머신러닝을 활용한 제주노지감귤 비상품량 예측

**Forcasting of Jeju orange by machine learning**

2019312048 유새하

김효정

이관구

이상협

홍형근

**목차**

1. 연구과제의 개요
2. 연구 목적
3. 감귤

1) 감귤의 분류

2) 비상품의 정의

3) 노지온주와 날씨의 상관관계

1. 연구 과제 수행내용
2. 연구 내용
3. 실패코드
4. 코드
5. 결과
6. 연구 수행 일정
7. 연구 과제 수행결과 및 성과
8. 출처
9. 연구 결과 및 성과
10. 기대 효과 및 연구 활용 계획
11. 연구 프로젝트 멘토링 결과

**Abstract**

머신러닝을 활용하여 제주노지감귤 비상품량을 예측해보았다. 풍년임에도 과잉공급된 감귤은 가격하락으로 많은 생산량에도 불구하고 손해를 가져온다. 하지만 이를 잘 예측해낸다면 흉년일지라도 농가에 이익을 극대화할 수 있다. 따라서, 날씨에 따른 수확량을 예측하고 이를 비상품량에 적용한다면 수확량과 관계없이 농가의 이익을 최대화할 수 있다. 기상청 날씨데이터 개방포털의 날씨데이터를 활용하여 기상을 예측했고, 통계적 방법으로는 선형회귀분석을 적용한 Tensorflow RNN 툴을 사용하여 예측해보았다. 그 후, RMSE를 적용하여 분석의 유효성을 확인했다.

**ⅰ. 연구과제의 개요**

1. 연구목적

농업산업의 활성화를 위해서는 농산품의 가격을 적절하게 유지하여 시장을 안정시키는 것이 상당히 중요하다. 풍년이 들어 생산량이 너무 많아진다면, 공급과잉에 의하여 농산품 가격이 하락하여 오히려 농가에 손해를 가져올 수 있다. 반면, 흉년임에도 불구하고 가격 정책을 잘 유지한다면, 농가에 많은 이익이 발생 할 수 있다. 이러한 요인들을 통제하기 위해서 재배한 모든 농산품을 시장에 내놓는 것이 아니라, 상대적으로 낮은 품질의 농산품들은 비상품으로 분류하여 시장에서 격리시키는 등의 가격유지 정책을 활용하게 된다. 이렇듯, 시장에 내놓는 농산품의 양을 조절하는 것은 상당히 중요하다. 노지온주는 매해 날씨에 영향을 많이 받아, 날씨에 따른 노지온주의 비상품량 예측을 해보고자 한다. 본 프로젝트의 목적은 감귤의 비상품량을 예측하여 감귤농가의 이익증진에 활용되고자 한다.

본 프로젝트에서는 제주농산품에서 가장 많은 비중을 차지하는 감귤산업에 초점을 맞추는데 특히, 다른 품종과 달리 매해 날씨에 영향을 많이 받고 감귤생산량에서 가장 많은 비중을 차지하는 노지온주를 분석한다. 현재 제주 감귤 산업에서는 비상품 감귤 처리가 큰 문제로 자리잡고 있다. 2017년 비상품 감귤량은 예년보다 훨씬 적었지만 비상품 감귤 처리난은 여전히 마찬가지일 뿐만 아니라 오히려 더 심한 상황이었다. 제주 농협 관계자는 그 원인이 비상품 감귤을 가공용으로 수매하는 공장을 늦게 가동한 것에 있다고 지적하였다. 따라서 본 프로젝트의 목적은 날씨에 따른 노지온주의 비상품량을 예측하여 감귤농가와 제주 감귤 산업의 이익 증진에 활용되고자 한다.

1. 감귤
2. 감귤의 분류

감귤은 품종을 기준으로 온주밀감과 만감류로 구분된다.

온주밀감은 일남1호, 유라조생, 궁천조생, 홍진조생등의 품종이 있으며, 만감류는 온주밀감을 제외한 다른 감귤류들을 통틀어 부르는 말로, 한라봉, 천혜향, 레드향등의 품종이 있다. 온주밀감은 재배장소를 기준으로 노지온주와 하우스온주로 구분된다. 노지에서 자란 온주밀감을 노지온주, 하우스에서 자란 온주밀감을 하우스온주라고 일컫는다. 온주밀감을 익는시기를 기준으로 구별하기도 한다. 10월에 익는 극조생온주밀감과 11월에 익는 조생온주밀감으로 구별된다.

1. 비상품의 정의

감귤의 품질과 생산량에 따라 감귤의 시장 가격이 급변하는 것을 막기 위해 저품질 감귤의 일부를 비상품으로 구분해 처리한다. 제주특별자치도 감귤생산 및 유통에 관한 조례 제2조 제3호에 따르면 ‘비상품 감귤’은 제21조에 규정에 의해 감귤의 상품기준에 해당하지 않는 감귤을 말한다. 제주특별자치도 감귤생산 및 유통에 관한 조례 시행규칙에 의하면 제18조 제4항 상품감귤 품질기준은 다음과 같다.

|  |  |
| --- | --- |
| 구 분 | 상 품 기 준 |
| 과일의 크기 | o 온주밀감의 상품 규격은 과실 크기가 횡경 49밀리미터 이상 71밀리미터 미만 또는 감귤 1과의 무게가 53그램 이상 136그램 미만인 밀감   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 구 분 | **2S** | **S** | **M** | **L** | **2L** | | 규격  (밀리미터) | 49이상  ~54미만 | 54이상  ~59미만 | 59이상  ~63미만 | 63이상  ~67미만 | 67이상  ~71미만 | | 무게  (그램) | 53이상  ~63미만 | 63이상  ~83미만 | 83이상  ~107미만 | 107이상  ~124미만 | 124이상  ~136미만 |   o 위 상품 규격 기준에도 불구하고 당도 10브릭스(°Bx)이상인 온주밀감 중 감귤출하연합회와 협의하여 도지사가 고시한 온주밀감은 상품으로 본다. |
| 당 도 | o 하우스재배 온주밀감과 월동비가림 온주밀감은 당도 10브릭스(°Bx) 이상  o 극조생 온주밀감은당도 8브릭스(°Bx) 이상  o 조생 및 보통 온주밀감은 당도 9브릭스(°Bx) 이상  o 단, 노지 온주밀감의 당도검사는 11월 15일 이전 검사품에 한한다. |
| 껍질뜬것 | o 육안으로 보아 껍질의 뜬 정도가 껍질 내표 면적의 50퍼센트 미만(중간 정도 이하)인 것 |
| 결점과의 정 도 | o 중결점과에 해당되는 것이 섞여 있지 않을 것 |

[ 표1 <http://citrus.or.kr/board/view.php?btable=bbs&bno=253&p=1&cate=0> ]

온주밀감은 크기, 당도, 껍질이 뜬 정도, 결점과 정도를 기준으로 상품 판정을 하는데 비상품 기준은 이러한 상품 기준 이외의 것을 말한다. 고품질 감귤생산과 가격안정 도모를 위해 시장유통이 차단된 비상품 감귤은 가공용으로 수매되거나 제주도 농민 자체적으로 농가에 감귤을 자가격리한다.

1. 노지온주와 날씨의 상관관계

감귤 생육 정보는 다음과 같다.

4월 초~중순 : 발아기

5월 초~중순 : 개화기

5월 중순~7월 중순 : 생리낙과 \*열매랑 꼭지 같이 떨어지는 1차 낙과, 열매만은 2차 낙과

7월 중순~9월 초순 : 과실비대기

9월 중순~11월 중순 : 성숙기

9월 초순~11월 하순 : 생리적 화아분화

2월 초순~3월 중순 : 형태적 화아분화

-감귤수확

10월 : 극조생 수확

11월 : 조생 수확

본 프로젝트는 노지온주의 품질에 가장 영향을 많이 끼치는 개화기 ~ 과실비대기를 중심으로 분석한다.

또한, 가장 크기 및 당도에 영향을 많이주는 요인인 강수량, 일조량, 기온, 풍속, 습도, 기압, 지면온도를 변수로 설정하였다.

**ⅱ. 연구과제 수행내용**

1. 연구 내용

0) 실패코드

|  |  |
| --- | --- |
|  | 전처리 과정 |
| 8 | from datetime import datetime  for year in range(1994,2022):    str = "weatherAverage{0} = weather\_MayToSep\_{0}.resample(rule='M').mean()".format(year)    exec(str)    str = "weatherAverage{0} = weatherAverage{0}.iloc[:-1,:]".format(year)    exec(str)  weatherAverage2001 |

초기 학습데이터의 구성은 시간별, 날짜별 날씨를 모두 각각의 변수로 설정하였다. 년도별로 5월부터 9월까지 날짜수 \* 24시간\* 날씨변수 7가지 = 25711 개의 변수와 그에 따른 비상품량을 y값으로 설정하였다. 그 결과, 20000개 이상의 많은 변수에 비하여 27개의 적은 feature 의 크기로 인하여 train error 값은 잘 도출이 되지만, test error가 커지는 결과가 나왔다. 이러한 과적합 현상이 발생하는 것을 막기 위하여 위의 8번코드를 실행하여 월별 평균값을 변수로 설정하여, x값을 연도당 35개로 대폭 축소하였다.

그래서 위 코드를 삽입하였다.

1) 코드

|  |  |
| --- | --- |
|  | 사용할 패키지 import |
| 1 | from sklearn.linear\_model import LinearRegression  from sklearn.model\_selection import train\_test\_split  from sklearn.model\_selection import KFold, cross\_validate  from sklearn.preprocessing import StandardScaler  from sklearn.metrics import mean\_squared\_error, r2\_score  import numpy as np  import pandas as pd |

|  |  |
| --- | --- |
|  | 비상품량 업로드 |
| 2 | from google.colab import files  uploaded = files.upload()  #2020~1994 가공품량  y = pd.read\_excel('비상품량.xlsx').iloc[:,-1]  # 2020~1994 생산량  #y=[515778, 491149, 467600, 440254, 466817, 519243, 573442, 554007,558942, 500106, 480565, 655046, 520350, 677770, 568920, 600511, 536668, 597373, 739266, 600140, 518731, 593188, 510644, 659121, 416557, 508445, 548945]  y |

|  |  |
| --- | --- |
|  | 날씨 엑셀파일 업로드 및 전처리 |
| 3 | for year in range(1994,2022):    str = "weather\_row{0} = pd.read\_excel('{0}weather.xlsx')".format(year)    exec(str)  '''  weather\_row2010 = pd.read\_excel('2010weather.xlsx')  weather\_row2011 = pd.read\_excel('2011weather.xlsx')  weather\_row2012 = pd.read\_excel('2012weather.xlsx')  weather\_row2013 = pd.read\_excel('2013weather.xlsx')  weather\_row2014 = pd.read\_excel('2014weather.xlsx')  weather\_row2015 = pd.read\_excel('2015weather.xlsx')  weather\_row2016 = pd.read\_excel('2016weather.xlsx')  weather\_row2017 = pd.read\_excel('2017weather.xlsx')  weather\_row2018 = pd.read\_excel('2018weather.xlsx')  weather\_row2019 = pd.read\_excel('2019weather.xlsx')  weather\_row2020 = pd.read\_excel('2020weather.xlsx')  weather\_row2021 = pd.read\_excel('2021weather.xlsx')  '''  weather\_row2010.head(5) |
| 4 | #필요없는 행, 열 제거  #서귀포 지점에 해당하는 데이터만, 변수값은 임의로 기온, 강수량, 풍속, 습도, 현지기압, 일조, 지면온도 으로 설정하여 추출함.  for year in range(1994,2022):    str1 = "weather\_row{0}\_extract = weather\_row{1}[weather\_row{0}['지점']==189]".format(year,year)    str2 = "weather\_row{0}\_extract = weather\_row{1}\_extract.iloc[:,[2,3,5,7,11,15,19,31]]".format(year,year)    exec(str1)    exec(str2) |
| 5 | # 5,6,7,8,9월 데이터만 가져옴.  for year in range(1994,2022):    str\_temp = "weather\_MayToSep\_{0}\_temp = weather\_row{0}\_extract['{0}-04-30 23:00:00'<weather\_row{0}\_extract['일시']]".format(year)    str = "weather\_MayToSep\_{0} = weather\_MayToSep\_{0}\_temp[weather\_row{0}\_extract['일시']<'{0}-10-01 01:00:00']".format(year)    exec(str\_temp)    exec(str)    weather\_MayToSep\_1996 |
| 6 | #nan 을 0으로 바꾸기  for year in range(1994,2022):      str = "weather\_MayToSep\_{0}.loc[np.isnan(weather\_MayToSep\_{0}['강수량(mm)']), '강수량(mm)'] = 0".format(year)    exec(str)    str = "weather\_MayToSep\_{0}.loc[np.isnan(weather\_MayToSep\_{0}['일조(hr)']), '일조(hr)'] = 0".format(year)    exec(str)    str = "weather\_MayToSep\_{0}.loc[np.isnan(weather\_MayToSep\_{0}['기온(°C)']), '기온(°C)'] = 0".format(year)    exec(str)    str = "weather\_MayToSep\_{0}.loc[np.isnan(weather\_MayToSep\_{0}['풍속(m/s)']), '풍속(m/s)'] = 0".format(year)    exec(str)    str = "weather\_MayToSep\_{0}.loc[np.isnan(weather\_MayToSep\_{0}['습도(%)']), '습도(%)'] = 0".format(year)    exec(str)    str = "weather\_MayToSep\_{0}.loc[np.isnan(weather\_MayToSep\_{0}['현지기압(hPa)']), '현지기압(hPa)'] = 0".format(year)    exec(str)    str = "weather\_MayToSep\_{0}.loc[np.isnan(weather\_MayToSep\_{0}['지면온도(°C)']), '지면온도(°C)'] = 0".format(year)    exec(str) |
| 7 | for year in range(1994,2022):    str = "weather\_MayToSep\_{0} = weather\_MayToSep\_{0}.reset\_index(drop=True)".format(year)    exec(str)    str = "weather\_MayToSep\_{0} = weather\_MayToSep\_{0}.set\_index('일시')".format(year)    exec(str)  weather\_MayToSep\_2001 |
| 8 | from datetime import datetime  for year in range(1994,2022):    str = "weatherAverage{0} = weather\_MayToSep\_{0}.resample(rule='M').mean()".format(year)    exec(str)    str = "weatherAverage{0} = weatherAverage{0}.iloc[:-1,:]".format(year)    exec(str)  weatherAverage2001 |
| 9 | #5개월치 날씨 DataFrame 안의 모든 데이터를 1차원 리스트로 변환.  for year in range(1994,2022):    #빈 리스트 형성    makelist\_str = 'x\_list\_{0} = list()'.format(year)    exec(makelist\_str)    for j in range(0,weatherAverage2010.shape[1]):      str = 'x\_list\_{0}.extend(list(weatherAverage{0}.iloc[:,j]))'.format(year)      exec(str)  x\_list\_2001 |
| 10 | #연도별 1차원 리스트들을 합하여 2차원 리스트로 변환.  x\_list = list()  for year in range(2020,1993,-1):    str = 'x\_list.append(x\_list\_{0})'.format(year)    exec(str)  len(x\_list)  x\_list  #길이체크  len(x\_list[0]) |

|  |  |
| --- | --- |
|  | 모델 실행 및 결과 확인 |
| 11 | y\_test  #선형회귀  X\_train, X\_test, y\_train, y\_test = train\_test\_split(x\_list, y, test\_size = 0.03)  scaler = StandardScaler()  scaler.fit(X\_train)  X\_train\_scaled = scaler.transform(X\_train)  X\_test\_scaled = scaler.transform(X\_test)  reg = LinearRegression()  reg.fit(X\_train\_scaled,y\_train)  y\_train\_hat = reg.predict(X\_train\_scaled)  y\_test\_hat = reg.predict(X\_test\_scaled)  print('RMSE')  print('train: ',mean\_squared\_error(y\_train,y\_train\_hat)\*0.5,'test: ',mean\_squared\_error(y\_test,y\_test\_hat)\*0.5)  print('R\_Squared')  print('train: ',r2\_score(y\_train,y\_train\_hat)\*0.5,'test: ',r2\_score(y\_test,y\_test\_hat)\*0.5) |
| 12 | #릿지  from sklearn.linear\_model import Ridge  ridge = Ridge(alpha=0.001)  ridge.fit(X\_train\_scaled,y\_train)  y\_train\_hat = ridge.predict(X\_train\_scaled)  y\_test\_hat = ridge.predict(X\_test\_scaled)  print('RMSE')  print('train: ',mean\_squared\_error(y\_train,y\_train\_hat)\*0.5,'test: ',mean\_squared\_error(y\_test,y\_test\_hat)\*0.5)  print('R\_Squared')  print('train: ',r2\_score(y\_train,y\_train\_hat)\*0.5,'test: ',r2\_score(y\_test,y\_test\_hat)\*0.5) |
| 13 | #라쏘  from sklearn.linear\_model import Lasso  lasso = Lasso(alpha=1)  lasso.fit(X\_train\_scaled,y\_train)  y\_train\_hat = lasso.predict(X\_train\_scaled)  y\_test\_hat = lasso.predict(X\_test\_scaled)  print('RMSE')  print('train: ',mean\_squared\_error(y\_train,y\_train\_hat)\*0.5,'test: ',mean\_squared\_error(y\_test,y\_test\_hat)\*0.5)  print('R\_Squared')  print('train: ',r2\_score(y\_train,y\_train\_hat)\*0.5,'test: ',r2\_score(y\_test,y\_test\_hat)\*0.5) |
| 14 | #뉴럴 네트워크  from sklearn.neural\_network import MLPRegressor  mlp = MLPRegressor()  mlp.fit(X\_train\_scaled,y\_train)  y\_train\_hat = mlp.predict(X\_train\_scaled)  y\_test\_hat = mlp.predict(X\_test\_scaled)  print('RMSE')  print('train: ',mean\_squared\_error(y\_train,y\_train\_hat)\*0.5,'test: ',mean\_squared\_error(y\_test,y\_test\_hat)\*0.5)  print('R\_Squared')  print('train: ',r2\_score(y\_train,y\_train\_hat)\*0.5,'test: ',r2\_score(y\_test,y\_test\_hat)\*0.5) |
| 15 | from sklearn.linear\_model import LinearRegression, Ridge, Lasso  from sklearn.neighbors import KNeighborsRegressor  from sklearn.ensemble import RandomForestRegressor  from sklearn.svm import SVR  from sklearn.neural\_network import MLPRegressor  from sklearn.model\_selection import train\_test\_split  from sklearn.model\_selection import KFold, GridSearchCV  from sklearn.preprocessing import StandardScaler, MinMaxScaler  from sklearn.metrics import mean\_squared\_error, r2\_score  from sklearn.pipeline import Pipeline |
| 16 | pipe = Pipeline([('preprocessing',None),('regressor',LinearRegression())])  hyperparam\_grid = [                     {'regressor': [LinearRegression(),Ridge(),Lasso()],'preprocessing':[StandardScaler(),MinMaxScaler(),None]},                     {'regressor': [KNeighborsRegressor()],'preprocessing':[StandardScaler(),MinMaxScaler(),None],                      'regressor\_\_n\_neighbors':[1,2,3,4,5],'regressor\_\_weights':['uniform','distance'],'regressor\_\_p':[1,2,3]},                     {'regressor': [RandomForestRegressor()],'preprocessing':[StandardScaler(),MinMaxScaler(),None],                      'regressor\_\_max\_features':['auto','sqrt','log2']},                     {'regressor': [SVR()],'preprocessing':[StandardScaler(),MinMaxScaler(),None],                      'regressor\_\_epsilon':[0,0.1,0.01,0.001],'regressor\_\_gamma':[0.001,0.01,0.1],'regressor\_\_C':[1,10,100]},                     {'regressor': [MLPRegressor()],'preprocessing':[StandardScaler(),MinMaxScaler(),None]}  ]  kfold = KFold(n\_splits=27)  grid = GridSearchCV(pipe, hyperparam\_grid,scoring='r2',cv=kfold)  grid.fit(x\_list,y)  print(grid.best\_params\_)  print(grid.best\_estimator\_) |

2) 결과

|  |  |  |
| --- | --- | --- |
| 1 | 사용할 패키지 import | |
| 2 | 비상품량 upload | |
| 3 |  | |
| 4 |  | |
| 5 |  | |
| 6 | /usr/local/lib/python3.7/dist-packages/pandas/core/indexing.py:1763: SettingWithCopyWarning:  A value is trying to be set on a copy of a slice from a DataFrame.  Try using .loc[row\_indexer,col\_indexer] = value instead  See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy>  isetter(loc, value) | |
| 7 |  | |
| 8 |  | |
| 9 | [19.77620967741939,  22.238055555555576,  27.410483870967752,  27.93024193548384,  24.420694444444468,  0.21841397849462346,  0.3775,  0.3672043010752688,  0.325268817204301,  0.10333333333333335,  2.0879032258064507,  2.8700000000000006,  2.306451612903222,  3.1323924731182773,  4.203888888888887,  70.96774193548387,  78.58888888888889,  81.74596774193549,  71.91801075268818,  68.46388888888889,  334.9073924731184,  334.0080555555555,  334.16344086021525,  334.25134408602156,  335.3929166666667,  0.17217741935483866,  0.09624999999999997,  0.21263440860215055,  0.27809139784946246,  0.28333333333333327,  3.677956989247311,  3.860694444444443,  4.945564516129032,  4.906048387096774,  4.228749999999999] | |
| 10 | [[18.65510752688173,  22.38194444444446,  23.424865591397857,  27.236021505376343,  22.137638888888862,  0.3279569892473117,  0.5640277777777778,  0.43279569892473135,  0.21397849462365587,  0.5901388888888892,  1.6177419354838705,  1.5494444444444446,  1.5989247311827952,  1.8280913978494622,  2.10972222222222,  89.83870967741936,  91.02638888888889,  93.34543010752688,  91.83064516129032,  81.80972222222222,  1004.9088709677419,  1001.0786111111115,  1001.2322580645157,  1002.4969086021509,  1004.9104166666667,  0.2643817204301076,  0.2105555555555555,  0.1517473118279571,  0.2459677419354838,  0.2286111111111112,  21.903091397849455,  25.62652777777778,  26.395161290322577,  31.217473118279568,  24.39500000000003],  [19.9590053763441,  21.678611111111124,  24.202822580645158,  27.088575268817216,  24.144861111111116,  0.5063172043010753,  0.26263888888888887,  0.6076612903225808,  0.426478494623656,  0.5484722222222222,  1.6194892473118279,  1.5569444444444436,  1.886021505376343,  1.6952956989247305,  2.2374999999999994,  65.39112903225806,  88.35694444444445,  96.68413978494624,  92.67069892473118,  91.38055555555556,  1006.4834677419358,  1001.4516666666673,  1000.7409946236563,  1001.1418010752681,  1006.62388888889,  0.3458333333333335,  0.2909722222222224,  0.1549731182795699,  0.26720430107526866,  0.2105555555555555,  23.625268817204315,  25.713194444444433,  27.052822580645156,  30.316397849462362,  25.974305555555556],  [18.7505376344086,  21.960694444444457,  26.675134408602133,  27.780913978494663,  23.537361111111103,  0.4786290322580647,  0.42263888888888873,  0.031182795698924733,  0.21451612903225803,  0.6580555555555557,  2.0344086021505383,  1.7076388888888903,  1.7995967741935488,  2.2663978494623658,  1.9869444444444448,  84.08602150537635,  88.48333333333333,  93.21102150537635,  91.60887096774194,  85.41666666666667,  1006.1380376344082,  1002.0863888888895,  1002.227150537634,  1000.8720430107509,  1006.1338888888879,  0.19650537634408588,  0.10333333333333335,  0.3036290322580647,  0.30698924731182775,  0.2090277777777778,  20.877150537634417,  24.753750000000032,  30.94825268817203,  31.6130376344086,  25.635416666666696],  [19.23225806451611,  22.128333333333355,  27.510887096774205,  28.251209677419343,  24.179444444444425,  0.0795698924731183,  0.34361111111111114,  0.06787634408602151,  0.31330645161290316,  0.14277777777777773,  1.5311827956989246,  1.4549999999999992,  1.7599462365591414,  1.7481182795698929,  2.105833333333335,  73.52956989247312,  80.56388888888888,  94.92204301075269,  83.63306451612904,  75.31944444444444,  1007.3932795698933,  1003.0279166666659,  1003.0975806451605,  1001.4299731182812,  1004.391805555555,  0.32889784946236555,  0.24583333333333326,  0.24650537634408595,  0.328763440860215,  0.2155555555555555,  22.902016129032265,  25.59347222222222,  31.794354838709687,  32.80430107526882,  25.99375000000001],  [18.79825268817205,  21.67444444444442,  25.806451612903224,  27.738306451612868,  23.71611111111109,  0.2911290322580644,  0.39055555555555554,  0.2731182795698924,  0.2297043010752688,  0.2698611111111111,  1.5018817204301085,  1.5798611111111107,  1.5872311827956982,  1.7245967741935462,  1.8095833333333315,  75.60349462365592,  87.94027777777778,  91.74596774193549,  77.21505376344086,  83.11111111111111,  1006.3930107526885,  1002.3786111111116,  1002.8442204301067,  999.9732526881721,  1006.0327777777784,  0.25416666666666665,  0.15402777777777785,  0.23749999999999993,  0.37997311827957,  0.17500000000000007,  22.284677419354832,  25.118750000000006,  30.216666666666672,  33.79112903225807,  25.123194444444437],  [19.090970350404294,  21.195416666666674,  24.230376344086007,  26.23615591397849,  22.790555555555535,  0.5747978436657682,  0.3770833333333333,  0.5333333333333335,  0.2114247311827956,  0.39847222222222217,  1.7925876010781667,  1.849861111111109,  2.3567204301075275,  1.9932795698924721,  2.5530555555555567,  69.46495956873315,  84.71111111111111,  89.26344086021506,  81.02956989247312,  73.46111111111111,  1004.4990566037728,  1001.3477777777781,  1001.2380376344081,  1002.3176075268815,  1006.7780555555562,  0.3405660377358491,  0.1984722222222222,  0.22715053763440865,  0.292741935483871,  0.2915277777777777,  23.554582210242597,  24.323472222222225,  27.466532258064525,  30.898387096774197,  25.153472222222202],  [19.41868279569893,  21.577916666666685,  25.293413978494613,  25.493279569892433,  24.27277777777777,  0.3709677419354839,  0.4179166666666667,  0.5840053763440862,  0.7467741935483873,  0.24819444444444447,  2.1297043010752676,  2.113611111111111,  2.104704301075271,  2.882123655913977,  2.6287500000000006,  66.5725806451613,  82.56805555555556,  85.25940860215054,  83.72849462365592,  73.36805555555556,  1006.1650537634413,  1000.6397222222225,  1001.7512096774186,  1001.2505376344088,  1006.2105555555559,  0.36478494623655916,  0.21222222222222228,  0.17029569892473115,  0.19233870967741934,  0.2580555555555556,  23.084946236559162,  24.59208333333332,  27.898387096774176,  27.235483870967734,  24.860972222222234],  [19.169758064516124,  22.187777777777782,  27.31720430107528,  29.114516129032253,  25.34069444444445,  0.2157258064516129,  0.21666666666666662,  0.02526881720430107,  0.23024193548387098,  0.09722222222222222,  2.320833333333334,  2.394166666666666,  2.5348118279569873,  2.160349462365594,  2.525416666666667,  72.51881720430107,  85.18888888888888,  85.40860215053763,  77.9260752688172,  68.16805555555555,  1006.2443548387087,  1001.9597222222222,  1002.4510752688167,  1002.0747311827948,  1007.7104166666682,  0.3458333333333332,  0.17472222222222225,  0.26330645161290306,  0.35994623655913993,  0.28680555555555565,  24.128897849462383,  25.169861111111125,  32.99865591397851,  34.00040322580647,  26.91305555555554],  [19.888440860215034,  22.3852777777778,  26.069086021505374,  27.918413978494648,  22.985277777777775,  0.12983870967741934,  0.2681944444444444,  0.26411290322580644,  1.2455645161290323,  0.2683333333333333,  2.5200268817204345,  2.696111111111108,  2.0655913978494618,  3.1243279569892475,  2.508055555555556,  69.03629032258064,  80.19722222222222,  87.62231182795699,  81.875,  73.55694444444444,  1005.3373655913979,  1001.2066666666667,  1000.9939516129034,  1001.8255376344072,  1006.8629166666666,  0.2935483870967744,  0.1488888888888888,  0.20846774193548387,  0.23051075268817203,  0.2572222222222223,  23.065322580645145,  25.17124999999999,  29.267204301075253,  30.04314516129034,  24.65305555555556],  [18.698924731182796,  21.655555555555555,  26.69206989247314,  26.33508064516129,  25.07222222222226,  0.16922043010752683,  0.8118055555555556,  0.17526881720430104,  0.3309139784946236,  0.15583333333333332,  2.6759408602150527,  2.251805555555557,  2.320833333333334,  2.924327956989246,  2.5419444444444435,  76.22983870967742,  88.9486111111111,  84.44462365591399,  84.8508064516129,  68.07083333333334,  1005.628091397849,  1001.8041666666657,  1001.1162634408602,  1002.3767473118284,  1005.1508333333325,  0.20940860215053772,  0.08555555555555558,  0.16787634408602156,  0.14838709677419362,  0.3309722222222223,  22.045698924731187,  24.275833333333317,  29.159274193548416,  28.207123655913975,  28.3763888888889],  [18.354704301075277,  21.970555555555556,  25.92029569892475,  28.257795698924735,  25.513472222222223,  0.14771505376344085,  0.5375,  0.4112903225806452,  0.6602150537634408,  0.22611111111111112,  2.7431451612903253,  2.3441666666666694,  2.2672043010752705,  2.5982526881720425,  2.695416666666669,  69.12768817204301,  77.74444444444444,  85.12768817204301,  82.72849462365592,  75.45833333333333,  1005.6631720430114,  1003.7230555555559,  1002.8973118279574,  1004.958198924732,  1006.8912499999996,  0.26706989247311835,  0.17791666666666642,  0.13266129032258067,  0.22741935483870954,  0.22749999999999992,  22.02876344086022,  25.39888888888888,  28.214112903225775,  30.23830645161292,  26.620277777777762],  [19.926881720430117,  22.152916666666655,  25.438306451612917,  26.59274193548386,  24.4561111111111,  0.1389784946236559,  0.5298611111111111,  0.4577956989247314,  0.21330645161290318,  0.1865277777777778,  2.481720430107526,  2.4119444444444436,  2.6107526881720426,  3.0388440860215074,  2.759861111111114,  60.333333333333336,  77.47083333333333,  83.10618279569893,  76.44354838709677,  68.65555555555555,  1006.9716397849469,  1000.6729166666657,  1000.740725806451,  1002.6198924731177,  1006.7688888888879,  0.3395161290322584,  0.20847222222222223,  0.1331989247311829,  0.2040322580645161,  0.25833333333333347,  23.454166666666662,  25.102777777777778,  26.944220430107517,  28.59220430107525,  25.61694444444444],  [18.61344086021503,  20.802361111111118,  27.156854838709673,  27.12862903225805,  24.67388888888888,  0.2821236559139785,  0.5240277777777779,  0.08346774193548388,  0.2399193548387097,  0.47666666666666674,  2.4284946236559146,  2.819166666666667,  1.9760752688172032,  2.671505376344088,  2.516388888888887,  69.90456989247312,  83.70083333333334,  83.41935483870968,  77.10349462365592,  75.06805555555556,  1003.0907258064518,  1002.6558333333325,  1002.3975806451615,  1001.6680107526871,  1005.9422222222229,  0.2803763440860216,  0.08527777777777777,  0.25000000000000006,  0.24731182795698922,  0.17458333333333315,  20.846102150537618,  22.114861111111104,  30.29529569892477,  29.111693548387123,  25.991388888888917],  [19.42123655913978,  21.827499999999997,  25.392607526881704,  28.244758064516127,  25.097222222222218,  0.20698924731182797,  0.20055555555555551,  0.6239247311827958,  0.44368279569892477,  0.7029166666666667,  2.7778225806451573,  2.799027777777777,  2.5668010752688195,  2.8940860215053736,  3.0877777777777755,  61.64650537634409,  80.82777777777778,  81.78763440860214,  78.41397849462365, | 74.9,  334.6102150537635,  334.2184722222222,  333.00551075268805,  334.4631720430108,  334.85736111111095,  0.31451612903225795,  0.1720833333333333,  0.22150537634408604,  0.26747311827957004,  0.18166666666666656,  3.5017473118279567,  3.9337500000000007,  4.696908602150536,  5.036424731182794,  4.284166666666666],  [17.87513440860215,  21.63499999999999,  25.41236559139786,  28.5950268817204,  23.094444444444456,  0.24018817204301074,  0.40555555555555556,  0.4184139784946236,  0.24596774193548387,  0.2252777777777778,  3.2506720430107556,  2.461666666666665,  2.6159946236559146,  2.223252688172039,  3.4356944444444437,  74.76747311827957,  80.36944444444444,  88.14381720430107,  75.67204301075269,  67.94305555555556,  335.4259408602151,  333.9488888888887,  333.63951612903253,  334.04905913978484,  335.488611111111,  0.20430107526881716,  0.2019444444444443,  0.14489247311827957,  0.32392473118279574,  0.21986111111111117,  3.348118279569892,  4.017916666666669,  4.48064516129032,  5.0497311827956946,  4.028472222222221],  [18.07500000000002,  22.32166666666666,  25.601612903225828,  26.166666666666686,  24.756666666666657,  0.20591397849462365,  0.13374999999999998,  0.2616935483870968,  0.3272849462365591,  0.02361111111111111,  2.7543010752688186,  2.8706944444444464,  3.0471774193548433,  3.0783602150537654,  3.737222222222223,  64.48924731182795,  75.86111111111111,  80.9274193548387,  78.45295698924731,  73.96527777777777,  334.9958333333333,  333.58555555555574,  333.82459677419337,  334.28776881720444,  335.907638888889,  0.32836021505376345,  0.2538888888888889,  0.1854838709677419,  0.22137096774193551,  0.2537500000000001,  3.927553763440858,  4.510555555555555,  4.816129032258065,  4.871908602150538,  4.967638888888885],  [19.353091397849465,  23.603333333333335,  27.285215053763423,  28.935483870967733,  24.790694444444437,  0.5661290322580645,  0.2505555555555556,  0.07110215053763441,  0.5651881720430109,  0.45208333333333334,  2.6233870967741906,  2.612916666666664,  2.9700268817204325,  3.3451612903225807,  3.88291666666667,  74.16397849462365,  70.72222222222223,  82.30510752688173,  74.35483870967742,  74.82777777777778,  334.901747311828,  334.59611111111116,  334.22204301075266,  333.6798387096774,  335.7847222222221,  0.22244623655913978,  0.24305555555555544,  0.25672043010752693,  0.32419354838709685,  0.1831944444444444,  3.3942204301075276,  4.184444444444446,  4.925134408602149,  5.0751344086021515,  4.139305555555554],  [19.378494623655918,  22.54,  24.638978494623707,  27.621236559139792,  25.52972222222223,  0.5268817204301075,  0.5236111111111111,  0.43790322580645163,  0.41747311827956984,  0.31013888888888885,  3.4084677419354854,  2.8433333333333333,  3.4297043010752697,  2.8505376344086053,  3.2383333333333346,  72.54032258064517,  75.77222222222223,  87.76747311827957,  78.92876344086021,  71.71527777777777,  335.17647849462367,  333.6955555555556,  333.9116935483872,  334.28602150537625,  335.65805555555545,  0.18749999999999992,  0.23333333333333342,  0.07755376344086021,  0.21975806451612903,  0.3080555555555556,  3.318817204301075,  4.0222222222222195,  4.14032258064516,  4.768682795698925,  4.342083333333334],  [19.16129032258063,  22.444583333333338,  25.39341397849464,  26.661693548387078,  23.863888888888887,  0.26142473118279563,  0.1348611111111111,  0.496505376344086,  0.5651881720430108,  0.18763888888888888,  2.922043010752686,  3.0026388888888897,  3.178091397849461,  3.3264784946236583,  3.3055555555555545,  72.07123655913979,  74.16944444444445,  85.59408602150538,  81.37768817204301,  69.55555555555556,  335.25698924731176,  334.01986111111097,  332.99879032258053,  333.95510752688176,  335.8436111111111,  0.1955645161290323,  0.26416666666666666,  0.15322580645161288,  0.20645161290322583,  0.27819444444444436,  3.5061827956989235,  4.286805555555555,  4.43709677419355,  4.608870967741935,  3.99236111111111],  [19.77620967741939,  22.238055555555576,  27.410483870967752,  27.93024193548384,  24.420694444444468,  0.21841397849462346,  0.3775,  0.3672043010752688,  0.325268817204301,  0.10333333333333335,  2.0879032258064507,  2.8700000000000006,  2.306451612903222,  3.1323924731182773,  4.203888888888887,  70.96774193548387,  78.58888888888889,  81.74596774193549,  71.91801075268818,  68.46388888888889,  334.9073924731184,  334.0080555555555,  334.16344086021525,  334.25134408602156,  335.3929166666667,  0.17217741935483866,  0.09624999999999997,  0.21263440860215055,  0.27809139784946246,  0.28333333333333327,  3.677956989247311,  3.860694444444443,  4.945564516129032,  4.906048387096774,  4.228749999999999],  [18.581451612903237,  22.3525,  27.16935483870969,  28.281720430107544,  23.766388888888876,  0.21774193548387094,  0.2315277777777777,  0.18252688172043008,  0.32836021505376345,  0.2613888888888887,  2.1784946236559137,  2.01486111111111,  2.5147849462365586,  2.7770161290322575,  3.1894444444444447,  70.02150537634408,  78.46388888888889,  81.19220430107526,  76.4758064516129,  72.44166666666666,  335.08991935483886,  333.97611111111104,  333.62674731182784,  334.1702956989248,  334.94374999999974,  0.2275537634408602,  0.12986111111111107,  0.2025537634408602,  0.2365591397849463,  0.24166666666666667,  3.446908602150537,  3.9470833333333317,  4.985887096774192,  5.009274193548387,  4.045555555555554],  [18.288575268817226,  21.593888888888905,  23.599596774193543,  25.262231182795713,  25.007361111111116,  0.3924731182795699,  0.41083333333333333,  1.234139784946237,  0.8955645161290323,  0.440138888888889,  2.6768817204301083,  2.3747222222222213,  3.5107526881720417,  3.5701612903225786,  3.3338888888888905,  23.381720430107528,  26.27638888888889,  27.942204301075268,  28.155913978494624,  27.104166666666668,  335.01438172043026,  333.8509722222221,  332.9112903225808,  333.36774193548393,  334.8188888888887,  0.33709677419354833,  0.2025,  0.13911290322580647,  0.1287634408602151,  0.18555555555555556,  3.6907258064516126,  3.9183333333333317,  4.206586021505378,  4.432930107526881,  4.2848611111111135],  [19.689112903225805,  21.193333333333342,  26.19220430107525,  27.498252688172016,  24.97888888888888,  0.12500000000000003,  0.4395833333333335,  0.3798387096774193,  0.14919354838709672,  0.6631944444444446,  2.7232526881720456,  3.0713888888888876,  2.9767473118279537,  2.670564516129034,  3.306527777777778,  22.802419354838708,  25.8625,  26.756720430107528,  26.911290322580644,  24.930555555555557,  335.6475806451613,  333.89777777777783,  333.1591397849465,  334.2725806451615,  334.96069444444447,  0.22163978494623665,  0.15680555555555561,  0.20053763440860214,  0.24032258064516138,  0.27555555555555566,  3.780779569892473,  3.8520833333333306,  4.901612903225806,  5.1724462365591375,  4.6768055555555526],  [17.89153225806451,  21.537222222222276,  24.41612903225805,  25.176344086021505,  22.555277777777793,  0.16760752688172034,  0.18902777777777782,  0.31599462365591396,  0.3361559139784944,  0.045972222222222234,  2.3498655913978492,  2.2641666666666675,  2.8590053763440877,  3.299596774193542,  3.448194444444442,  24.190860215053764,  24.65277777777778,  27.52956989247312,  25.169354838709676,  21.86111111111111,  334.34059139784995,  333.8961111111108,  333.8741935483872,  333.9766129032257,  335.5251388888891,  0.26236559139784943,  0.24958333333333327,  0.18884408602150535,  0.1931451612903226,  0.2593055555555557,  3.6998655913978498,  4.5586111111111105,  4.793951612903228,  4.799462365591398,  4.393888888888891],  [17.869086021505396,  20.32333333333335,  23.998118279569905,  26.67809139784944,  22.91027777777778,  0.16142473118279566,  0.5581944444444448,  0.10887096774193544,  0.44784946236559137,  0.031111111111111114,  1.9973118279569877,  2.1034722222222237,  2.4373655913978483,  2.2948924731182805,  2.544305555555558,  21.553763440860216,  27.729166666666668,  29.045698924731184,  26.774193548387096,  23.543055555555554,  335.51935483870966,  333.8183333333335,  334.29099462365593,  334.08548387096783,  335.32888888888897,  0.31599462365591396,  0.1306944444444445,  0.208736559139785,  0.24153225806451614,  0.2515277777777777,  3.6668010752688165,  4.021805555555552,  4.819623655913979,  5.408333333333335,  4.53486111111111],  [16.674327956989252,  19.322638888888882,  23.273655913978516,  26.707123655913957,  22.288194444444432,  0.49825268817204293,  0.2820833333333333,  1.0622311827956998,  0.39905913978494606,  0.09805555555555555,  2.2395161290322574,  2.9380555555555565,  2.3802419354838724,  2.6044354838709682,  2.873333333333335,  21.66532258064516,  25.395833333333332,  29.580645161290324,  27.899193548387096,  22.97361111111111,  335.36935483870985,  333.61833333333317,  334.233064516129,  335.0067204301077,  335.58916666666676,  0.2821236559139784,  0.25527777777777777,  0.14623655913978498,  0.3012096774193549,  0.22055555555555562,  3.241532258064516,  3.8187499999999983,  4.55981182795699,  5.228629032258065,  4.202222222222222],  [18.278763440860235,  20.22013888888887,  27.747177419354816,  26.777822580645147,  23.177499999999988,  0.2861559139784945,  0.2609722222222221,  0.08346774193548388,  0.5981182795698926,  0.18416666666666665,  2.6848118279569886,  2.909444444444446,  2.358467741935484,  3.8909946236559128,  2.863194444444447,  24.018817204301076,  26.154166666666665,  26.22983870967742,  28.045698924731184,  20.947222222222223,  334.51827956989246,  333.9629166666667,  333.6689516129034,  334.3638440860216,  334.9261111111112,  0.3010752688172042,  0.20236111111111113,  0.4016129032258064,  0.3135752688172043,  0.35611111111111116,  3.626478494623656,  4.0126388888888895,  5.812096774193548,  5.1103494623655905,  4.712777777777781]]  35 |
| 11 | RMSE  train: 1.0815959172631836e-20 test: 395172970.8158775  R\_Squared  train: 0.5 test: nan  /usr/local/lib/python3.7/dist-packages/sklearn/metrics/\_regression.py:582: UndefinedMetricWarning: R^2 score is not well-defined with less than two samples.  warnings.warn(msg, UndefinedMetricWarning) | |
| 12 | RMSE  train: 5241.212223725862 test: 2217085157.738646  R\_Squared  train: 0.4999971037177965 test: -1.787170723811236 | |
| 13 | RMSE  train: 144735.64409511082 test: 2681129652.2096486  R\_Squared  train: 0.4999200194053755 test: -2.265884398202707  /usr/local/lib/python3.7/dist-packages/sklearn/linear\_model/\_coordinate\_descent.py:476: ConvergenceWarning: Objective did not converge. You might want to increase the number of iterations. Duality gap: 13414483.07808632, tolerance: 4524086.27288304  positive) | |
| 14 | RMSE  train: 4358387858.346138 test: 3551564239.3724823  R\_Squared  train: -1.908435425109992 test: -3.1638347984399884  /usr/local/lib/python3.7/dist-packages/sklearn/neural\_network/\_multilayer\_perceptron.py:571: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimization hasn't converged yet.  % self.max\_iter, ConvergenceWarning) | |
| 15 |  | |
| 16 |  | |

2. 연구 수행 일정

|  |  |
| --- | --- |
| 5월 | 연구 주제 선정, 기본 지식 개별 공부 |
| 6월 | 필요 데이터 수집 |
| 7월 | 데이터 전처리 |
| 8월 1일 ~ 8월 28일 | 코드 개발 및 결론 도출 |
| 8월 28일 ~ 8월 31일 | 보고서 작성 |

**ⅲ. 연구 과제 수행 및 성과**

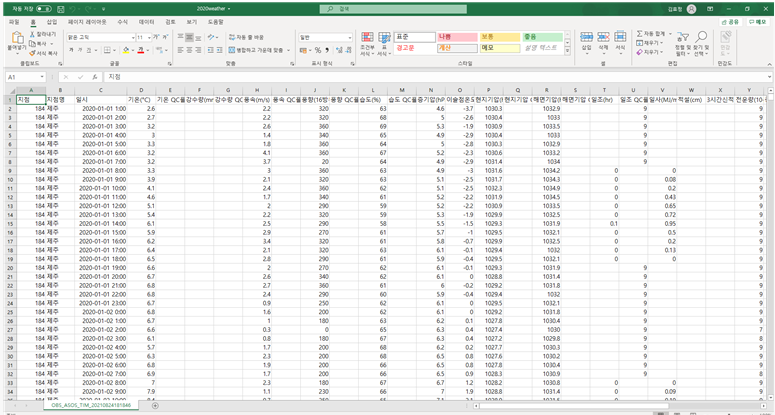
1. 출처

활용한 Raw 데이터

1. 날씨데이터

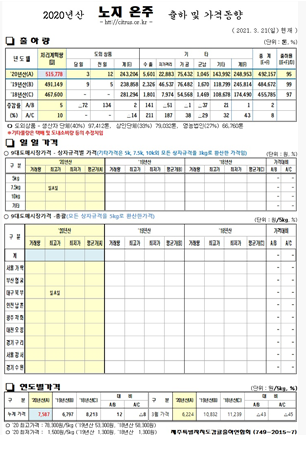


<https://data.kma.go.kr/data/grnd/selectAsosRltmList.do?pgmNo=36>



기상청의 기상 자료 개방 포털에서 제주도의 기상관측지점에서의 기온, 강수, 바람, 기압, 습도, 일사, 일조, 눈, 구름, 시정, 지면상태, 지면 · 초상온도, 일기현상, 증발량, 현상번호를 다운로드 가능하도록 데이터셋을 제공하였다.

1. 감귤 출하량 데이터



http://citrus.or.kr/board/list.php?btable=daily&p=&cate=0

제주특별자치도 감귤출하연합회에서 일일단위로 감귤의 출하량정보를 기록하여 pdf로 제공하고 있다.

이를 수작업하여 엑셀파일로 변환하여 활용하였다.

1. 연구 결과 및 성과
2. 기대 효과 및 연구 활용 계획

|  |  |  |
| --- | --- | --- |
| **ⅳ. 연구 프로젝트 멘토링 결과** | | |
| **연번** | **일시** | **멘토링 지도 및 수행내용 (\*캡쳐 화면 또는 사진 첨부 필수)** |
| 1 | 4/28 | * 주제 선정 |
| 2 | 6/28 | * 선형회귀분석 |
| 3 | 8/24 | * 다중공산성, RNN |
| 4 | 8/26 | * 코드 자문 |
| **멘토 교수 : (인)** | | |