리뷰 수집	워드 클라우드 그래프	LSTM GRU		Push-Slack WebService



Server Instance





Managing Source Codes













KMS, Wiki, Managing Project







Languages











PIMS



Database





Data Sources



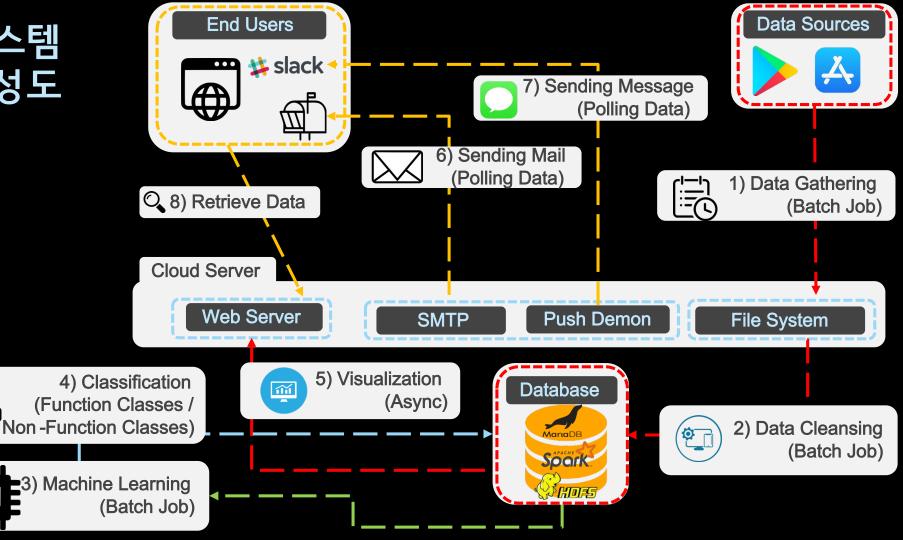


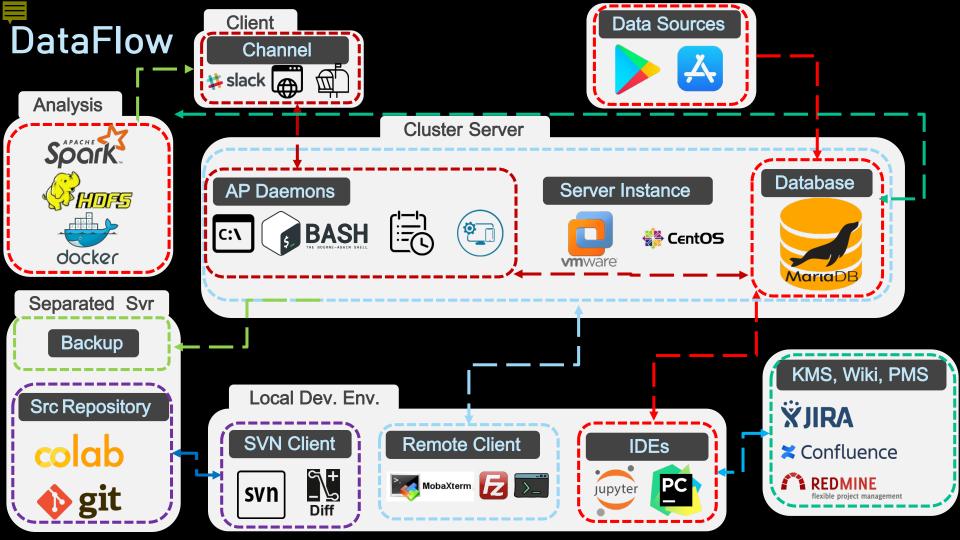
Channel





시스턴 구성5





(비고)그룹 내 관련서비스(앱리뷰모니터링)

1.Graph/Filter/DashBoard

2021-10-11

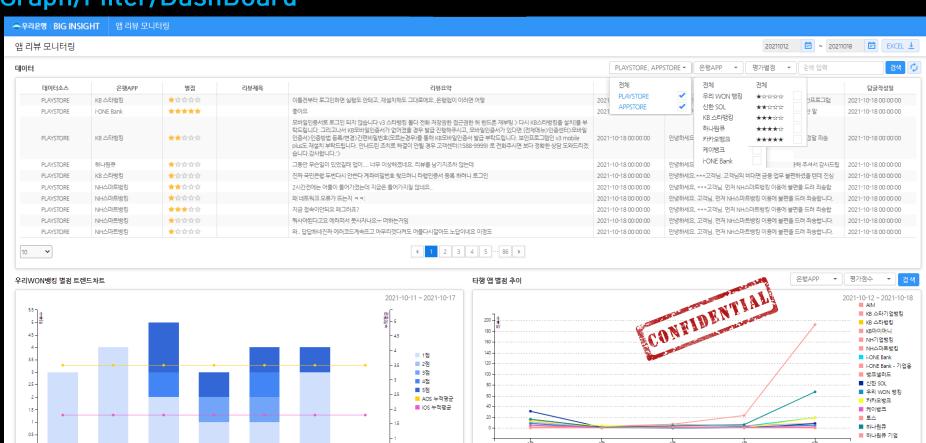
2021-10-12

2021-10-13

2021-10-14

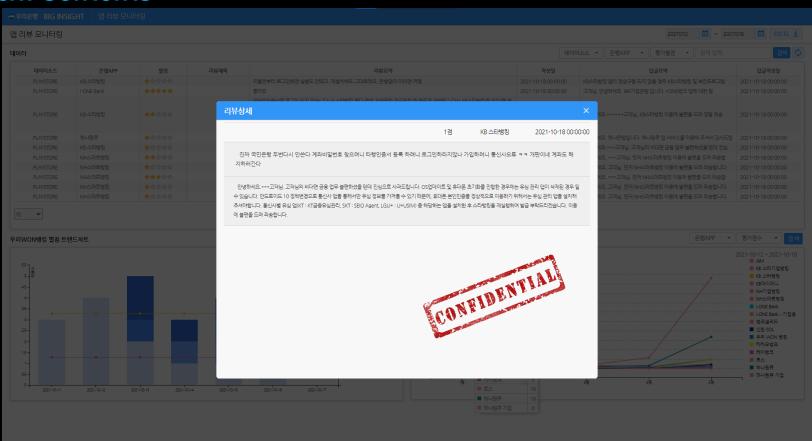
2021-10-15

2021-10-16



(비고)그룹 내 관련서비스(앱리뷰모니터링)

2. Review Text Contents



(비고)그룹 내 관련서비스(앱리뷰모니터링)

3. Internal Email Push



메일발송문의:80700-4437

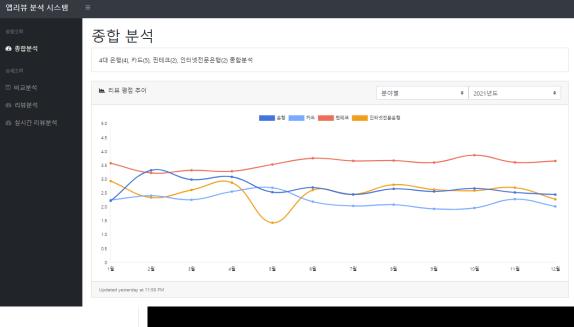


구축 서비스 소개

WebService / Push / Batch / Etc

Web Service

Report(평점 변화 추이, 궁/부정 Words) http://kaydash.iptime.org







Updated yesterday at 11:59 PM

♣ 부정 KEYWORD-WORD CLOUD

Web Service

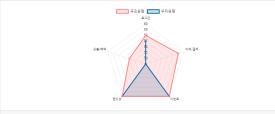
Differencial Analytics

우리은행 vs 국민은행 우리카드 vs 현대카드



조회 대상:국민은행(분야:은행)

Le 키워드별 평점 분석



리뷰분석 건수 : 국민은행 (104 건) / 우리은행 (36 건)

LM 강점 상세				
키워드	최근 평가			
이벤트	이벤트도다른쪽보다많			
	오류때문에회원가입이			
이체/결제	입출금카톡알임정재식			
	단축이체한번에다건이			
로그인	안녕하세요갑자기지문			
	앱토스만큼깔끔하고빨			
기능/편의성	이벤트도다른쪽보다많			
	업데이트다운로드가안			

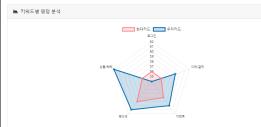
LM 약점 상세		
키워드	최근 평가	
이벤트	회원가입에어려움이크	
	진짜너무짜증나네요회	
이체/결제	방금에이불통장에입금	
	보안이슈가심각합니다	
로그인	국민은행을떠날때가된	
	업데이트후에로그인이	
기능/편의성	국민은행을떠날때가된	
	업데이트후에로그인이	





금융사별 분석

조회 대상 : 현대카드 (분야: 카드사)



₩ 약정 상세

리뷰분석 건수 : 현대카드 (103 건) / 우리카드 (102 건)

2 00 0 1				
키워드		최근 평가		
	이벤트	혜택이다양하고좋아요		
		좋은할인정보많아서좋		
이체/결제		스마일페이랑네이버카		
		카드사용이편리하긴한		
	로그인	앱화면이안뜹니다비번		
		인터페이스편하고세련		
	기능/편의성	편리하게사용하고있습		

서비스최고...

키워드	최근 평가
이벤트	쓰레기같은리불빙안내
이팬트	눈치없이자꾸눌리는현
이체/결제	빨리다시업데이트해봐
이제/될세	앱카드결제먹통앱삭제
로그인	1결제후며칠지나야결
	지문등록하고휴대폰인
기능/편의성	앱카드결제먹통앱삭제
10/2-10	1결제후며칭지나야결
	1일세주네일시니아일



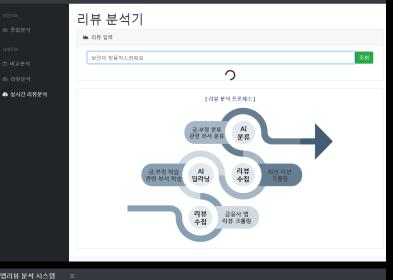


Copyright © Your Website 2020 Privacy Policy - Terms & Conditions

Web Service

앱리뷰 분석 시스템

Text Analytics





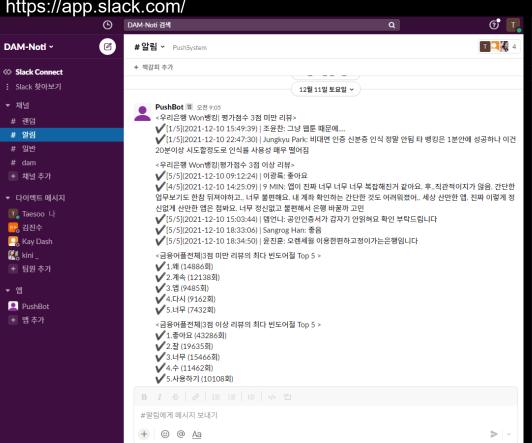
리뷰 분석

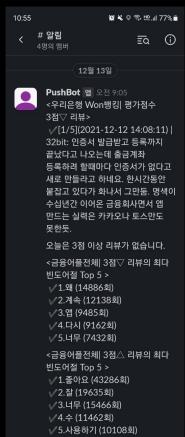




Push Service(Web / Mobile)

App Push(Slack, daily AM 09:00 ~ 09:05 Push) https://app.slack.com/







Data Lineage

Level	Task(For)	Data(Do)	Source(From)	Destination(To)	Parallel Proc(How Many) ETL Type(How to)	ETL Method(What)	Note.(Memo)	Loop
0	데이터 스키마 정의	Business	Psuedo Code	Logical Objects	1 Abstraction to Knowledge	Realize	Create Table	Once
0	테이블 파티셔닝	Split	Table	Partitioning Table	1 -	-	for Performance	Once
0	인덱스 생성	B*Tree	RootNode	LeafNodes	1 -	-	for Performance	Once
0	모니터링 환경 구축	Performance Data	Resource Usage	WebService	1 text To Visual	Data Gathering	Prepare for DataLineage	Once
1	웹 데이터 수집	Web Documents	GooglePlayStore	ServerStorage	20 Text To File	Web Crawling	Incremental Data	Daily
2	원천 테이블 적재	ReviewTextFile	ServerStorage	DBMS	20 File To DBMS	DBMS Injection	RDBMS(MySQL)	Daily
3	데이터 클렌징	Partitioned Tables	Raw Table	Raw Table	20 -	-	Cleansing Duplicated Data	Daily
4	요약 테이블 생성	Partitioned Tables	Raw Table	Summary Table	1 -	-	Summary(Score/Perf/MainDept)	Daily
5 .	보고서 테이블 생성	Partitioned Tables	Summary Table	Report Table	1 -	-	Report(Score/Perf/MainDept)	Daily
6	테이블 최적화	Statistics	Raw Data	Metadata	1 -	-	27 Tables	Daily
7	WordCloud 이미지 생성	Image Files	Data	Image	1 Data to MediaFiles	DataProcessing	Renewal Imgs.	Daily
8	Text 추출	TextFile	Data	TextFile	1 Data to Text	DataProcessing	Extract Text	Daily
9	HDFS 업로드	put HDFS	TextFile	HDFS	1 put HDFS	put HDFS	TextFile to HDFS	bit15
10	WordCount	Word Count	HDFS(input)	HDFS(output)	1 WordCount	hadoop WordCount	hdfs jar	Daily
11	word Ranking	Ranking	HDFS(output)	Data	1 HDFS to Data	Store Ranking Data	Data	Daily
12	Push Msg	Push Msg	Data	Push Msg	1 Data to Push Msg	Push	Slack	Daily

Batch TimeLine

```
[root@localhost ~]# crontab -l
5 0 * * * sh /Data/Batch/1 removeCSV.sh
                                                                   > /Data/Batch/1_removeCSV.log
                                                                                                                                                       # 1 min
                                                                                                                     2>&1
                                                                                                                               # start At. 00:05
10 0 * * * sh /Data/Batch/3 getNewFiles.sh
                                                                   > /Data/Batch/3 getNewFiles.log
                                                                                                                                                        # 15 min
                                                                                                                               # start At. 00:10
25 1 * * * sh /Data/Batch/4 loadAllFiles.sh
                                                                   > /Data/Batch/4 loadAllFiles.log
                                                                                                                     2>&1
                                                                                                                               # start At. 01:25
                                                                                                                                                        # 5 min
30 1 * * * sh /Data/Batch/9 deleteDupRows.sh
                                                                   > /Data/Batch/9 deleteDupRows.log
                                                                                                                     2>&1
                                                                                                                                                        # 5 min(No.9 is faster after others.)
                                                                                                                               # start At. 01:30
35 1 * * * sh /Data/Batch/6-2_summary.sh
                                                                   > /Data/Batch/6-2_summary.log
                                                                                                                                                        # 10 min(parallel with 6-2,3,4)
                                                                                                                     2>&1
                                                                                                                               # start At. 01:35
45 1 * * * sh /Data/Batch/6-3_getPerfData.sh
                                                                   > /Data/Batch/6-3_getPerfData.log
                                                                                                                     2>&1
                                                                                                                                                        # 10 min(parallel with 6-2,3,4)
                                                                                                                               # start At. 01:45
55 1 * * * sh /Data/Batch/6-4_genWordCloud.sh
                                                                   > /Data/Batch/6-4 genWordCloud.log
                                                                                                                     2>&1
                                                                                                                               # start At. 01:55
                                                                                                                                                        # 60 min(parallel with 6-2,3,4)
00 2 * * * sh /Data/Batch/6 OPTIMIZE TABLE.sh
                                                                   > /Data/Batch/6 OPTIMIZE TABLE.log
                                                                                                                     2>&1
                                                                                                                               # start At. 03:00
                                                                                                                                                        # 30 min
00 4 * * * sh /Data/Batch/8 getRows.sh
                                                                   > /Data/Batch/8 getRows.log
                                                                                                                     2>&1
                                                                                                                               # start At. 04:00
                                                                                                                                                        # 1 min
30 4 * * * /IDEs/anaconda3/bin/python /Data/Batch/modelTrainer.py > /Data/Batch/modelTrainer.log
                                                                                                                     2>&1
                                                                                                                               # start At. 04:30
                                                                                                                                                        # 2h 10 min
00 8 * * * /IDEs/anaconda3/bin/python /Data/Batch/B01 word count hadoop.py > /Data/Batch/B01 word count hadoop.log 2>&1
                                                                                                                               # start At. 08:00
                                                                                                                                                        # 30 min
00 9 * * * sh /Data/Batch/A01 enqueue avg msgs.sh
                                                                  > /Data/Batch/A01 enqueue avg msgs.log
                                                                                                                     2>&1
                                                                                                                               # start At. 09:00
                                                                                                                                                        # 1 min
05 9 * * * /IDEs/anaconda3/bin/python/Data/Batch/A02 dequeue avg msgs.pv > /Data/Batch/A02 dequeue avg msgs.log
                                                                                                                    2>&1
                                                                                                                               # start At. 09:05
                                                                                                                                                        # 1 min
```

Data Interaction Process

RDBMS -> HDFS(WordCount) -> RDBMS(Rank)

Text Cleans

Put HDFS

```
os.system('su - hdroot -c "/data/hadoop-3.3.1/bin/hdfs dfs -put
/Data/Batch/word_count/source/low/*.csv /input/low/"')
os.system('su - hdroot -c "/data/hadoop-3.3.1/bin/hdfs dfs -put
/Data/Batch/word_count/source/high/*.csv /input/high/"')
```

Ranked Data Push Process

RDBMS -> HDFS(WordCount) -> RDBMS(Rank)

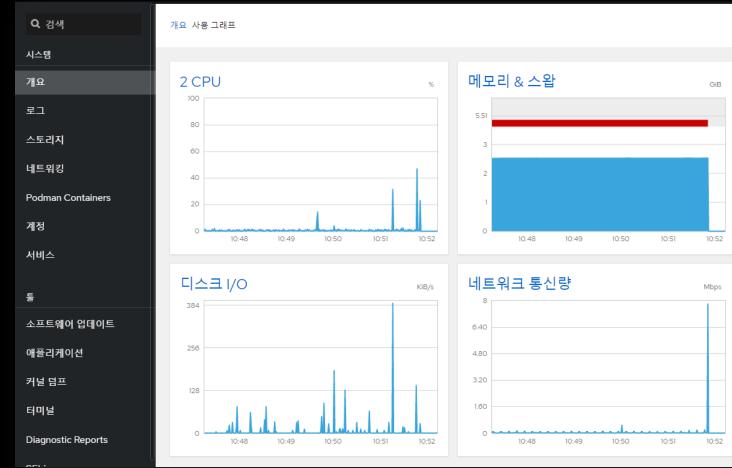
Data Ranking

```
sql='''update summary.under_avg_msg_set_send_flag=1 where send_flag=0 and
at >= date add(now(), interval - 72 hour)'''
cursor execute(sql)
connection.commit()
sql='''select score,at,username,original content,reviewid from
low rvs= pd.read sql query(sql,connection)
sql='''update summary.under_avg_msg set send_flag=2 where send flag=1'''
cursor.execute(sql)
connection.commit()
```

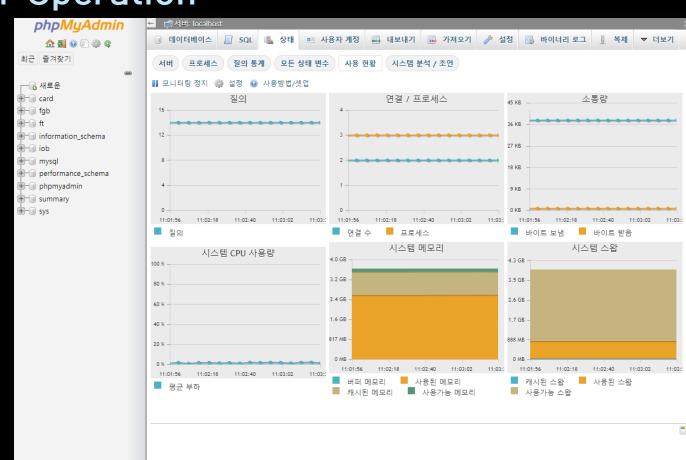
Enqueue/Dequeue Message(Push)

```
msg txt='''<금융어플전체│ 3점△ 리뷰의 최다 빈<u>도어절 Top 5 ></u>
i=0
for index, row in high words.iterrows():
   i=i+1
   msg txt=msg txt+''' ✔{ranking}.{words} ({count}회)
'''.format(ranking=int(i),words=row['word'],count=int(row['cnt']))
if len(high words)>0:
   post message(myToken, chnl, msg txt)
print(msg_txt)
```

Server(Linux)



DBMS(MySQL)



■ 콘솔

SVN

SubversionEdge by CollabNet Repositories MAINTENANCE Status Server Logs Software Updates Statistics Jobs CONFIGURATION Server Settings Authentication Proxy Server Mail Server Logging Server Monitoring EXTENSIONS TeamForge

Tip: Need help configuring or managing your Subversion environment?
CollabNet provides support, training and services. Learn more on the Extensions page.

정보

Users

Administration

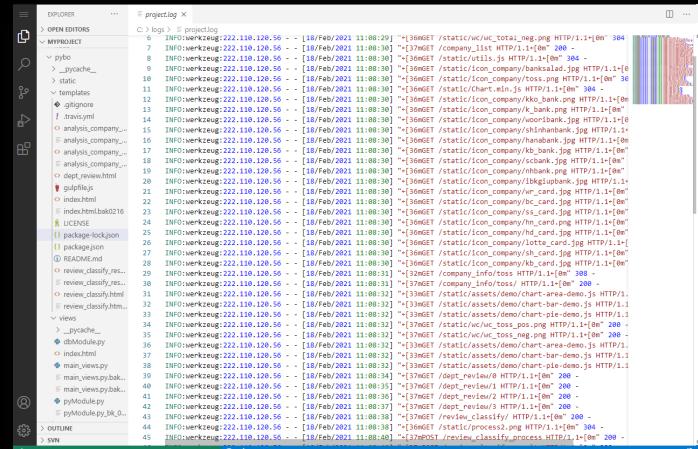
Extensions

Software version 5.2.4-4429.59 Subversion version 1.8.19-4429.59 Running since 02/18/2021 09:51:15 KST Repo health Total repositories: 1 - Go to the Repositories tab to create or discover repositories. Throughput on primary interface 0.0 B/s IN (over about 5 minutes); 0.0 B/s OUT (over about 5 minutes) Disk Usage as of 02/18/2021 10:53:16 KST Used space on root volume 27.32 GB Used space by repositories 0.0 B Free space on repository volume 36.51 GB	<u> </u>	
Running since 02/18/2021 09:51:15 KST Repo health Total repositories: 1 - Go to the Repositories tab to create or discover repositories. Throughput on primary interface 0.0 B/s IN (over about 5 minutes); 0.0 B/s OUT (over about 5 minutes) Disk Usage as of 02/18/2021 10:53:16 KST Used space on root volume 27.32 GB Used space by repositories 0.0 B	Software version	5.2.4-4429.59
Repo health Total repositories: 1 - Go to the Repositories tab to create or discover repositories. Throughput on primary interface 0.0 B/s IN (over about 5 minutes); 0.0 B/s OUT (over about 5 minutes) Disk Usage as of 02/18/2021 10:53:16 KST Used space on root volume 27.32 GB Used space by repositories 0.0 B	Subversion version	1.8.19-4429.59
Throughput on primary interface 0.0 B/s IN (over about 5 minutes); 0.0 B/s OUT (over about 5 minutes) Disk Usage as of 02/18/2021 10:53:16 KST Used space on root volume 27.32 GB Used space by repositories 0.0 B	Running since	02/18/2021 09:51:15 KST
Disk Usage as of 02/18/2021 10:53:16 KST Used space on root volume 27.32 GB Used space by repositories 0.0 B	Repo health	Total repositories: 1 - Go to the Repositories tab to create or discover repositories.
Used space on root volume 27.32 GB Used space by repositories 0.0 B	Throughput on primary interface	0.0 B/s IN (over about 5 minutes); 0.0 B/s OUT (over about 5 minutes)
Used space by repositories 0.0 B	Disk Usage	as of 02/18/2021 10:53:16 KST
	Used space on root volume	27.32 GB
Free space on repository volume 36.51 GB	Used space by repositories	0.0 B
	Free space on repository volume	36.51 GB

Super Administrator (admin)

로그아웃

Web Server(Flask)





기대 효과

프로젝트 기대효과 및 활용방안



기대 효과

- 입력 한 금융사의 기능/비기능적 강/약점 정보 제공
- 타 금융사의 장•단점을 분석하고 비교하여 우리금융의 개선방향 도출

활용방안

개선점 자동 알림

우리WON뱅킹 외 우리 금융 앱(optional) 개선의견 자동 알림으로 서비스 품질 개선 -> 기존 데이터 전처리 과정 축약을 통한 불만사항 인지 시간 감소(최대 1일)

리뷰 이력 조회

멀티 플랫폼 서비스로 내/외부망에 국한되지 않는 알림 이력 조회

신규 서비스 수요 예측

고객 선호 서비스 학습 -> 신사업 성공여부 예측



Fin.