

# 기업의 도산 가능성 분석 및 예측

(중소기업 데이터 활용)

학      기  
강      의  
지      도 교  
조      수  
조      장  
         원

2020-1  
산학 캡스톤디자인-1  
정 현 숙  
임 형 열  
김 동 진  
조 재 혁  
김 태 완

발 표 일 자

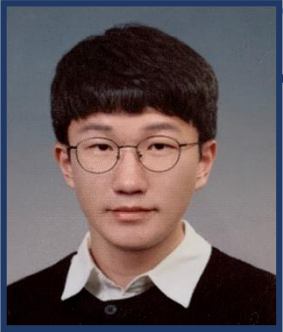
2020. 4. 22. (수)

# 목 차

---

- 1. 소 개
- 2. 목 적
- 3. 개 발 환 경
- 4. 진 행 상 황
- 5. 의 견
- 6. 참 고 자 료

# 1. 소개



**임형열**

컴퓨터공학과 13  
@cuti2crab

Big Data, Python



**김동진**

컴퓨터공학과 14  
@dori.yeni

Big Data, JAVA



**조재혁**

컴퓨터공학과 14  
@jhst\_c

Big Data, Open CV, R



**김태완**

컴퓨터공학과 14  
@tae.\_.wan

Big Data, Web Programming

## 2. 목 적

### '기업의 도산 가능성 분석 및 예측'

#### 코로나 19

- 단기 유동성 문제 등
- 도산 기업의 증가

#### 관련 정책

- 정부의 기업 맞춤형 정책 부족

#### 기업과 투자자

- 문제 직관적 파악 어려움
- 기업에 투자해도 괜찮을지?

데이터 분석 & 시각화 - 문제를 직관적으로 파악  
기존 값을 활용한 미래 예측 - 정책 수립, 컨설팅 자료로 활용

# 3. 개발 환경

Web Framework (Django)  
- 웹 개발 : 설계 & 구현  
(일반 사용자가 사용하기 쉽도록..)

```
#Adj Close : 마감가를 이동하여 히스토그램 작성  
sam_g = df_sam["Adj Close"]##삼성전자의 평균 마감가  
ax1 = sam_g.plot()  
lg_g = df_lg["Adj Close"]##lg전자의 평균마감가  
ax2 = lg_g.plot(grid=True)  
plt.show()
```



Python 3.7 (Jupyter Notebook)  
- 데이터 분석, 시각화  
(데이터 전처리 과정 필요)

# 4. 진행 상황

## 데이터셋 소개, 특성 선택

항목번호	조사항목 내용	항목설명
global_id	-	조사간 통합 연계키
x1	지역	11: 서울 / 21: 부산 / 22: 대구 / 23: 인천 / 24: 광주 / 25: 대전 / 26: 울산 / 30: 경기북부 / 31: 경기남부 / 32: 강원 / 33: 충북 / 34: 충남 / 35: 전북 / 36: 전남 / 37: 경북 / 38: 경남 / 39: 제주
x2	규모	1규모 ~ 6규모
x3	산업분류	
x16	기업유형	1: 일반기업 / 2: 벤처,이노비즈,경영혁신기업
x17	수출여부	1: 하고있다/2: 하고있지 않다
x17_1	대기업납품여부	1: 하고있다/2: 하고있지 않다
x20	경기실적	1: 매우나쁨 / 2: 다소나쁨 / 3: 보통 / 4: 다소좋음 / 5: 매우좋음



x66	원자재(원재료)구독난	0: 미선택 / 1: 선택
x67	설비노후 및 부족	0: 미선택 / 1: 선택
x68	계절적비수기	0: 미선택 / 1: 선택
x69	환율변동	0: 미선택 / 1: 선택
x70	고금리	0: 미선택 / 1: 선택
x71	기업(대기업)과의 불공정거래	0: 미선택 / 1: 선택

	항목번호	percentage
기업코드	global_id	
1.자금사정실적	X38	25.0%
2.내수전망	X33	15.0%
3.판매대금회수지연	X57	11.0%
4.자금조달곤란	X58	9.0%
5.영업이익실적	X36	8.5%
6.업체간 과당경쟁	X59	7.5%
7.경기전망	X21	7.0%
8.인력확보난	X60	6.5%
9.인건비상승	X61	6.0%
10.수출전망	X35	3.5%

## 4. 진행 상황

### 특성 추출

전문가 자문을 통한 주요 요소를 추출

X1 X2 X3 X16					X69 X70 X71			X16 X21 X33 X35 X36 X38 X57 X58 X59 X60 X61												
global_id								global_id												
134160	24	1	47	1	0	0	0	134160	1	1	1	0	1	1	0	1	0	0	0	1
134161	24	3	75	1	0	0	0	134161	1	2	2	0	2	2	1	0	0	0	0	0
134163	24	3	41	1	0	0	0	134163	1	3	2	0	2	3	0	0	1	0	1	1
134164	24	2	75	1	0	1	0	134164	1	3	3	0	3	3	0	0	0	0	0	1
126648	24	2	62	2	0	0	0	126648	2	2	2	0	1	2	0	0	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
118866	36	2	11	1	0	1	0	118866	1	3	3	3	3	3	0	1	0	0	0	0
135633	36	1	22	1	0	0	0	135633	1	3	3	0	4	4	0	1	1	0	0	0
135950	36	5	52	2	0	0	0	135950	2	3	3	0	2	2	0	0	0	0	0	0
135921	36	3	74	1	0	0	0	135921	1	3	3	0	3	3	0	0	1	0	0	0

## 4. 진행 상황

### 월 별 존재하는 기업 카운트

총 개월 수 중 기준 개수 이상 존재하는 기업 갯수

```
In [359]: tmp = val_cnt >= 6
          tmp_cnt = val_cnt.loc[tmp]
          tmp_cnt
          #우리가 사용할 값들을 나열했다.
```

```
Out [359]: 118756    11
           134733    11
           134611    11
           134389    11
           134383    11
           ..
           113092     8
           118726     8
           134347     8
           134317     7
           135626     6
           Length: 199, dtype: int64
```

6개 이상



```
In [355]: tmp = val_cnt >= 8
          tmp_cnt = val_cnt.loc[tmp]
          tmp_cnt
          #우리가 사용할 값들을 나열했다.
```

```
Out [355]: 118756    11
           134733    11
           134611    11
           134389    11
           134383    11
           ..
           136205     9
           134166     8
           113092     8
           118726     8
           134347     8
           Length: 197, dtype: int64
```

8개 이상

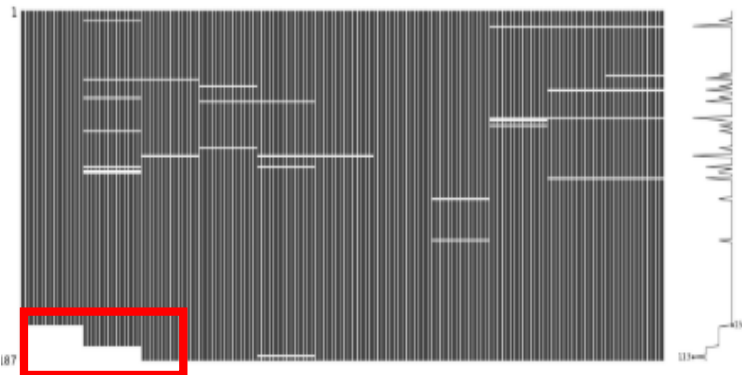


# 4. 진행 상황

## 데이터 뷰

0	127117	2.0	2.0	2.0	0.0	3.0	2.0	0.0	1.0	1.0	...	2.0	0.0	1.0
1	134161	1.0	2.0	2.0	0.0	2.0	2.0	1.0	0.0	0.0	...	3.0	0.0	2.0
2	126988	1.0	3.0	2.0	0.0	2.0	2.0	0.0	1.0	0.0	...	2.0	0.0	2.0
3	134162	1.0	3.0	3.0	0.0	3.0	3.0	0.0	0.0	0.0	...	1.0	0.0	1.0
4	134163	1.0	3.0	2.0	0.0	2.0	3.0	0.0	0.0	1.0	...	3.0	0.0	3.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
182	13620	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	2.0	0.0	1.0
183	13620	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	1.0	1.0	1.0
184	11309	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	3.0	0.0	2.0
185	13621	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	3.0	4.0	3.0
186	13621	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	3.0	0.0	2.0

187 rows x 133 columns



```
result9.set_index('월')
```

	global_id	기업유형	경기전망	내수전망	수출전망	영업이익실적	자금사정실적	판매대금회수지연	자금조달곤란	업체간과당경쟁	인력확보난	인건비상승
월												
2	127117	2	2	2	0	3	2	0	1	1	0	0
2	134161	1	2	2	0	2	2	1	0	0	0	0
2	126988	1	3	2	0	2	2	0	1	0	1	1
2	134162	1	3	3	0	3	3	0	0	0	0	1
2	134163	1	3	2	0	2	3	0	0	1	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...
12	118866	1	3	3	3	3	3	0	1	0	0	0
12	135633	1	3	3	0	4	4	0	1	1	0	0
12	135950	2	3	3	0	2	2	0	0	0	0	0
12	135921	1	3	3	0	3	3	0	0	1	0	0
12	135133	1	3	3	0	3	3	0	0	0	0	1

## 5. 의 견

### 분기 기준

각 분기에 결측치 존재시  
해당 분기의 평균을 이용

### 전월 / 익월 기준

특정 월 데이터가 결측치일 경우  
해당월의 전월과 익월의 평균을 이용

## '결측치'

### 첫 번째 방식을 보완

각 분기를 그룹화하여 평균값으로 대체  
분기를 나누는 기준 생각하기

## 6. 참고자료

---

파이썬으로 데이터 주무르기 - 민형기, 비제이퍼블릭

파이썬 라이브러리를 활용한 데이터 분석 - 웨스 맥키니, 한빛미디어

Django로 배우는 쉽고 빠른 웹 개발 : 파이썬 웹 프로그래밍 - 김석훈, 한빛미디어