머신러닝을 이용한 지하철 역 일일 승하차 인원 예측 모델링

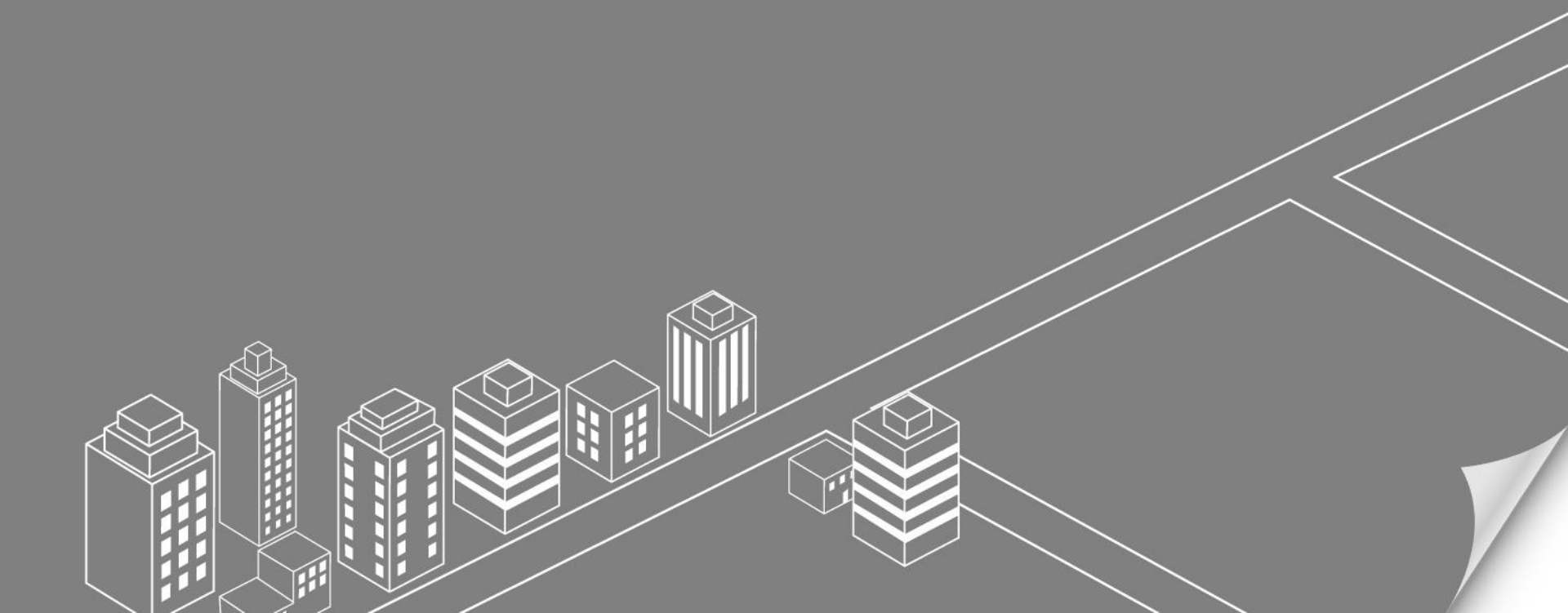
2017. 11. 22.

조경아

머신러닝을 이용한 지하철 역 일일 승하차 인원 예측 모델링

Contents

I. 목적



I. 목적

※ 의정부 경전철 파산

이와 같이 의정부 경전철 파산과 같은 관련 보도를 참고하면 지하철 역 선정 과정에서 지하철 수요 예측의 어려움을 짐작할 수 있다.

→ 따라서 보다 정확한 지하철 수요 예측 모델을 구현하고자 한다.

II. 데이터 수집 및 전처리

- 2016년도 전국 지하철 승하차 인원 관련 데이터
- 2016년도 12월 기준 소상공인 상권정보 상가업소 데이터
- 2016년도 전국 아파트(매매) 실거래 데이터

전국 지하철 승하차 인원 관련 데이터

1. 지하철 승하차 인원 관련 데이터

사용일자	노선명	역ID	역명	승차총승객수	하차총승객수	등록일자
20160101	2호선	210	뚝섬	5374	5117	20160109
20160101	2호선	211	성수	6730	6858	20160109
20160101	2호선	212	건대입구	31453	33343	20160109
20160101	2호선	213	구의	10553	9629	20160109
20160101	2호선	214	강변	32259	35307	20160109
20160101	2호선	215	잠실나루	6116	5839	20160109
20160101	2호선	216	잠실	50864	45006	20160109
20160101	2호선	217	신천	12433	12627	20160109
20160101	2호선	218	종합운동장	6469	6142	20160109
20160101	2호선	219	삼성	19324	19310	20160109
20160101	2호선	220	선릉	11593	9255	20160109
20160101	2호선	221	역삼	9643	9718	20160109
20160101	2호선	222	강남	37893	39308	20160109
20160101	2호선	223	교대	9581	9762	20160109

1. 지하철 승하차 인원 관련 데이터

사용일자	노선명	역ID	역명	승차총승객수	하차총승객수	등록일자
20160101	2호선	210	뚝섬	5374	5117	20160109
20160101	2호선	211	한양대	7[illegible]	7858	20160109
20160101	2호선	212	왕십리	31458	33343	20160109
20160101	2호선	213	구의	10553	9629	20160109
20160101	2호선	214	신[illegible]	32[illegible]59	35307	20160109
20160101	2호선	215	잠실나루	6116	5839	20160109
20160101	2호선	216	잠[illegible]	43[illegible]86	45006	20160109
20160101	2호선	217	신천	12432	12627	20160109
20160101	2호선	218	종합운동장	6469	6142	20160109
20160101	2호선	219	삼[illegible]	19324	19310	20160109
20160101	2호선	220	선릉	11593	9255	20160109
20160101	2호선	221	역[illegible]	9643	9718	20160109
20160101	2호선	222	강남	37893	39308	20160109
20160101	2호선	223	교대	9581	9762	20160109

서울, 수도권 : 476개 역
대구 : 86개 역
부산 : 102개 역
대전 : 22개 역
광주 : 20개 역
용인 경전철 : 14개 역
의정부 경전철 : 14개 역

1. 지하철 승하차 인원 관련 데이터

```
# 2016년도 월별 지하철 승하차 인원 데이터 불러오기

m01<-read.csv("01.csv", header = T, stringsAsFactors = F)
m02<-read.csv("02.csv", header = T, stringsAsFactors = F)
m03<-read.csv("03.csv", header = T, stringsAsFactors = F)
m04<-read.csv("04.csv", header = T, stringsAsFactors = F)
m05<-read.csv("05.csv", header = T, stringsAsFactors = F)
m06<-read.csv("06.csv", header = T, stringsAsFactors = F)
m07<-read.csv("07.csv", header = T, stringsAsFactors = F)
m08<-read.csv("08.csv", header = T, stringsAsFactors = F)
m09<-read.csv("09.csv", header = T, stringsAsFactors = F)
m10<-read.csv("10.csv", header = T, stringsAsFactors = F)
m11<-read.csv("11.csv", header = T, stringsAsFactors = F)
m12<-read.csv("12.csv", header = T, stringsAsFactors = F)

# rbind - 1월 ~ 12월 데이터 통합
rawdata <- rbind(m01,m02,m03,m04,
                 m05,m06,m07,m08,
                 m09,m10,m11,m12)

# rawdata 에서 col3(역ID), col7(등록일자) 삭제
rawdata <- rawdata[,c(-3,-7)]
```

1. 지하철 승하차 인원 관련 데이터

```
# 총승객수 (승차총승객수 + 하차총승객수) 변수 생성
총승객수 <- rawdata$승차총승객수 + rawdata$하차총승객수
rawdata <- cbind(rawdata, 총승객수)

# 승차, 하차 변수 삭제
rawdata <- rawdata[,c(-4,-5)]

# rawdata 변수명 변경
colnames(rawdata) <- c("date","line_nm","station_nm","total")
```

1. 지하철 승하차 인원 관련 데이터

```
# 역 별 관측치 갯수 확인
A <- rawdata %>% group_by(station_nm, line_nm) %>% summarise(n = length(station_nm))

# 291 개 짜리 역명
B <- subset(A, select = c("station_nm"), subset = (n==291))
C <- subset(A, select = c("station_nm"), subset = (n==75))

station_name <- cbind(B,C)
station_name

# 역명 변환
rawdata$station_nm <- gsub("[[:punct:]]+.+[[:punct:]]", "", rawdata$station_nm)
rawdata$station_nm <- gsub("동두천 중앙", "동두천중앙", rawdata$station_nm)
rawdata$station_nm <- gsub("쌍용동", "쌍용", rawdata$station_nm)
rawdata$station_nm <- gsub("신천", "잠실새내", rawdata$station_nm)
rawdata$station_nm <- gsub("서울역", "서울", rawdata$station_nm)
rawdata$station_nm <- gsub("이수", "총신대입구", rawdata$station_nm)
rawdata$line_nm <- gsub("9호선2단계", "9호선", rawdata$line_nm)


# 역 별 관측치 갯수 확인
A <- rawdata %>% group_by(station_nm, line_nm) %>% summarise(n = length(station_nm))
table(A$line_nm)
```

1. 지하철 승하차 인원 관련 데이터

	station_nm	line_nm	n
33	경복궁	3호선	291
34	경복궁(정부서울청사)	3호선	75
35	경찰병원	3호선	366
36	계양	공항철도 1호선	366
37	고덕	5호선	366
38	고려대	6호선	291
39	고려대(종암)	6호선	75

station_nm	station_nm1
강변	강변(동서울터미널)
경복궁	경복궁(정부서울청사)
고려대	고려대(종암)
공릉	공릉(서울과학기술대)
광나루	광나루(장신대)
광화문	광화문(세종문화회관)
광흥창	광흥창(서강)
교대	교대(법원.검찰청)
교대	교대(법원.검찰청)
구의	구의(광진구청)

1. 지하철 승하차 인원 관련 데이터

```r
# 1~9호선 역 별 일일 사용자수 데이터 생성
# 1호선
line01 <-
  rawdata %>%
  filter(line_nm == c("1호선")) %>%
  group_by(station_nm, line_nm) %>%
  summarise(mean=round(mean(total)))

# 2호선
line02 <-
  rawdata %>%
  filter(line_nm == c("2호선")) %>%
  group_by(station_nm, line_nm) %>%
  summarise(mean=round(mean(total)))
```

```r
## 수도권 13개 노선 ::: []는 이상치
# 경부선 [서울]
cap01 <-
  rawdata %>%
  filter(line_nm == c("경부선")) %>%
  filter(station_nm != "서울") %>%
  group_by(station_nm, line_nm) %>%
  summarise(mean=round(mean(total)))
# 경원선 [창동]
cap02 <-
  rawdata %>%
  filter(line_nm == c("경원선")) %>%
  filter(station_nm != "창동") %>%
  group_by(station_nm, line_nm) %>%
  summarise(mean=round(mean(total)))
```

1. 지하철 승하차 인원 관련 데이터

```
# 1~9호선병합
line_all <- rbind(line01,line02,line03,line04,line05,line06,line07,line08,line09)

# 수도권 병합
cap_all <- rbind(cap01,cap02,cap03,cap04,cap05,cap06,cap07,cap08,cap09,cap10,cap11,cap12,cap13)

# 서울 + 수도권 병합
subway_01 <- rbind(line_all, cap_all)

# 서울 수도권 지하철 중복 노선 통합
subway_01_dt <-
  subway_01 %>%
  group_by(station_nm) %>%
  summarise(user = round(sum(mean)))
```

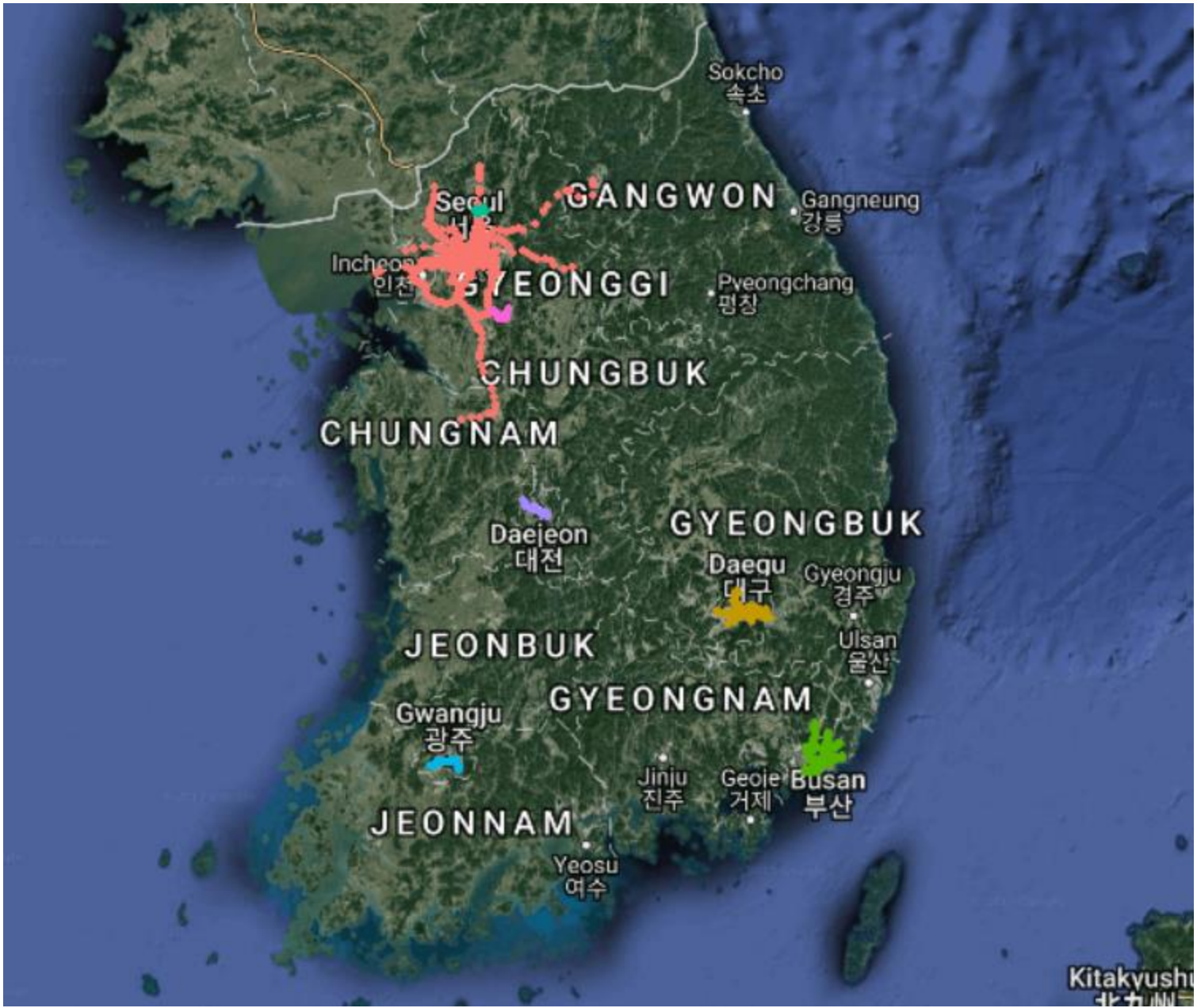
1. 지하철 승하차 인원 관련 데이터

```
# Data 불러오기
subway_01_dt <- read.csv("D:/R/subway/subway_01_dt.csv", header = T, stringsAsFactors = F)

# 검색어 word 생성 :: 지하철 이름 UTF-8 인코딩
word <- enc2utf8(paste(subway_01_dt$station_nm, "역", sep=""))

# geocode를 이용해 경도, 위도 생성
gc <- read.csv("D:/R/subway/gc_01.csv", header = T, stringsAsFactors = F)

# 좌표 합치기
subway_01_set <- cbind(subway_01_dt, gc)
```

1. 지하철 승하차 인원 관련 데이터

index	station_nm	lon	lat	user
1	가능	127.0443	37.74836	15038
1	가락시장	127.1181	37.49232	32841
1	가산디지털	126.8826	37.48089	114668
1	가양	126.8542	37.56155	40613
1	가좌	126.9148	37.56861	8506
1	가천대	127.1267	37.44878	19973
1	가평	127.5107	37.81452	5760
1	간석	126.6934	37.46469	13494
1	갈매	127.1148	37.63414	1589
1	강남	127.0276	37.49794	199596
1	강남구청	127.0413	37.51717	50015
1	강동	127.1322	37.53594	35657
1	강동구청	127.1205	37.53051	20862
1	강매	126.8454	37.6126	4591
1	강변	127.0947	37.53514	99851

II. 데이터 수집 및 전처리

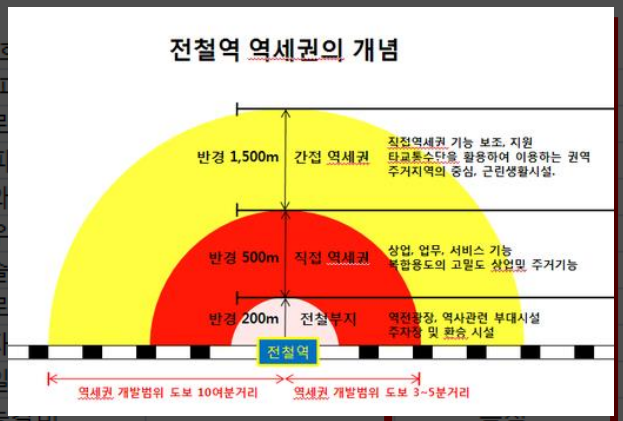


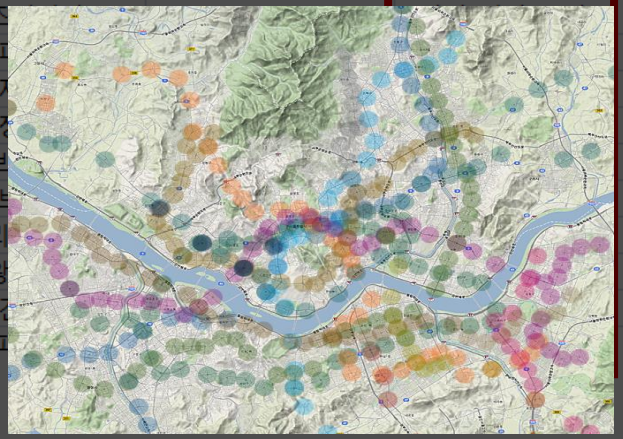
소상공인 상권정보 상가업소 데이터

2. 소상공인 상권정보 상가업소 데이터

상호명	지점명	상권업종대분류명	상권업종중분류명	상권업종소분류명	지번주소	경도	위도
커피빈	코리아선릉로93길점	음식	커피점/카페	커피전문점/카페/다방	서울특별시 강남구 역삼동 696-42	127.0479	37.50568
프로포즈		음식	유흥주점	호프/맥주	서울특별시 금천구 독산동 162-1	126.8992	37.47171
싱싱커피&토스트		음식	패스트푸드	토스트전문	부산광역시 사상구 괘법동 578	128.9805	35.15977
와라와라호프		음식	유흥주점	호프/맥주	서울특별시 강남구 대치동 604	127.061	37.49392
가락사우나내스낵		생활서비스	대중목욕탕/휴게	사우나/증기탕/온천	서울특별시 송파구 석촌동 256	127.1041	37.50025
허술한집		음식	분식	라면김밥분식	서울특별시 강서구 공항동 45-89	126.81	37.56201
씨유씨로지스틱스		소매	종합소매점	편의점	서울특별시 중구 순화동 5-2	126.9708	37.56259
피자마루	약수점	음식	패스트푸드	피자전문	서울특별시 중구 신당동 372-95	127.0109	37.55215
원일슈퍼		소매	종합소매점	수퍼마켓	서울특별시 강동구 천호동 315-6	127.1272	37.54766
홍능갈비		음식	한식	갈비/삼겹살	서울특별시 강서구 화곡동 1086-3	126.8361	37.53893
프렌즈피씨방		관광/여가/오락	PC/오락/당구/볼링등	인터넷PC방	서울특별시 양천구 신월동 146-1	126.8289	37.53567
진커피호프		음식	유흥주점	호프/맥주	서울특별시 양천구 신정동 947-4	126.8544	37.52465
김명희자연비누		소매	종합소매점	시장/종합상가	서울특별시 양천구 신정동 294-5	126.8724	37.51786
대성정육점		소매	음/식료품소매	정육점	서울특별시 중구 신당동 43-14	127.0194	37.56027
에브리돈		음식	한식	한식/백반/한정식	서울특별시 서대문구 북가좌동 307-2	126.9103	37.57903
라이브7080		음식	유흥주점	룸살롱/단란주점	서울특별시 강남구 논현동 106-18	127.0361	37.51735
우리식당		음식	한식	한식/백반/한정식	서울특별시 광진구 화양동 507-3	127.0768	37.54546
왕실		음식	커피점/카페	커피전문점/카페/다방	서울특별시 중구 명동2가 105	126.9824	37.56227
신호건강원		소매	건강/미용식품	건강원	서울특별시 강서구 화곡동 369-53	126.8403	37.53542
Mr.피터팬		음식	분식	라면김밥분식	서울특별시 송파구 풍납동 399-17	127.1159	37.52748

2. 소상공인 상권정보 상가업소 데이터





지하철역 반경 700m 이내에 분포한 상가업소의 개수관련 변수 생성

상권업종중분류명	상권업종소분류명	지번주소	경도	위도
커피점/카페	커피전문점/카페/다방	서울특별시 강남구 역삼동 696-42	127.0479	37.50568
유흥주점	호프/맥주	서울특별시 금천구 독산동 162-1	126.8992	37.47171
패스트푸드	토스트전문	부산광역시 사상구 괘법동 578	128.9805	35.15977
유흥주	맥주	서울특별시 강남구 대치동 604	127.061	37.49392
대중목욕	탕/	서울특별시 [illegible]구	[illegible].1041	37.50025
분식	라면김밥분식	서울특별시 강서구 공항동 45-89	126.81	37.56201
종합소매점	편의	서울특별시 [illegible] 5-2	126.9708	37.56259
패스트푸드	피	서울특별시 [illegible]372-95	127.0109	37.55215
종합소매점	수퍼마켓	서울특별시 강동구 천호동 315-6	127.1272	37.54766
한	일반삼겹살	서울특별시 강서구 화곡동 [illegible]	[illegible]8361	37.53893
PC/오락/	인터넷	서울특별시 [illegible]	[illegible]8289	37.53567
유흥주	맥주	서울특별시 양천구 신정동 [illegible]4	[illegible]8544	37.52465
종합소매점	시장/종합시가	서울특별시 양천구 신정동 294-5	126.8724	37.51786
음/식료품소매	정	서울특별시 신당동 43-14	127.0194	37.56027
한식	한식/백반/한정식	서울특별시 서 구 북가좌동 307-2	126.9103	37.57903
유흥주점	룸살롱/단란주점	서울특별시 강남구 논현동 106-18	127.0361	37.51735
한식	한식/백반/한정식	서울특별시 광진구 화양동 507-3	127.0768	37.54546
커피점/카페	커피전문점/카페/다방	서울특별시 중구 명동2가 105	126.9824	37.56227
건강/미용식품	건강원	서울특별시 강서구 화곡동 369-53	126.8403	37.53542
분식	라면김밥분식	서울특별시 송파구 풍납동 399-17	127.1159	37.52748

2. 소상공인 상권정보 상가업소 데이터

```
store01 <- read.csv("D:/R/subway/store01.csv", header = T, stringsAsFactors = F)
store02 <- read.csv("D:/R/subway/store02.csv", header = T, stringsAsFactors = F)
store03 <- read.csv("D:/R/subway/store03.csv", header = T, stringsAsFactors = F)
store04 <- read.csv("D:/R/subway/store04.csv", header = T, stringsAsFactors = F)

store01 <- store01[,c(2,3,5,7,9,32,38,39)]
store02 <- store02[,c(2,3,5,7,9,32,38,39)]
store03 <- store03[,c(2,3,5,7,9,32,38,39)]
store04 <- store04[,c(2,3,5,7,9,32,38,39)]

store <- rbind(store01, store02, store03, store04)

# store 변수명 재지정하기

colnames(store) <- c("brand_nm","branch_nm","main_cg",
                     "middle_cg","small_cg","address",
                     "lon","lat")

# store.csv 생성 :: 전국 상가 데이터
write.csv(store, "D:/R/subway/store.csv", row.names = F)
```

2. 소상공인 상권정보 상가업소 데이터

```
# ▶ 대분류 필터링
main_01 <- store %>% filter(main_cg=="음식")
main_02 <- store %>% filter(main_cg=="생활서비스")
main_03 <- store %>% filter(main_cg=="소매")
main_04 <- store %>% filter(main_cg=="관광/여가/오락")
main_05 <- store %>% filter(main_cg=="학문/교육")
main_06 <- store %>% filter(main_cg=="의료")
main_07 <- store %>% filter(main_cg=="부동산")
main_08 <- store %>% filter(main_cg=="숙박")

# 관측치 1개씩 밖에 없으므로 아래 대분류는 제외
# main09 <- store %>% filter(main_cg=="전자/정보통신")
# main10 <- store %>% filter(main_cg=="도매/유통/무역")
# main11 <- store %>% filter(main_cg=="1차산업")

write.csv(main_01, "D:/R/subway/main_01.csv", row.names = F)
write.csv(main_02, "D:/R/subway/main_02.csv", row.names = F)
write.csv(main_03, "D:/R/subway/main_03.csv", row.names = F)
write.csv(main_04, "D:/R/subway/main_04.csv", row.names = F)
write.csv(main_05, "D:/R/subway/main_05.csv", row.names = F)
write.csv(main_06, "D:/R/subway/main_06.csv", row.names = F)
write.csv(main_07, "D:/R/subway/main_07.csv", row.names = F)
write.csv(main_08, "D:/R/subway/main_08.csv", row.names = F)
```

2. 소상공인 상권정보 상가업소 데이터

```
# 상가 카운팅 프로그램

for (j in 1:8){

  var <- paste("main_0", j, sep="")
  csv <- paste(var,".csv",sep="")
  url <- paste("D:/R/subway/",csv,sep="")

  variance <- read.csv(url, header = T, stringsAsFactors = F)
  variance <- variance %>% group_by(lon, lat) %>% summarise(a = sum(lon))
  variance <- data.frame(lon=variance$lon, lat=variance$lat)

  # 지구 반지름 r (단위:km) 위키백과 기준
  r=6378.137
  # 지하철 역 좌표
  lon1<-NISTdegTOradian(subway_all_set$lon)
  lat1<-NISTdegTOradian(subway_all_set$lat)
  # 상가 좌표
  lon2<-NISTdegTOradian(variance$lon)
  lat2<-NISTdegTOradian(variance$lat)
```

2. 소상공인 상권정보 상가업소 데이터

```
# 각각의 지하철 역에 대해 모든 상가와의 거리 저장
#paste(var,"_n",sep = "") <- NA
var_n <- NA
for (i in (1:length(subway_all_set[,1]))) {
  print(i)

  dlon<-lon2-lon1[i]
  dlat<-lat2-lat1[i]

  a<-sin(dlat/2)^2+cos(lat1[i])*cos(lat2)*sin(dlon/2)^2
  c<-2*atan2(sqrt(a),sqrt(1-a))
  distance <- r*c

  index <- distance <= 0.7

  var_n[i] <-
    length(variance[index, 1])
}

# 데이터 셋에 변수 추가
subway_all_set <- cbind(subway_all_set, var_n)
colnames(subway_all_set)[5+j] <- paste("var_n",j,sep="")
}
```

2. 소상공인 상권정보 상가업소 데이터

index	station_nn	lon	lat	user	var_n1	var_n2	var_n3	var_n4	var_n5	var_n6	var_n7	var_n8
1	가능	127.0443	37.74836	15038	417	301	485	32	110	51	50	7
1	가락시장	127.1181	37.49232	32841	312	195	277	107	89	38	69	7
1	가산디지털	126.8826	37.48089	114668	312	189	263	77	47	47	49	43
1	가양	126.8542	37.56155	40613	114	90	102	20	56	36	39	0
1	가좌	126.9148	37.56861	8506	177	127	195	16	51	31	26	9
1	가천대	127.1267	37.44878	19973	113	121	167	16	32	10	22	7
1	가평	127.5107	37.81452	5760	23	14	18	0	2	0	4	2
1	간석	126.6934	37.46469	13494	316	287	426	70	125	62	73	9
1	갈매	127.1148	37.63414	1589	20	9	27	1	2	0	7	0
1	강남	127.0276	37.49794	199596	641	347	368	110	190	135	177	25
1	강남구청	127.0413	37.51717	50015	448	467	450	76	104	98	117	11
1	강동	127.1322	37.53594	35657	602	398	448	120	130	109	148	55
1	강동구청	127.1205	37.53051	20862	334	242	267	44	126	75	75	1
1	강매	126.8454	37.6126	4591	43	40	56	13	46	17	22	0
1	강변	127.0947	37.53514	99851	150	98	127	33	78	24	54	4

II. 데이터 수집 및 전처리

	변수명	형식	설명
1	index	factor	지역 구분 (1:서울,2:대구,3:부산,4:의정부,5:광주,6:대전,7:용인)
2	station_nm	char	지하철 역 이름
3, 4	lon, lat	num	지하철 역 경도, 위도
5	user	int	지하철 역 일일 승하차 인원 (명)
6~13	var_n1~var_n8	int	지하철 역 인근 상가 업소 (대분류 별) 수 (개) (음식/서비스/소매/오락/교육/의료/부동산/숙박)
14	var_n9	int	지하철 역 인근 도서관 수 (개)
15	var_n10	int	지하철 역 인근 초중고교 수 (개)
16	var_n11	int	지하철 역 인근 대학교 수 (개)
17	var_n12	int	지하철 역 인근 공원 수 (개)
18	var_n13	int	지하철 역 인근 CCTV 수 (개)

지하철 역 반경 700m 이내에 분포한 개수 관련 변수 생성

전국 아파트(매매) 실거래 데이터

1. 전국 아파트(매매) 실거래 데이터

시군구	번지	단지명	전용면적(㎡)	계약일	거래금액(만원)	층	건축년도	도로명
서울특별시 강남구 개포동	141	개포주공 1단지	35.64	1~10	64,000	4	1982	개포로
서울특별시 강남구 개포동	141	개포주공 1단지	35.64	21~31	65,200	5	1982	개포로
서울특별시 강남구 개포동	141	개포주공 1단지	42.55	21~31	75,500	5	1982	개포로
서울특별시 강남구 개포동	141	개포주공 1단지	45.26	11~20	76,500	4	1982	개포로
서울특별시 강남구 개포동	141	개포주공 1단지	49.56	1~10	88,000	2	1982	선릉로
서울특별시 강남구 개포동	141	개포주공 1단지	49.56	11~20	90,000	3	1982	선릉로
서울특별시 강남구 개포동	141	개포주공 1단지	49.56	21~31	90,000	1	1982	선릉로
서울특별시 강남구 개포동	141	개포주공 1단지	50.64	11~20	89,000	3	1982	개포로
서울특별시 강남구 개포동	141	개포주공 1단지	50.64	11~20	87,300	2	1982	개포로
서울특별시 강남구 개포동	138	개포주공 3단지	42.55	21~31	87,500	4	1982	삼성로
서울특별시 강남구 개포동	138	개포주공 3단지	50.67	21~31	107,000	4	1982	삼성로
서울특별시 강남구 개포동	189	개포주공 4단지	35.87	21~31	64,500	5	1982	삼성로
서울특별시 강남구 개포동	189	개포주공 4단지	41.99	1~10	72,000	2	1982	삼성로
서울특별시 강남구 개포동	189	개포주공 4단지	41.99	11~20	72,900	3	1982	삼성로
서울특별시 강남구 개포동	189	개포주공 4단지	41.99	11~20	74,300	3	1982	삼성로
서울특별시 강남구 개포동	189	개포주공 4단지	41.99	21~31	74,000	4	1982	삼성로
서울특별시 강남구 개포동	189	개포주공 4단지	42.55	11~20	72,400	1	1982	삼성로
서울특별시 강남구 개포동	189	개포주공 4단지	42.55	11~20	73,000	3	1982	삼성로
서울특별시 강남구 개포동	189	개포주공 4단지	42.55	21~31	75,000	5	1982	삼성로

1. 전국 아파트(매매) 실거래 데이터

시군구	번지	단지명	전용면적(㎡)	계약일	거래금액(만원)	층	건축년도	도로명
서울특별시 강남구 개포동	141	개포주공 1단지	35.64	1~10	64,000	4	1982	개포로
서울특별시 강남구 개포동	141	개포주공 1단지	35.64	21~31	65,200	5	1982	개포로
서울특별시 강남구 개포동	141	개포주공 1단지	42.55	21~31	75,500	5	1982	개포로
서울특별시 강남구 개포동	141	개포주공 1단지	45.26	11~20	76,500	4	1982	개포로
서울특별시 강남구 개포동	141	개포주공 1단지	49.56	1~10	88,000	2	1982	선릉로
서울특별시 강남구 개포동	141	개포주공 1단지	49.56	11~20	90,000	3	1982	선릉로
서울특별시 강남구 개포동	141	개포주공 1단지	49.56	21~31	90,000	1	1982	선릉로
서울특별시 강남구 개포동	141	개포주공 1단지	50.64	11~20	87,300	2	1982	개포로
서울특별시 강남구 개포동	141	개포주공 1단지	50.64	11~20	87,300	2	1982	개포로
서울특별시 강남구 개포동	189	개포주공 4단지	35.87	1~10	64,000	4	1982	삼성로
서울특별시 강남구 개포동	189	개포주공 4단지	35.87	21~31	64,500	5	1982	삼성로
서울특별시 강남구 개포동	189	개포주공 4단지	41.99	1~10	72,000	2	1982	삼성로
서울특별시 강남구 개포동	189	개포주공 4단지	41.99	11~20	72,900	3	1982	삼성로
서울특별시 강남구 개포동	189	개포주공 4단지	41.99	11~20	74,300	3	1982	삼성로
서울특별시 강남구 개포동	189	개포주공 4단지	41.99	21~31	74,000	4	1982	삼성로
서울특별시 강남구 개포동	189	개포주공 4단지	42.55	11~20	72,400	1	1982	삼성로
서울특별시 강남구 개포동	189	개포주공 4단지	42.55	11~20	73,000	3	1982	삼성로
서울특별시 강남구 개포동	189	개포주공 4단지	42.55	21~31	75,000	5	1982	삼성로

지하철 역인근 아파트 매매 가격의 평균 평당 가격 변수생성

1. 전국 아파트(매매) 실거래 데이터

```
# 경기 실거래가 데이터 불러오기
kg <- NA;kg <- na.omit(kg)
for (i in 1:12){
  print(i)
  name <- paste("경기", i, sep="")
  name <- paste(name, "실", sep="")
  csv <- paste(name, ".csv", sep="")
  url <- paste("D:/R/subway/실거래가/", csv, sep="")
  kg_new <- read.csv(url, skip=6, header = T, stringsAsFactors = F)
  kg <- rbind(kg, kg_new)
}

# 서울 실거래가 데이터 불러오기
se <- NA;se <- na.omit(se)
for (i in 1:12){
  print(i)
  name <- paste("서울", i, sep="")
  name <- paste(name, "실", sep="")
  csv <- paste(name, ".csv", sep="")
  url <- paste("D:/R/subway/실거래가/", csv, sep="")
  se_new <- read.csv(url, skip=6, header = T, stringsAsFactors = F)
  se <- rbind(se, se_new)
}
```

1. 전국 아파트(매매) 실거래 데이터

```
#-----------------------------------------
#  경기 kg  #  서울 se  #  인천 ic  #  강원 gw
#  대전 dj  #  대구 dg  #  부산 bs  #  광주 gj
#  충남 cn
#-----------------------------------------

# 전국 실거래가 통합 데이터 생성
house <- rbind(kg, se, ic, gw, dj, dg, bs, gj, cn)
colnames(house)[6] <- "price"

house$price <- gsub(",", "", house$price)
house$price <- as.numeric(house$price)

head(house)
colnames(house)

house <- house[ ,c(1,2,3,4,6)]

summary(house$전용면적...)
```

1. 전국 아파트(매매) 실거래 데이터

```
# 전용면적 기준
unique(house$전용면적...)
length(house[house$전용면적...>=59.56 & house$전용면적...<=84.96, 1])
summary(house$전용면적...)

# 사분위수 범위 추출
house_size<-house[house$전용면적...>=59.56 & house$전용면적... <=84.96, ]


# 전국 부동산 가격 통합 데이터 house.csv 생성
write.csv(house_size,"D:/R/subway/실거래가/house_size.csv",row.names=F)
```

1. 전국 아파트(매매) 실거래 데이터

```
# ------------------------------------------------
# total_house 생성 코드 (평당 가격)
# ------------------------------------------------

house$평 <- house$전용면적...*0.3025
house$평당가격 <- (house$price) / (house$평)


house$address <- paste(house$시군구,house$단지명,sep=" ")
total_house <-house %>% group_by(address) %>% summarise(mean = mean(평당가격))

write.csv(total_house, "D:/R/subway/실거래가/total_house_norm.csv", row.names=F)

summary(total_house$mean)
```

1. 전국 아파트(매매) 실거래 데이터

```
gc_1000 <- read.csv("D:/R/subway/실거래가/gc_1000.csv", header=T, stringsAsFactors=F)
gc_2000 <- read.csv("D:/R/subway/실거래가/gc_2000.csv", header=T, stringsAsFactors=F)
gc_3000 <- read.csv("D:/R/subway/실거래가/gc_3000.csv", header=T, stringsAsFactors=F)
gc_4000 <- read.csv("D:/R/subway/실거래가/gc_4000.csv", header=T, stringsAsFactors=F)
gc_5000 <- read.csv("D:/R/subway/실거래가/gc_5000.csv", header=T, stringsAsFactors=F)
gc_6000 <- read.csv("D:/R/subway/실거래가/gc_6000.csv", header=T, stringsAsFactors=F)
gc_7000 <- read.csv("D:/R/subway/실거래가/gc_7000.csv", header=T, stringsAsFactors=F)
gc_8000 <- read.csv("D:/R/subway/실거래가/gc_8000.csv", header=T, stringsAsFactors=F)
gc_9000 <- read.csv("D:/R/subway/실거래가/gc_9000.csv", header=T, stringsAsFactors=F)
gc_10500 <- read.csv("D:/R/subway/실거래가/gc_10500.csv", header=T, stringsAsFactors=F)
gc_12000 <- read.csv("D:/R/subway/실거래가/gc_12000.csv", header=T, stringsAsFactors=F)
gc_13000 <- read.csv("D:/R/subway/실거래가/gc_13000.csv", header=T, stringsAsFactors=F)
gc_14000 <- read.csv("D:/R/subway/실거래가/gc_14000.csv", header=T, stringsAsFactors=F)
gc_15013 <- read.csv("D:/R/subway/실거래가/gc_15013.csv", header=T, stringsAsFactors=F)
```

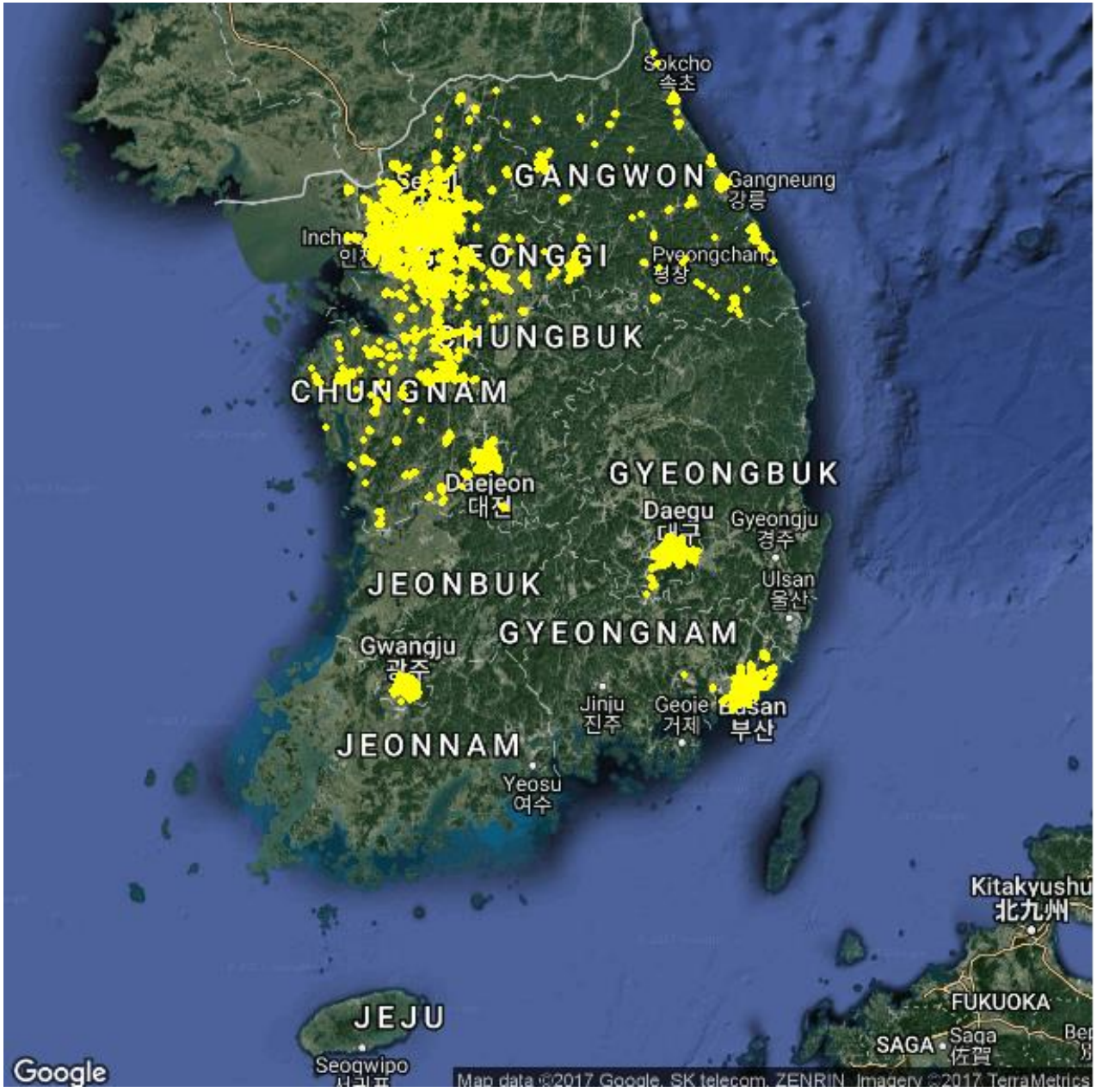
1. 전국 아파트(매매) 실거래 데이터

```
#==================================
# var_n15 :: 최근접 28개 평당 가격
#==================================

var_n15 <- NA
for (i in (1:length(subway_all_set[,1]))) {
  print(i)

  dlon<-lon2-lon1[i]
  dlat<-lat2-lat1[i]

  a<-sin(dlat/2)^2+cos(lat1[i])*cos(lat2)*sin(dlon/2)^2
  c<-2*atan2(sqrt(a),sqrt(1-a))
  distance <- r*c

  index <- sort(distance, index.return=T)$ix[1:28]

  var_n15[i] <-
    mean(variance[index, 3])
}

# 데이터 셋에 변수 추가
subway_all_set <- cbind(subway_all_set, var_n15)
```

1. 전국 아파트(매매) 실거래 데이터

var_n13	var_n14	var_n15
72	10	907.544
39	24	2015.999
72	23	1285.406
24	36	1679.144
27	43	1984.135
60	6	1349.804
10	5	608.1446
72	41	914.0371
4	1	1473.598
93	39	2793.145
115	43	3447.989
97	100	1672.809
72	43	1918.852
22	5	1276.761
56	28	2112.083

II. 데이터 수집 및 전처리

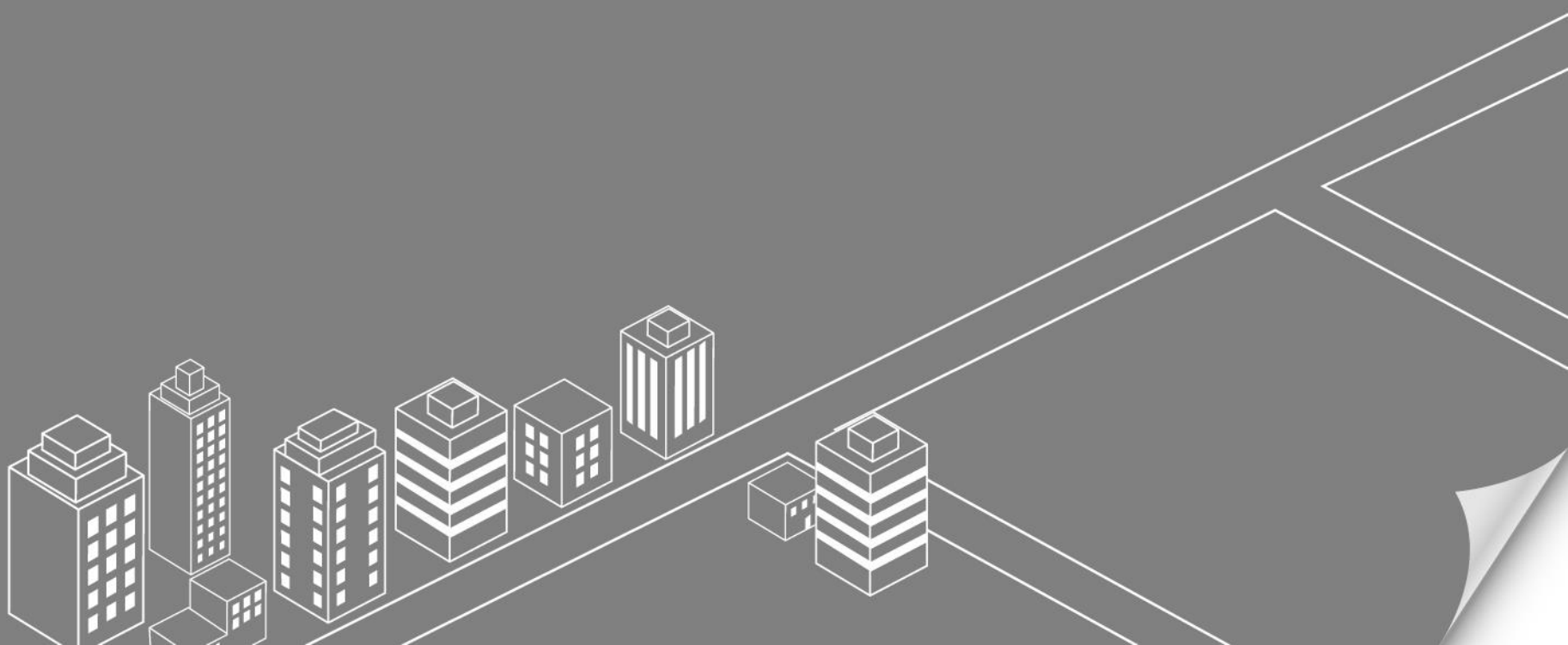


II. 데이터 수집 및 전처리

	변수명	형식	설명
1	index	factor	지역 구분 (1:서울,2:대구,3:부산,4:의정부,5:광주,6:대전,7:용인)
2	station_nm	char	지하철 역 이름
3, 4	lon, lat	num	지하철 역 경도, 위도
5	user	int	지하철 역 일일 승하차 인원 (명)
6~13	var_n1~var_n8	int	지하철 역 인근 상가 업소 (대분류 별) 수 (개) (음식/서비스/소매/오락/교육/의료/부동산/숙박)
14	var_n9	int	지하철 역 인근 도서관 수 (개)
15	var_n10	int	지하철 역 인근 초중고교 수 (개)
16	var_n11	int	지하철 역 인근 대학교 수 (개)
17	var_n12	int	지하철 역 인근 공원 수 (개)
18	var_n13	int	지하철 역 인근 CCTV 수 (개)
19	var_n14	int	지하철 역 반경 1km 이내 아파트 매매 거래 횟수 (회)
20	var_n15	num	지하철 역 기준 최근접 28개 아파트 매매 평균 평당 가격 (원)

III. 데이터 모델링

- Multi-Regression, Artificial Neural Network을 분석 모델로 선정한 후 지하철 수요 예측 모델링을 수행을 한다.



왜 General linear model(poisson)인가?

diff(range(w$var_n1)) # 1447
diff(range(w$var_n2)) # 242
diff(range(w$var_n3)) # 126
diff(range(w$var_n4)) # 217
diff(range(w$var_n5)) # 276
diff(range(w$var_n6)) # 257
diff(range(w$var_n7)) # 285
diff(range(w$var_n8)) # 319
diff(range(w$var_n9)) # 17
diff(range(w$var_n10)) # 14
diff(range(w$var_n11)) # 3
diff(range(w$var_n12)) # 28
diff(range(w$var_n13)) # 242
diff(range(w$var_n14)) # 126
diff(range(w$var_n15)) # 3985.317

diff(range(w$user)) # 199402

1. General Linear Model (poisson)

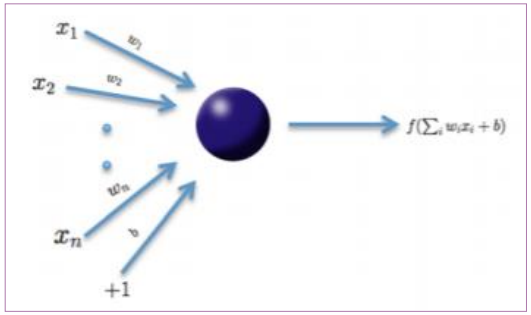
- Poisson regression은 반응변수가 Count 변수 이거나 오차가 포아송 분포를 가진다고 가정한다. 반응변수가 음수가 아닌 어떤 데이터에도 적용 가능하다.
- $\hat{y} = e^{x^T \beta + \beta_0}$
- 다음의 Penalized Likelihood를 최대로 높이게끔 적합 된다.

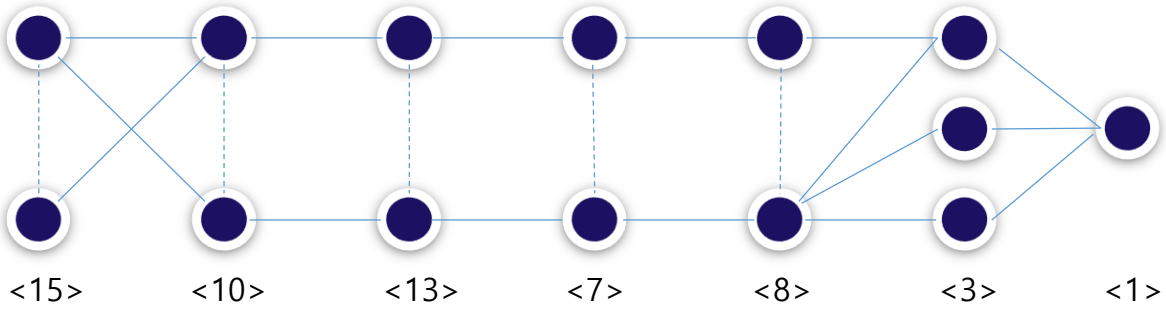
  → $\max_{\beta,\beta_0} \frac{1}{N} \sum_{i=1}^{N} \left(y_i (x_i^\top \beta + \beta_0) - e^{x_i^\top \beta + \beta_0} \right) - \lambda \left(\alpha \|\beta\|_1 + \frac{1}{2}(1-\alpha)\|\beta\|_2^2 \right)$

- 상응하는 분산은 다음과 같다.

  → $D = -2 \sum_{i=1}^{N} (y_i \log(y_i/\hat{y}_i) - (y_i - \hat{y}_i))$

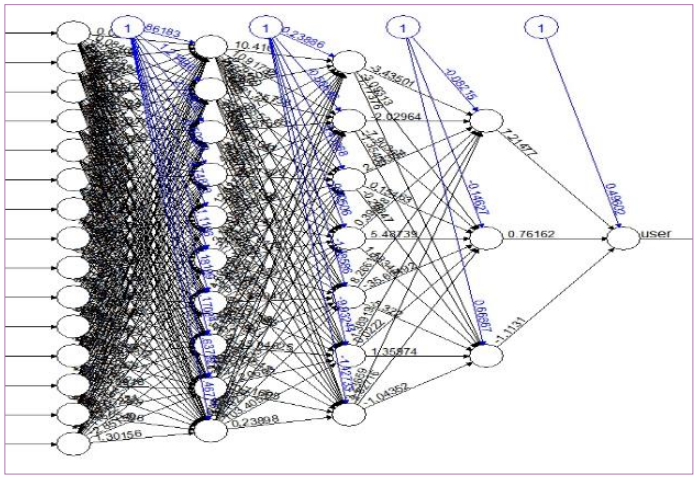
2. Feed-forward network (Using Stochastic Gradient Descent)

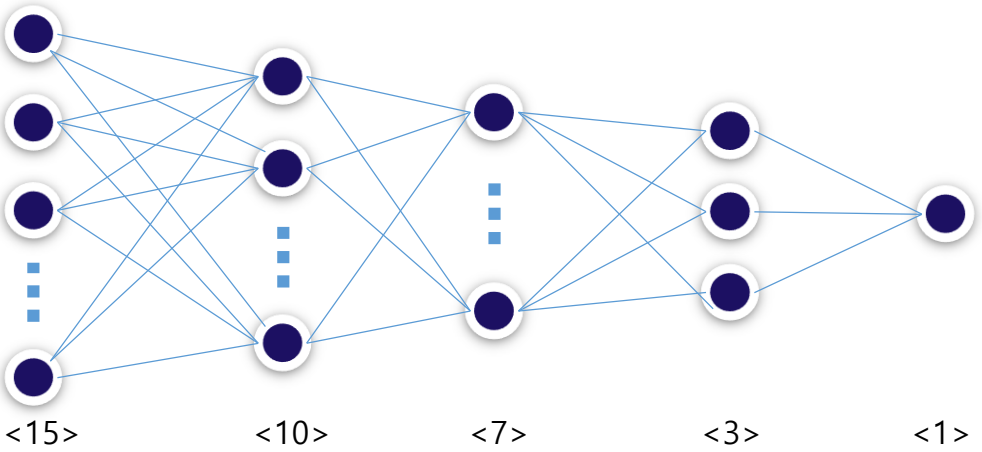




- 활성화 함수: "Tanh" / Supervised learning
- 각각의 학습에서 j에 관하여 "Minimized L(w, β| j)"가 목표
- $W = \{W_i\}$들의 집합 (단, i는 1부터 N-1 까지)
  → W_i= Connecting Layers i and $i+1$ for Network of N layers
- $\beta = \{b_i\}$들의 집합 (단, i는 1부터 N-1 까지)
  → b_i = Column Vector of Biases for layer $i+1$
- Stochastic Gradient Descent를 이용하여 Back Propagation 적용.
- SGD는 Online Gradient Descent 라고도 부른다. SGD는 mini-batch가 1인 경우이다. Batch Gradient Descent보다 Loss-function을 계산하는데 있어 훨씬 효율적이다.

3. Feed-forward network (Using Resilient Backpropagation)





- $$\Delta w_{ij}(t) = \begin{cases} +\Delta_{ij}(t) & , \text{ if } \frac{\partial E}{\partial w_{ij}}(t) > 0 \\ -\Delta_{ij}(t) & , \text{ if } \frac{\partial E}{\partial w_{ij}}(t) < 0 \\ 0 & , \text{ otherwise} \end{cases}$$

- $$\Delta_{ij}(t) = \begin{cases} \eta^{+} \cdot \Delta_{ij}(t-1) & , \text{ if } s_{ij} > 0 \\ \eta^{-} \cdot \Delta_{ij}(t-1) & , \text{ if } s_{ij} < 0 \\ \Delta_{ij}(t-1) & , \text{ otherwise} \end{cases}$$

- $$\text{where } s_{ij} = \frac{\partial E}{\partial w_{ij}}(t-1) \cdot \frac{\partial E}{\partial w_{ij}}(t)$$

Extra (DBM)

Conditional Probabilities (RBM with real-valued input data)

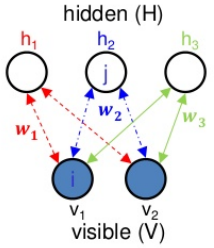
- Given $\mathbf{v}$, all the h_j are conditionally independent

$$P(h_j = 1|\boldsymbol{v}) = \frac{\exp\left(\frac{1}{\sigma}\sum_i W_{ij}v_j + b_j\right)}{\exp\left(\frac{1}{\sigma}\sum_i W_{ij}v_j + b_j\right)+1}$$

$$= \text{sigmoid}\left(\frac{1}{\sigma}\sum_i W_{ij}\, v_j + b_j\right)$$

$$= \text{sigmoid}\left(\frac{1}{\sigma}\boldsymbol{w}_j^T\boldsymbol{v} + b_j\right)$$

  - P(**h**|**v**) can be used as "features"



- Given $\mathbf{h}$, all the v_i are conditionally independent

$P(v_i|\mathbf{h}) = \mathcal{N}\left(\sigma\sum_j W_{ij}\, h_j + c_i, \sigma^2\right)$ or

$P(\mathbf{v}|\mathbf{h}) = \mathcal{N}(\sigma\boldsymbol{W}\mathbf{h} + \mathbf{c}, \sigma^2\mathbf{I})$.

15

Training RBMs

- Model: $P_\theta(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\exp(-E(\mathbf{v},\mathbf{h};\theta))$
- How can we find parameters θ that maximize $P_\theta(\mathbf{v})$?

$$\frac{\partial}{\partial\theta}\log P(\mathbf{v}) = \mathbb{E}_{\mathbf{h}\sim P_\theta(\mathbf{h}|\mathbf{v})}\left[-\frac{\partial}{\partial\theta}E(\mathbf{h},\mathbf{v})\right] - \mathbb{E}_{\mathbf{v}',\mathbf{h}\sim P_\theta(\mathbf{v},\mathbf{h})}\left[-\frac{\partial}{\partial\theta}E(\mathbf{h}',\mathbf{v})\right]$$

Data Distribution (posterior of h given v)

Model Distribution

- We need to compute P(h|v) and P(v,h), and derivative of E wrt parameters {W,b,c}
  - P(h|v): tractable
  - P(v,h): intractable
    - Can approximate with Gibbs sampling, but requires lots of iterations

17

17

Contrastive Divergence

- An approximation of the log-likelihood gradient for RBMs
  1. Replace the average over all possible inputs by samples

$$\frac{\partial}{\partial\theta}\log P(\mathbf{v}) = \mathbb{E}_{\mathbf{h}\sim P_\theta(\mathbf{h}|\mathbf{v})}\left[-\frac{\partial}{\partial\theta}E(\mathbf{h},\mathbf{v})\right] - \mathbb{E}_{\mathbf{v}',\mathbf{h}\sim P_\theta(\mathbf{v},\mathbf{h})}\left[-\frac{\partial}{\partial\theta}E(\mathbf{h}',\mathbf{v})\right]$$

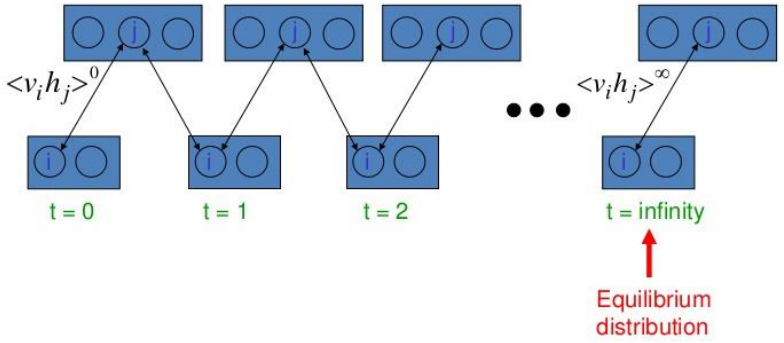
  2. Run the MCMC chain (Gibbs sampling) for only k steps starting from the observed example

```
Initialize with v^0 = v
Sample h^0 from P(h|v^0)

For t = 1,...,k {
        Sample v^t from P(v^t|h^{t-1})
        Sample h^t from P(h|v^t)
}
```

18

18

A picture of the maximum likelihood learning algorithm for an RBM



Slide Credit: Geoff Hinton

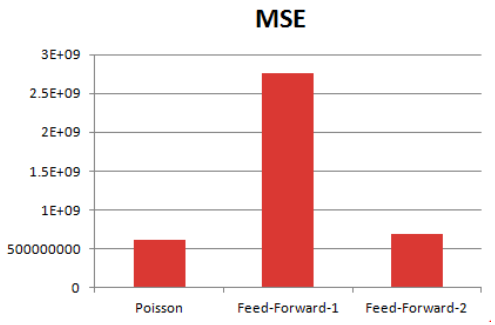
19

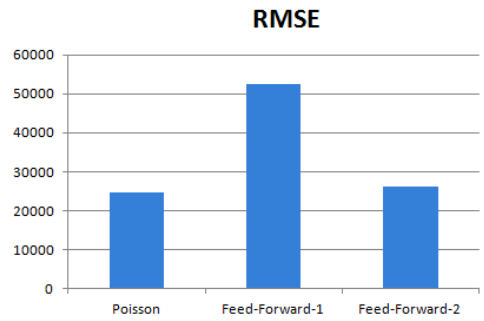
※ 성능 평가

- $Y = \{Y_1, Y_2, \cdots, Y_n\},\ \hat{Y} = \{\hat{Y}_1, \hat{Y}_2, \cdots, \hat{Y}_n\},\ \hat{Y}_i = f(x_1, \cdots, x_n)$

$$\text{MSE} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \times \frac{1}{n}$$

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \times \frac{1}{n}}$$



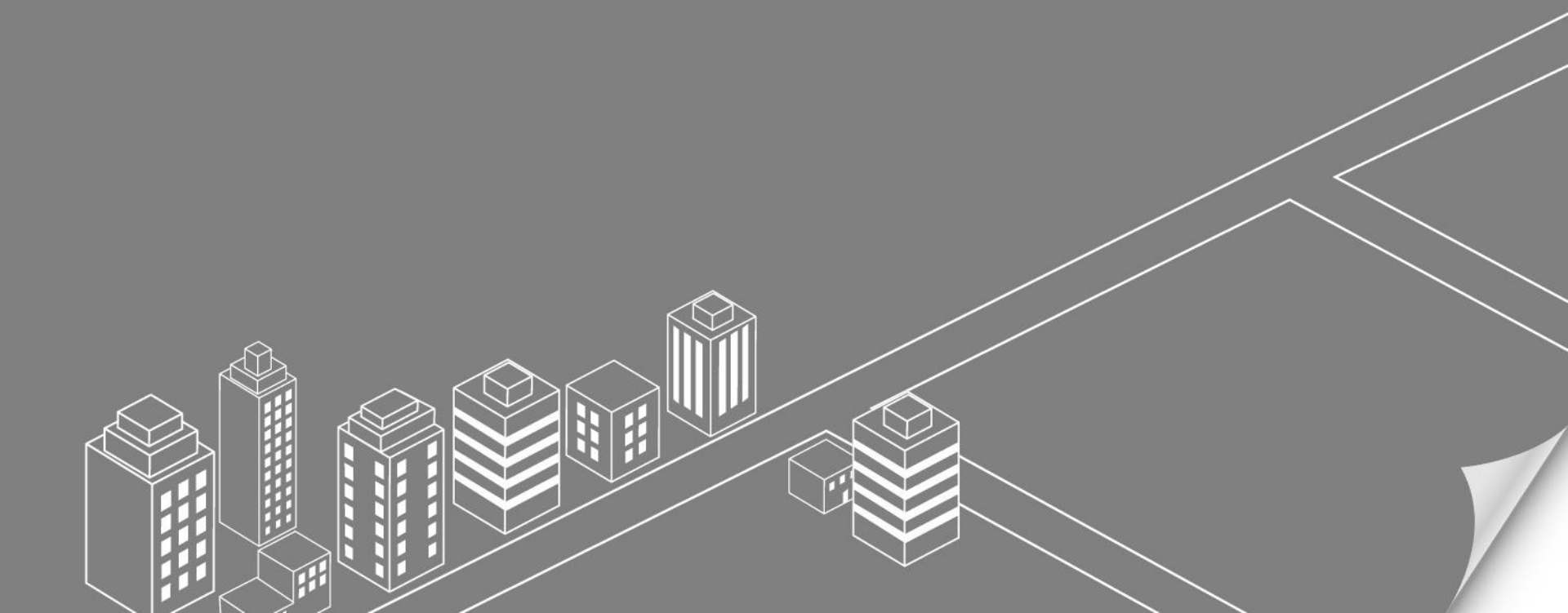


	Poisson model	Feed-Forward [Resilient Propagation]	Feed-Forward [Stochastic Gradient Descent]
MSE	608868882	2761791337	683040948
RMSE	24675.27	52552.75	26135.05

결론

- 알고리즘의 성능 향상을 위해 대중교통 이용 관련 데이터의 추가적인 수집 및 분석 필요성이 있음.
- 시계열 데이터 분석을 통해 계절적 요인을 고려한 모델 개발 필요성이 있음.
- 약한 예측력을 가진 신경망 모델들의 효과적인 앙상블 모형인 EENCL (Evolutionary Ensembles with Negative Correlation Learning) 또는 CNNE(Cooperative Neural Network Ensembles)의 적용이 필요함

V. 출처



V. 출처

구 분	내 용			
데이터	지하철			
출처	https://www.data.go.kr/ https://www.open.go.kr/			
기관	공공데이터포털 및 정보공개포털			
데이터	전국 상가			
분류	8개 항목 (음식, 생활서비스, 소매 등)			
출처	http://www.semas.or.kr/web/main/index.kmdc			
기관	소상공인시흥진흥공단			
데이터	CCTV			
출처	https://www.data.go.kr/dataset/15013094/standard.do			
기관	공공데이터포털			
데이터	도서관	초중고	대학	공원
출처	https://www.data.go.kr/dataset/15013109/standard.do	https://www.data.go.kr/dataset/15021148/standard.do	https://www.data.go.kr/dataset/3038867/fileData.do	https://www.data.go.kr/dataset/15012890/standard.do
기관	공공데이터포털			
데이터	부동산 실거래가 (아파트)			
출처	http://www.semas.or.kr/web/main/index.kmdc			
기관	소상공인시흥진흥공단			

▶ 참고 데이터

서울교통공사
대구도시철도공사
부산교통공사
대전광역시도시철도공사
광주광역시도시철도공사
경기도 용인시
경기도 의정부시
소상공인시장진흥공단
국토교통부 실거래가 공개시스템

V. 출처

▶ 참조

Learning internal representations by error propagation / David E.Rumelhart,Geoffrey E.Hinton, Ronald j.Williams / 1985
Learning Deep Generative Models / Ruslan Salakhutdinov / 2009
Classification of Sets Using Restricted Boltzmann Machines / Jerome Louradour, Hugo Larochelle
Classification and Segmentation of Satellite Orthoimagery Using Convolutional Neural Networks
Ensemble Approachs for Regression:a survey / Joao M.Moreira, Carlos Soares, Alipio M. Jorge, Jorge Freire de Sousa / 2007
Neural Network Ensembles, Cross Validation, and Active Learning / Anders Krogh, Jesper Vedelsby
R and Data mining / Yanchang Zhao / Academic Press / 2012

https://en.wikipedia.org/wiki/Ensemble_averaging_(machine_learning)
http://www.endmemo.com/program/R/lm.php
https://www.r-bloggers.com/simple-linear-regression-2/
Generalized Linear Models / J.A.Nelder, R.W.M. WedderBurn
https://en.wikipedia.org/wiki/Generalized_linear_model
http://www.learnbymarketing.com/tutorials/neural-networks-in-r-tutorial/
https://www.r-bloggers.com/fitting-a-neural-network-in-r-neuralnet-package/
https://en.wikipedia.org/wiki/Rprop
http://aass.oru.se/~lilien/ml/seminars/2007_03_12c-Markus_Ingvarsson-RPROP.pdf
https://github.com/maddin79/darch/issues/5
http://www.rblog.uni-freiburg.de/2017/02/07/deep-learning-in-r/
https://cran.r-project.org/web/packages/darch/darch.pdf
https://cran.r-project.org/web/packages/deepnet/deepnet.pdf
https://www.r-bloggers.com/deep-learning-in-r-2/
http://r-bong.blogspot.kr/2016/11/neuralnet.html
https://github.com/h2oai/h2o-tutorials/blob/master/tutorials/deeplearning/H2ODeepLearning.pdf
https://github.com/h2oai/h2o-tutorials/tree/master/tutorials/deeplearning
https://stackoverflow.com/questions/43324287/h2o-glm-does-not-match-glm-in-r-for-linear-regressions
https://rdrr.io/cran/h2o/man/h2o.scale.html
http://www.rblog.uni-freiburg.de/2017/02/07/deep-learning-in-r/
http://docs.h2o.ai/h2o/latest-stable/h2o-docs/booklets/DeepLearningBooklet.pdf
http://solarisailab.com/archives/113

THANK YOU FOR WATCHING