HW2_Classification

March 27, 2020

1 Final Results - Classification

- Model - SoftMax Classifier
- Best parameters: {'C': 30}
- Cross-validation scores: 0.8252484472049689
- Train score: 0.8277
- Test score: 0.8667

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from sklearn.metrics import mean_squared_error, r2_score
from math import sqrt
from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_score
%matplotlib inline
```

[2]:
```
data = pd.read_csv(r'C:\Users\nabhs\OneDrive\BUAN - Semester 2\BUAN 6341 -␣
↪Applied Machine Learning\Datasets\titanic.csv')

data.head()
```

[2]:
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin Embarked
```

```
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

[3]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
PassengerId    891 non-null int64
Survived       891 non-null int64
Pclass         891 non-null int64
Name           891 non-null object
Sex            891 non-null object
Age            714 non-null float64
SibSp          891 non-null int64
Parch          891 non-null int64
Ticket         891 non-null object
Fare           891 non-null float64
Cabin          204 non-null object
Embarked       889 non-null object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

[4]:
```
data.head(5)
```

[4]:
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
```

```
 4      0            373450   8.0500   NaN        S
```

[5]:
```
# let's inspect the variable values

for var in data.columns:
    print(var, data[var].unique()[0:20], '\n')
```

```
PassengerId [ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20]

Survived [0 1]

Pclass [3 1 2]

Name ['Braund, Mr. Owen Harris'
 'Cumings, Mrs. John Bradley (Florence Briggs Thayer)'
 'Heikkinen, Miss. Laina' 'Futrelle, Mrs. Jacques Heath (Lily May Peel)'
 'Allen, Mr. William Henry' 'Moran, Mr. James' 'McCarthy, Mr. Timothy J'
 'Palsson, Master. Gosta Leonard'
 'Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)'
 'Nasser, Mrs. Nicholas (Adele Achem)' 'Sandstrom, Miss. Marguerite Rut'
 'Bonnell, Miss. Elizabeth' 'Saundercock, Mr. William Henry'
 'Andersson, Mr. Anders Johan' 'Vestrom, Miss. Hulda Amanda Adolfina'
 'Hewlett, Mrs. (Mary D Kingcome) ' 'Rice, Master. Eugene'
 'Williams, Mr. Charles Eugene'
 'Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)'
 'Masselmani, Mrs. Fatima']

Sex ['male' 'female']

Age [22. 38. 26. 35. nan 54.  2. 27. 14.  4. 58. 20. 39. 55. 31. 34. 15. 28.
  8. 19.]

SibSp [1 0 3 4 2 5 8]

Parch [0 1 2 5 3 4 6]

Ticket ['A/5 21171' 'PC 17599' 'STON/O2. 3101282' '113803' '373450' '330877'
 '17463' '349909' '347742' '237736' 'PP 9549' '113783' 'A/5. 2151'
 '347082' '350406' '248706' '382652' '244373' '345763' '2649']

Fare [ 7.25   71.2833  7.925  53.1     8.05    8.4583 51.8625 21.075  11.1333
 30.0708 16.7    26.55   31.275   7.8542 16.     29.125  13.     18.
  7.225  26.    ]

Cabin [nan 'C85' 'C123' 'E46' 'G6' 'C103' 'D56' 'A6' 'C23 C25 C27' 'B78' 'D33'
 'B30' 'C52' 'B28' 'C83' 'F33' 'F G73' 'E31' 'A5' 'D10 D12']
```

```
Embarked ['S' 'C' 'Q' nan]
```

[6]:
```
# make list of variables  types

# numerical: discrete vs continuous
discrete = [var for var in data.columns if data[var].dtype!='O' and var!
 ↪='Survived' and data[var].nunique()<10]
continuous = [var for var in data.columns if data[var].dtype!='O' and var!
 ↪='Survived' and var not in discrete]

# mixed
mixed = ['Cabin']

# categorical
categorical = [var for var in data.columns if data[var].dtype=='O' and var not␣
 ↪in mixed]

print(f'There are {len(discrete)} discrete variables')
print(f'There are {len(continuous)} continuous variables')
print(f'There are {len(categorical)} categorical variables')
print(f'There are {len(mixed)} mixed variables')
```

```
There are 3 discrete variables
There are 3 continuous variables
There are 4 categorical variables
There are 1 mixed variables
```

[7]:
```
# missing values
data.isnull().mean()
```

[7]:
```
PassengerId    0.000000
Survived       0.000000
Pclass         0.000000
Name           0.000000
Sex            0.000000
Age            0.198653
SibSp          0.000000
Parch          0.000000
Ticket         0.000000
Fare           0.000000
Cabin          0.771044
Embarked       0.002245
dtype: float64
```

[8]:
```
# cardinality (number of different categories)
```

```
data[categorical+mixed].nunique()
```

[8]:
```
Name        891
Sex           2
Ticket      681
Embarked      3
Cabin       147
dtype: int64
```

[9]:
```
# Cabin- mixed variable
# \d regular expression for digts . \d+ one or more digits
data['Cabin_num'] = data['Cabin'].str.extract('(\d+)') # captures numerical part
data['Cabin_num'] = data['Cabin_num'].astype('float')
data['Cabin_cat'] = data['Cabin'].str[0] # captures the first letter

# show dataframe
data.head()
```

[9]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Cabin_num	Cabin_cat
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S	NaN	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th…	female	38.0	1	0	PC 17599	71.2833	C85	C	85.0	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	NaN	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	123.0	C
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S	NaN	NaN

[10]:
```
data['Title'] = data['Name'].str.split(',').str[1].str.split('\s+').str[1]
data.head()
```

[10]:

	PassengerId	Survived	Pclass
0	1	0	3
1	2	1	1
2	3	1	3
3	4	1	1

```
4            5         0       3

                                                 Name     Sex   Age  SibSp  \
0                             Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                              Heikkinen, Miss. Laina  female  26.0      0
3        Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                            Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin Embarked  Cabin_num Cabin_cat  Title
0      0         A/5 21171   7.2500   NaN        S        NaN       NaN    Mr.
1      0          PC 17599  71.2833   C85        C       85.0         C   Mrs.
2      0  STON/O2. 3101282   7.9250   NaN        S        NaN       NaN  Miss.
3      0            113803  53.1000  C123        S      123.0         C   Mrs.
4      0            373450   8.0500   NaN        S        NaN       NaN    Mr.
```

[11]:
```
data['Title'].value_counts()
```

[11]:
```
Mr.          517
Miss.        182
Mrs.         125
Master.       40
Dr.            7
Rev.           6
Major.         2
Mlle.          2
Col.           2
Mme.           1
Jonkheer.      1
Lady.          1
the            1
Don.           1
Sir.           1
Ms.            1
Capt.          1
Name: Title, dtype: int64
```

[12]:
```
data['Cabin_cat'].value_counts()
```

[12]:
```
C    59
B    47
D    33
E    32
A    15
F    13
G     4
T     1
```

```
Name: Cabin_cat, dtype: int64
```

[13]:
```
# drop original mixed
data.head()
```

[13]:
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin Embarked  Cabin_num Cabin_cat  Title
0      0         A/5 21171   7.2500   NaN        S        NaN       NaN    Mr.
1      0          PC 17599  71.2833   C85        C       85.0         C   Mrs.
2      0  STON/O2. 3101282   7.9250   NaN        S        NaN       NaN  Miss.
3      0            113803  53.1000  C123        S      123.0         C   Mrs.
4      0            373450   8.0500   NaN        S        NaN       NaN    Mr.
```

[14]:
```
data.drop(['PassengerId','Name','Ticket','Cabin'], axis=1,inplace=True)
```

[15]:
```
data.head()
```

[15]:
```
   Survived  Pclass     Sex   Age  SibSp  Parch     Fare Embarked  Cabin_num  \
0         0       3    male  22.0      1      0   7.2500        S        NaN
1         1       1  female  38.0      1      0  71.2833        C       85.0
2         1       3  female  26.0      0      0   7.9250        S        NaN
3         1       1  female  35.0      1      0  53.1000        S      123.0
4         0       3    male  35.0      0      0   8.0500        S        NaN

  Cabin_cat  Title
0       NaN    Mr.
1         C   Mrs.
2       NaN  Miss.
3         C   Mrs.
4       NaN    Mr.
```

[16]:
```
data.describe()
```

[16]:
```
          Survived      Pclass         Age       SibSp       Parch        Fare  \
count  891.000000  891.000000  714.000000  891.000000  891.000000  891.000000
mean     0.383838    2.308642   29.699118    0.523008    0.381594   32.204208
std      0.486592    0.836071   14.526497    1.102743    0.806057   49.693429
min      0.000000    1.000000    0.420000    0.000000    0.000000    0.000000
25%      0.000000    2.000000   20.125000    0.000000    0.000000    7.910400
50%      0.000000    3.000000   28.000000    0.000000    0.000000   14.454200
75%      1.000000    3.000000   38.000000    1.000000    0.000000   31.000000
max      1.000000    3.000000   80.000000    8.000000    6.000000  512.329200

       Cabin_num
count  200.00000
mean    50.49000
std     35.39497
min      2.00000
25%     22.00000
50%     43.00000
75%     77.25000
max    148.00000
```

[17]:
```
# separate into training and testing set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    data.drop('Survived', axis=1),  # predictors
    data['Survived'],  # target
    test_size=0.1,  # percentage of obs in test set
    random_state=0)  # seed to ensure reproducibility

X_train.shape, X_test.shape
```

[17]:
```
((801, 10), (90, 10))
```

[18]:
```
X_train.head()
```

[18]:
```
     Pclass     Sex   Age  SibSp  Parch     Fare Embarked  Cabin_num  \
815       1    male   NaN      0      0   0.0000        S      102.0
877       3    male  19.0      0      0   7.8958        S        NaN
193       2    male   3.0      1      1  26.0000        S        2.0
523       1  female  44.0      0      1  57.9792        C       18.0
634       3  female   9.0      3      2  27.9000        S        NaN

    Cabin_cat    Title
815         B      Mr.
877       NaN      Mr.
193         F  Master.
523         B     Mrs.
634       NaN    Miss.
```

[19]:
```
X_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 801 entries, 815 to 684
Data columns (total 10 columns):
Pclass       801 non-null int64
Sex          801 non-null object
Age          643 non-null float64
SibSp        801 non-null int64
Parch        801 non-null int64
Fare         801 non-null float64
Embarked     799 non-null object
Cabin_num    175 non-null float64
Cabin_cat    179 non-null object
Title        801 non-null object
dtypes: float64(3), int64(3), object(4)
memory usage: 68.8+ KB
```

[20]:
```
# from feature-engine
from feature_engine import missing_data_imputers as mdi
# for one hot encoding with feature-engine
from feature_engine.categorical_encoders import OneHotCategoricalEncoder
from feature_engine.categorical_encoders import RareLabelCategoricalEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
```

[21]:
```
titanic_pipe = Pipeline([

    # missing data imputation
    ('imputer_num_arbit',
     mdi.ArbitraryNumberImputer(arbitrary_number=-1,
                                variables=['Cabin_num'])),
    ('imputer_num_mean',
     mdi.MeanMedianImputer(imputation_method='mean',variables=['Age'])),

    ('imputer_cat_freq',
     mdi.FrequentCategoryImputer(variables=['Embarked'])),

    ('imputer_cat_missing',
     mdi.CategoricalVariableImputer(variables=['Cabin_cat','Title'])),

    # categorical encoding
    ('encoder_rare_label',
     RareLabelCategoricalEncoder(tol=0.01,
                                 n_categories=4,
                                 variables=['Cabin_cat','Title'])),
    ('categorical_encoder',
```

```
    OneHotCategoricalEncoder( top_categories=None,
                              variables=['Sex',␣
↪'Embarked','Cabin_cat','Title'], # we can select which variables to encode
                              drop_last=True)),

])
```

[22]:
```
titanic_pipe.fit(X_train, y_train)
```

[22]:
```
Pipeline(memory=None,
         steps=[('imputer_num_arbit',
                 ArbitraryNumberImputer(arbitrary_number=-1,
                                        variables=['Cabin_num'])),
                ('imputer_num_mean',
                 MeanMedianImputer(imputation_method='mean',
                                   variables=['Age'])),
                ('imputer_cat_freq',
                 FrequentCategoryImputer(variables=['Embarked'])),
                ('imputer_cat_missing',
                 CategoricalVariableImputer(variables=['Cabin_cat', 'Title'])),
                ('encoder_rare_label',
                 RareLabelCategoricalEncoder(n_categories=4, tol=0.01,
                                             variables=['Cabin_cat', 'Title'])),
                ('categorical_encoder',
                 OneHotCategoricalEncoder(drop_last=True, top_categories=None,
                                          variables=['Sex', 'Embarked',
                                                     'Cabin_cat', 'Title']))],
         verbose=False)
```

[23]:
```
# Apply Transformations
X_train=titanic_pipe.transform(X_train)
X_test=titanic_pipe.transform(X_test)
```

2 DO NOT CHANGE STEPS BEFORE THIS POINT

2.1 Linear SVC

[24]:
```
from sklearn.svm import LinearSVC
from sklearn.metrics import accuracy_score
linear_svm = LinearSVC().fit(X_train, y_train)
print("Coefficient shape: ", linear_svm.coef_.shape)
print("Intercept shape: ", linear_svm.intercept_.shape)
```

```
Coefficient shape:  (1, 20)
Intercept shape:  (1,)
```

```
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
```

```
packages\sklearn\svm\_base.py:947: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  "the number of iterations.", ConvergenceWarning)
```

[25]:
```
cv_scores_linear = cross_val_score(linear_svm, X_train, y_train)
```

```
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\svm\_base.py:947: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  "the number of iterations.", ConvergenceWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\svm\_base.py:947: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  "the number of iterations.", ConvergenceWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\svm\_base.py:947: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  "the number of iterations.", ConvergenceWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\svm\_base.py:947: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  "the number of iterations.", ConvergenceWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\svm\_base.py:947: ConvergenceWarning: Liblinear failed to
converge, increase the number of iterations.
  "the number of iterations.", ConvergenceWarning)
```

2.1.1 Results

[26]:
```
# let's get the predictions
X_train_preds = linear_svm.predict(X_train)
X_test_preds = linear_svm.predict(X_test)

# check model performance:

print('train mse: {}'.format(mean_squared_error(y_train, X_train_preds)))
print('train rmse: {}'.format(sqrt(mean_squared_error(y_train, X_train_preds))))
print('train r2: {}'.format(r2_score(y_train, X_train_preds)))
print()
print('test mse: {}'.format(mean_squared_error(y_test, X_test_preds)))
print('test rmse: {}'.format(sqrt(mean_squared_error(y_test, X_test_preds))))
print('test r2: {}'.format(r2_score(y_test, X_test_preds)))
print()
print("Cross-validation scores: {}".format(cv_scores_linear))
print('Train score: {:.4f}'.format(linear_svm.score(X_train, y_train)))
print('Test score: {:.4f}'.format(linear_svm.score(X_test, y_test)))
```

```
train mse: 0.2908863920099875
```

```
train rmse: 0.5393388471174568
train r2: -0.2368483836335442

test mse: 0.36666666666666664
test rmse: 0.6055300708194983
test r2: -0.4932126696832577

Cross-validation scores: [0.80745342 0.7125     0.725      0.725      0.73125
]
Train score: 0.7091
Test score: 0.6333
```

2.2 Kernel SVC

[27]:
```
from sklearn.svm import SVC
param_svc = {#'C': [0.001, 0.01, 0.1, 1, 10,100, 1000],
             'C':[1,20,30],
             'kernel':['rbf','polynomial','sigmoid'],
             'gamma' :['auto']}
print("Parameter grid:\n{}".format(param_svc))
```

```
Parameter grid:
{'C': [1, 20, 30], 'kernel': ['rbf', 'polynomial', 'sigmoid'], 'gamma':
['auto']}
```

[28]:
```
from sklearn.model_selection import GridSearchCV
grid_svc = GridSearchCV(SVC(), param_svc, cv=5, return_train_score=True)
grid_svc.fit(X_train, y_train)
```

```
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
ValueError: 'polynomial' is not in list

  FitFailedWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
ValueError: 'polynomial' is not in list

  FitFailedWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
ValueError: 'polynomial' is not in list
```

```
  FitFailedWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
ValueError: 'polynomial' is not in list

  FitFailedWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
ValueError: 'polynomial' is not in list

  FitFailedWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
ValueError: 'polynomial' is not in list

  FitFailedWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
ValueError: 'polynomial' is not in list

  FitFailedWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
ValueError: 'polynomial' is not in list

  FitFailedWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
ValueError: 'polynomial' is not in list

  FitFailedWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
```

```
ValueError: 'polynomial' is not in list

  FitFailedWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
ValueError: 'polynomial' is not in list

  FitFailedWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
ValueError: 'polynomial' is not in list

  FitFailedWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
ValueError: 'polynomial' is not in list

  FitFailedWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
ValueError: 'polynomial' is not in list

  FitFailedWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
ValueError: 'polynomial' is not in list

  FitFailedWarning)
```

[28]:
```
GridSearchCV(cv=5, error_score=nan,
             estimator=SVC(C=1.0, break_ties=False, cache_size=200,
                           class_weight=None, coef0=0.0,
                           decision_function_shape='ovr', degree=3,
                           gamma='scale', kernel='rbf', max_iter=-1,
                           probability=False, random_state=None, shrinking=True,
                           tol=0.001, verbose=False),
             iid='deprecated', n_jobs=None,
```

```
param_grid={'C': [1, 20, 30], 'gamma': ['auto'],
            'kernel': ['rbf', 'polynomial', 'sigmoid']},
pre_dispatch='2*n_jobs', refit=True, return_train_score=True,
scoring=None, verbose=0)
```

2.2.1 Results

[29]:
```
# let's get the predictions
X_train_preds = grid_svc.predict(X_train)
X_test_preds = grid_svc.predict(X_test)

# check model performance:

print('train mse: {}'.format(mean_squared_error(y_train, X_train_preds)))
print('train rmse: {}'.format(sqrt(mean_squared_error(y_train, X_train_preds))))
print('train r2: {}'.format(r2_score(y_train, X_train_preds)))
print()
print('test mse: {}'.format(mean_squared_error(y_test, X_test_preds)))
print('test rmse: {}'.format(sqrt(mean_squared_error(y_test, X_test_preds))))
print('test r2: {}'.format(r2_score(y_test, X_test_preds)))
print()
print("Best parameters: {}".format(grid_svc.best_params_))
print("Best cross-validation score: {:.2f}".format(grid_svc.best_score_))
print('Train score: {:.4f}'.format(grid_svc.score(X_train, y_train)))
print('Test score: {:.4f}'.format(grid_svc.score(X_test, y_test)))
```

```
train mse: 0.056179775280898875
train rmse: 0.2370227315699886
train r2: 0.7611237027317189

test mse: 0.2111111111111111
test rmse: 0.45946829173634074
test r2: 0.14027149321266974

Best parameters: {'C': 20, 'gamma': 'auto', 'kernel': 'rbf'}
Best cross-validation score: 0.74
Train score: 0.9438
Test score: 0.7889
```

2.3 KNN Classifier

[30]:
```
from math import sqrt
print(sqrt(len(y_test)))
```

```
9.486832980505138
```

[31]:
```
# Train a KNN model, report the coefficients, the best parameters, and model␣
↪performance
# hint: find the optimal k

# YOUR CODE HERE

from sklearn.model_selection import GridSearchCV
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier()

# define a list of parameters

#param_knn = {'n_neighbors': range(5,25)}
param_knn = {'n_neighbors': range(1,10)}

#apply grid search
grid_knn = GridSearchCV(knn, param_knn, cv=5, return_train_score=True)
grid_knn.fit(X_train, y_train)

# Mean Cross Validation Score
print("Best Mean Cross-validation score: {:.2f}".format(grid_knn.best_score_))
print()

#find best parameters
print("KNN parameters: {}".format(grid_knn.best_params_))

# Check test data set performance
print("KNN Test Performance: ", grid_knn.score(X_test,y_test))
```

```
Best Mean Cross-validation score: 0.72

KNN parameters: {'n_neighbors': 7}
KNN Test Performance:  0.8111111111111111
```

2.3.1 Results

[32]:
```
# let's get the predictions
X_train_preds = grid_knn.predict(X_train)
X_test_preds = grid_knn.predict(X_test)

# check model performance:

print('train mse: {}'.format(mean_squared_error(y_train, X_train_preds)))
print('train rmse: {}'.format(sqrt(mean_squared_error(y_train, X_train_preds))))
print('train r2: {}'.format(r2_score(y_train, X_train_preds)))
print()
```

```
print('test mse: {}'.format(mean_squared_error(y_test, X_test_preds)))
print('test rmse: {}'.format(sqrt(mean_squared_error(y_test, X_test_preds))))
print('test r2: {}'.format(r2_score(y_test, X_test_preds)))
print()

print("Best parameters: {}".format(grid_knn.best_params_))
print("Best cross-validation score: {:.2f}".format(grid_knn.best_score_))
print('Train score: {:.4f}'.format(grid_knn.score(X_train, y_train)))
print('Test score: {:.4f}'.format(grid_knn.score(X_test, y_test)))
```

```
train mse: 0.21598002496878901
train rmse: 0.46473651133603544
train r2: 0.08165334605749719

test mse: 0.18888888888888888
test rmse: 0.4346134936801766
test r2: 0.23076923076923084

Best parameters: {'n_neighbors': 7}
Best cross-validation score: 0.72
Train score: 0.7840
Test score: 0.8111
```

2.4 DecisionTree Classifier

[33]:
```
# Train a Decision Tree model, report the coefficients, the best parameters,␣
↪and model performance (10 points)
# hint: find the optimal max_depth

# YOUR CODE HERE
from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier(random_state=0)

#define a list of parameters
param_dtree = {'max_depth': range(1,20)}

#apply grid search
grid_dtree = GridSearchCV(dtree, param_dtree, cv=5, return_train_score = True)
grid_dtree.fit(X_train, y_train)

# Mean Cross Validation Score
print("Best Mean Cross-validation score: {:.2f}".format(grid_dtree.best_score_))
print()

#find best parameters
print('Decision Tree parameters: ', grid_dtree.best_params_)
```

```
# Check test data set performance
print("Decision Tree Performance: ", grid_dtree.score(X_test,y_test))
```

```
Best Mean Cross-validation score: 0.83

Decision Tree parameters:  {'max_depth': 4}
Decision Tree Performance:  0.8222222222222222
```

2.4.1 Results

[34]:
```
# let's get the predictions
X_train_preds = grid_dtree.predict(X_train)
X_test_preds = grid_dtree.predict(X_test)

# check model performance:

print('train mse: {}'.format(mean_squared_error(y_train, X_train_preds)))
print('train rmse: {}'.format(sqrt(mean_squared_error(y_train, X_train_preds))))
print('train r2: {}'.format(r2_score(y_train, X_train_preds)))
print()
print('test mse: {}'.format(mean_squared_error(y_test, X_test_preds)))
print('test rmse: {}'.format(sqrt(mean_squared_error(y_test, X_test_preds))))
print('test r2: {}'.format(r2_score(y_test, X_test_preds)))
print()
print("Best parameters: {}".format(grid_dtree.best_params_))
print("Best cross-validation score: {:.2f}".format(grid_dtree.best_score_))
print('Train score: {:.4f}'.format(grid_dtree.score(X_train, y_train)))
print('Test score: {:.4f}'.format(grid_dtree.score(X_test, y_test)))
```

```
train mse: 0.14357053682896379
train rmse: 0.37890702926834674
train r2: 0.38953835142550397

test mse: 0.17777777777777778
test rmse: 0.4216370213557839
test r2: 0.27601809954751133

Best parameters: {'max_depth': 4}
Best cross-validation score: 0.83
Train score: 0.8564
Test score: 0.8222
```

2.5 Logisitic Regression

[35]:
```
param_logit = {'C': [0.001, 0.01, 0.1, 1, 10,1000],
               'penalty':['l1','l2']}
print("Parameter grid:\n{}".format(param_logit))
```

```
Parameter grid:
{'C': [0.001, 0.01, 0.1, 1, 10, 1000], 'penalty': ['l1', 'l2']}
```

[36]:
```
grid_logit = GridSearchCV(LogisticRegression(), param_logit, cv=5,␣
↪return_train_score=True)
```

[37]:
```
grid_logit.fit(X_train, y_train)
```

```
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penalty.

  FitFailedWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penalty.

  FitFailedWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

```
Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penalty.

  FitFailedWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
```

```
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penalty.

  FitFailedWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
```

```
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penalty.

  FitFailedWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
```

```
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
```

```
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\model_selection\_validation.py:536: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penalty.

  FitFailedWarning)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
```

```
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
```

[37]:
```
GridSearchCV(cv=5, error_score=nan,
             estimator=LogisticRegression(C=1.0, class_weight=None, dual=False,
                                          fit_intercept=True,
                                          intercept_scaling=1, l1_ratio=None,
                                          max_iter=100, multi_class='auto',
                                          n_jobs=None, penalty='l2',
                                          random_state=None, solver='lbfgs',
                                          tol=0.0001, verbose=0,
                                          warm_start=False),
             iid='deprecated', n_jobs=None,
             param_grid={'C': [0.001, 0.01, 0.1, 1, 10, 1000],
                         'penalty': ['l1', 'l2']},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=True,
             scoring=None, verbose=0)
```

2.5.1 Results

[38]:
```
# let's get the predictions
X_logittrain = grid_logit.predict(X_train)
X_logittest = grid_logit.predict(X_test)
```

```
# check model performance:

print('train mse: {}'.format(mean_squared_error(y_train, X_logittrain)))
print('train rmse: {}'.format(sqrt(mean_squared_error(y_train, X_logittrain))))
print('train r2: {}'.format(r2_score(y_train, X_logittrain)))
print()
print('test mse: {}'.format(mean_squared_error(y_test, X_logittest)))
print('test rmse: {}'.format(sqrt(mean_squared_error(y_test, X_logittest))))
print('test r2: {}'.format(r2_score(y_test, X_logittest)))
print()
print("Best parameters: {}".format(grid_logit.best_params_))
print("Best cross-validation score: {:.2f}".format(grid_logit.best_score_))
print('Train score: {:.4f}'.format(grid_logit.score(X_train, y_train)))
print('Test score: {:.4f}'.format(grid_logit.score(X_test, y_test)))
```

```
train mse: 0.1735330836454432
train rmse: 0.41657302318494316
train r2: 0.2621376595490873

test mse: 0.14444444444444443
test rmse: 0.38005847503304596
test r2: 0.41176470588235303

Best parameters: {'C': 1, 'penalty': 'l2'}
Best cross-validation score: 0.82
Train score: 0.8265
Test score: 0.8556
```

2.6 SoftMax

[39]:
```
from sklearn.linear_model import LogisticRegression
```

[40]:
```
param_soft = {#'C': [0.001, 0.01, 0.1, 1, 10,100, 1000],
              'C':[1,20,30]}
print("Parameter grid:\n{}".format(param_soft))
```

```
Parameter grid:
{'C': [1, 20, 30]}
```

[41]:
```
grid_soft = GridSearchCV(LogisticRegression(
    multi_class="multinomial",solver="lbfgs", C=15), param_soft, cv=5,␣
 ↪return_train_score=True)
```

[42]:
```
grid_soft.fit(X_train, y_train)
```

```
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
```

```
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
```

```
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
```

```
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
```

```
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\nabhs\Anaconda3\envs\buan6341_2020\lib\site-
packages\sklearn\linear_model\_logistic.py:940: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
```

[42]:
```
GridSearchCV(cv=5, error_score=nan,
             estimator=LogisticRegression(C=15, class_weight=None, dual=False,
                                          fit_intercept=True,
                                          intercept_scaling=1, l1_ratio=None,
                                          max_iter=100,
                                          multi_class='multinomial',
                                          n_jobs=None, penalty='l2',
                                          random_state=None, solver='lbfgs',
                                          tol=0.0001, verbose=0,
                                          warm_start=False),
             iid='deprecated', n_jobs=None, param_grid={'C': [1, 20, 30]},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=True,
             scoring=None, verbose=0)
```

2.6.1 Results

[43]:
```
# let's get the predictions
X_train_preds = grid_soft.predict(X_train)
X_test_preds = grid_soft.predict(X_test)

# check model performance:

print('train mse: {}'.format(mean_squared_error(y_train, X_train_preds)))
print('train rmse: {}'.format(sqrt(mean_squared_error(y_train, X_train_preds))))
print('train r2: {}'.format(r2_score(y_train, X_train_preds)))
print()
print('test mse: {}'.format(mean_squared_error(y_test, X_test_preds)))
print('test rmse: {}'.format(sqrt(mean_squared_error(y_test, X_test_preds))))
print('test r2: {}'.format(r2_score(y_test, X_test_preds)))
print()

print("Best parameters: {}".format(grid_soft.best_params_))
print("Cross-validation scores: {}".format(grid_soft.best_score_))
print('Train score: {:.4f}'.format(grid_soft.score(X_train, y_train)))
print('Test score: {:.4f}'.format(grid_soft.score(X_test, y_test)))
```

```
train mse: 0.17228464419475656
train rmse: 0.41507185425508747
train r2: 0.2674460217106047

test mse: 0.13333333333333333
test rmse: 0.3651483716701107
test r2: 0.4570135746606335

Best parameters: {'C': 30}
Cross-validation scores: 0.8252484472049689
Train score: 0.8277
Test score: 0.8667
```

[ ]:
```

```