Scikit-Learn Project

November 21, 2021

1 Exoplanet Exploration

1.1 Resources

- Exoplanet Data Source
- Scikit-Learn Tutorial Part 1
- Scikit-Learn Tutorial Part 2
- Grid Search

[1]:
```
# !pip install sklearn --upgrade
```

[2]:
```
# !pip install joblib
```

[3]:
```
import pandas as pd
```

2 Read the CSV and Perform Basic Data Cleaning

[4]:
```
df = pd.read_csv("exoplanet_data.csv")
# Dropping the null columns where all values are null
df = df.dropna(axis='columns', how='all')
# Dropping the null rows
df = df.dropna()
# Cleaning
mask = df["koi_disposition"] == "FALSE POSITIVE"
df.loc[mask, "koi_disposition"] = "False_Positive"
df
```

[4]:

	koi_disposition	koi_fpflag_nt	koi_fpflag_ss	koi_fpflag_co	\
0	CONFIRMED	0	0	0	
1	False_Positive	0	1	0	
2	False_Positive	0	1	0	
3	CONFIRMED	0	0	0	
4	CONFIRMED	0	0	0	
…	…	…	…	…	
6986	False_Positive	0	0	0	
6987	False_Positive	0	1	1	

```
6988       CANDIDATE              0              0              0
6989  False_Positive              0              0              1
6990  False_Positive              0              0              1

      koi_fpflag_ec  koi_period  koi_period_err1  koi_period_err2  \
0                 0   54.418383     2.479000e-04    -2.479000e-04
1                 0   19.899140     1.490000e-05    -1.490000e-05
2                 0    1.736952     2.630000e-07    -2.630000e-07
3                 0    2.525592     3.760000e-06    -3.760000e-06
4                 0    4.134435     1.050000e-05    -1.050000e-05
…               …          …               …               …
6986              1    8.589871     1.846000e-04    -1.846000e-04
6987              0    0.527699     1.160000e-07    -1.160000e-07
6988              0    1.739849     1.780000e-05    -1.780000e-05
6989              0    0.681402     2.430000e-06    -2.430000e-06
6990              1    4.856035     6.360000e-05    -6.360000e-05

      koi_time0bk  koi_time0bk_err1  …  koi_steff_err2  koi_slogg  \
0      162.513840          0.003520  …             -81      4.467
1      175.850252          0.000581  …            -176      4.544
2      170.307565          0.000115  …            -174      4.564
3      171.595550          0.001130  …            -211      4.438
4      172.979370          0.001900  …            -232      4.486
…           …                 …  …  …               …         …
6986   132.016100          0.015700  …            -152      4.296
6987   131.705093          0.000170  …            -166      4.529
6988   133.001270          0.007690  …            -220      4.444
6989   132.181750          0.002850  …            -236      4.447
6990   135.993300          0.010800  …            -225      4.385

      koi_slogg_err1  koi_slogg_err2  koi_srad  koi_srad_err1  koi_srad_err2  \
0              0.064          -0.096     0.927          0.105         -0.061
1              0.044          -0.176     0.868          0.233         -0.078
2              0.053          -0.168     0.791          0.201         -0.067
3              0.070          -0.210     1.046          0.334         -0.133
4              0.054          -0.229     0.972          0.315         -0.105
…              …              …       …             …              …
6986           0.231          -0.189     1.088          0.313         -0.228
6987           0.035          -0.196     0.903          0.237         -0.079
6988           0.056          -0.224     1.031          0.341         -0.114
6989           0.056          -0.224     1.041          0.341         -0.114
6990           0.054          -0.216     1.193          0.410         -0.137

             ra        dec  koi_kepmag
0     291.93423  48.141651      15.347
1     297.00482  48.134129      15.436
2     285.53461  48.285210      15.597
```

```
3      288.75488  48.226200       15.509
4      296.28613  48.224670       15.714
…           …          …            …
6986   298.74921  46.973351       14.478
6987   297.18875  47.093819       14.082
6988   286.50937  47.163219       14.757
6989   294.16489  47.176281       15.385
6990   297.00977  47.121021       14.826

[6991 rows x 41 columns]
```

3 Select features

[5]:
```
# Setingt features. This will also be used as x values
selected_features =␣
↪df[['koi_fpflag_nt','koi_fpflag_ss','koi_fpflag_co','koi_fpflag_ec','koi_period','koi_perio
↪'koi_impact',
      'koi_impact_err1', 'koi_impact_err2', 'koi_duration',
      'koi_duration_err1', 'koi_duration_err2', 'koi_depth', 'koi_depth_err1',
      'koi_depth_err2', 'koi_prad', 'koi_prad_err1', 'koi_prad_err2',
      'koi_teq', 'koi_insol', 'koi_insol_err1', 'koi_insol_err2',
      'koi_model_snr', 'koi_tce_plnt_num', 'koi_steff', 'koi_steff_err1',
      ␣
↪'koi_steff_err2','koi_slogg','koi_slogg_err1','koi_slogg_err2','koi_srad','koi_srad_err1','
```

4 Create a Train Test Split

Using `koi_disposition` for the y values

[6]:
```
from sklearn.model_selection import train_test_split
y = df["koi_disposition"]
X = selected_features
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1,␣
↪stratify=y)
```

[7]:
```
X_train.head()
```

[7]:
```
      koi_fpflag_nt  koi_fpflag_ss  koi_fpflag_co  koi_fpflag_ec  koi_period  \
4002              0              0              1              0   99.673478
4246              0              1              0              0    0.592244
548               0              1              1              0    9.991625
3953              0              1              0              0  178.412990
2362              0              0              0              0   45.294223

      koi_period_err1  koi_period_err2  koi_time0bk  koi_time0bk_err1  \
4002    3.463000e-04    -3.463000e-04   219.334830          0.002300
```

```
4246     9.000000e-08    -9.000000e-08   131.654831          0.000124
548      5.360000e-06    -5.360000e-06   137.447816          0.000445
3953     3.100000e-05    -3.100000e-05   218.225235          0.000127
2362     5.600000e-05    -5.600000e-05   138.678725          0.000987

      koi_time0bk_err2  …  koi_steff_err2  koi_slogg  koi_slogg_err1  \
4002         -0.002300  …            -148      4.777           0.040
4246         -0.000124  …            -146      4.664           0.056
548          -0.000445  …            -176      4.338           0.153
3953         -0.000127  …            -134      4.346           0.084
2362         -0.000987  …             -68      4.347           0.030

      koi_slogg_err2  koi_srad  koi_srad_err1  koi_srad_err2         ra  \
4002          -0.027     0.492          0.026         -0.027  293.05801
4246          -0.032     0.591          0.045         -0.045  290.28094
548           -0.187     1.096          0.309         -0.206  301.04239
3953          -0.126     1.148          0.202         -0.124  288.32785
2362          -0.030     1.044          0.057         -0.042  285.67938

            dec  koi_kepmag
4002  45.248821      15.801
4246  45.464260      15.653
548   45.022888      14.039
3953  38.627621      13.944
2362  50.241299      10.961

[5 rows x 40 columns]
```

5 Pre-processing

Scaling the data using the MinMaxScaler and performing some feature selection

[8]:
```
from sklearn.preprocessing import MinMaxScaler
X_scaler = MinMaxScaler().fit(X_train)

X_train_scaled = X_scaler.transform(X_train)
X_test_scaled = X_scaler.transform(X_test)
```

6 Training the Model: LogisticRegression

[9]:
```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X_train, y_train)

print(f"Training Data Score: {model.score(X_train_scaled, y_train)}")
print(f"Testing Data Score: {model.score(X_test_scaled, y_test)}")
```

```
Training Data Score: 0.26149151249284763
Testing Data Score: 0.2637299771167048

C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:764: 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)
```

7 Hyperparameter Tuning

Use `GridSearchCV` to tune the model's parameters

[10]:
```
# Creating the GridSearchCV model
from sklearn.model_selection import GridSearchCV
param_grid = {'C': [1, 5, 10],
              'penalty': ["l1", "l2"]}
grid = GridSearchCV(model, param_grid, verbose=3)
```

[11]:
```
# Training the model with GridSearch
grid.fit(X_train, y_train)
```

```
Fitting 5 folds for each of 6 candidates, totalling 30 fits
[CV] C=1, penalty=l1 …
[CV] … C=1, penalty=l1, score=nan, total=   0.0s
[CV] C=1, penalty=l1 …
[CV] … C=1, penalty=l1, score=nan, total=   0.0s
[CV] C=1, penalty=l1 …
[CV] … C=1, penalty=l1, score=nan, total=   0.0s
[CV] C=1, penalty=l1 …
[CV] … C=1, penalty=l1, score=nan, total=   0.0s

[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\model_selection\_validation.py:552: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\model_selection\_validation.py", line 531, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py", line 1304, in fit
```

```
    solver = _check_solver(self.solver, self.penalty, self.dual)
  File "C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py", line 443, in _check_solver
    "got %s penalty." % (solver, penalty))
ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penalty.

  FitFailedWarning)
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    0.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   2 out of   2 | elapsed:    0.0s remaining:    0.0s


[CV] C=1, penalty=l1 …
[CV] … C=1, penalty=l1, score=nan, total=   0.0s
[CV] C=1, penalty=l2 …

C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:764: 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)

[CV] … C=1, penalty=l2, score=0.610, total=   2.6s
[CV] C=1, penalty=l2 …

C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:764: 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)

[CV] … C=1, penalty=l2, score=0.641, total=   1.8s
[CV] C=1, penalty=l2 …

C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:764: 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)

[CV] … C=1, penalty=l2, score=0.626, total=   1.9s
[CV] C=1, penalty=l2 …

C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:764: 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)

[CV] … C=1, penalty=l2, score=0.648, total=   2.1s
[CV] C=1, penalty=l2 …

C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:764: 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\13306\Anaconda3\lib\site-
packages\sklearn\model_selection\_validation.py:552: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\model_selection\_validation.py", line 531, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py", line 1304, in fit
    solver = _check_solver(self.solver, self.penalty, self.dual)
  File "C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py", line 443, in _check_solver
    "got %s penalty." % (solver, penalty))
ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penalty.
```

```
  FitFailedWarning)

[CV] … C=1, penalty=l2, score=0.628, total=   1.9s
[CV] C=5, penalty=l1 …
[CV] … C=5, penalty=l1, score=nan, total=   0.0s
[CV] C=5, penalty=l1 …
[CV] … C=5, penalty=l1, score=nan, total=   0.0s
[CV] C=5, penalty=l1 …
[CV] … C=5, penalty=l1, score=nan, total=   0.0s
[CV] C=5, penalty=l1 …
[CV] … C=5, penalty=l1, score=nan, total=   0.0s
[CV] C=5, penalty=l1 …
[CV] … C=5, penalty=l1, score=nan, total=   0.0s
[CV] C=5, penalty=l2 …

C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:764: 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)

[CV] … C=5, penalty=l2, score=0.613, total=   5.7s
[CV] C=5, penalty=l2 …

C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:764: 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)

[CV] … C=5, penalty=l2, score=0.647, total=   2.4s
[CV] C=5, penalty=l2 …

C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:764: 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)

[CV] … C=5, penalty=l2, score=0.627, total=   1.9s
[CV] C=5, penalty=l2 …

C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:764: 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)

[CV] … C=5, penalty=l2, score=0.640, total=   2.1s
[CV] C=5, penalty=l2 …

C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:764: 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\13306\Anaconda3\lib\site-
packages\sklearn\model_selection\_validation.py:552: FitFailedWarning: Estimator
fit failed. The score on this train-test partition for these parameters will be
set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\model_selection\_validation.py", line 531, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py", line 1304, in fit
    solver = _check_solver(self.solver, self.penalty, self.dual)
  File "C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py", line 443, in _check_solver
    "got %s penalty." % (solver, penalty))
```

```
ValueError: Solver lbfgs supports only 'l2' or 'none' penalties, got l1 penalty.

  FitFailedWarning)

[CV] … C=5, penalty=l2, score=0.597, total=   2.4s
[CV] C=10, penalty=l1 …
[CV] … C=10, penalty=l1, score=nan, total=   0.0s
[CV] C=10, penalty=l1 …
[CV] … C=10, penalty=l1, score=nan, total=   0.0s
[CV] C=10, penalty=l1 …
[CV] … C=10, penalty=l1, score=nan, total=   0.0s
[CV] C=10, penalty=l1 …
[CV] … C=10, penalty=l1, score=nan, total=   0.0s
[CV] C=10, penalty=l1 …
[CV] … C=10, penalty=l1, score=nan, total=   0.0s
[CV] C=10, penalty=l2 …

C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:764: 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)

[CV] … C=10, penalty=l2, score=0.616, total=   2.6s
[CV] C=10, penalty=l2 …

C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:764: 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)

[CV] … C=10, penalty=l2, score=0.631, total=   2.6s
[CV] C=10, penalty=l2 …

C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:764: 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)

[CV] … C=10, penalty=l2, score=0.624, total=   3.5s
[CV] C=10, penalty=l2 …

C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:764: 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)

[CV] … C=10, penalty=l2, score=0.641, total=   4.0s
[CV] C=10, penalty=l2 …

C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:764: 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)
[Parallel(n_jobs=1)]: Done  30 out of  30 | elapsed:   40.2s finished

[CV] … C=10, penalty=l2, score=0.594, total=   2.2s

C:\Users\13306\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:764: 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)
```

[11]:
```
GridSearchCV(estimator=LogisticRegression(),
             param_grid={'C': [1, 5, 10], 'penalty': ['l1', 'l2']}, verbose=3)
```

[12]:
```
print(grid.best_params_)
print(grid.best_score_)
```

```
{'C': 1, 'penalty': 'l2'}
0.6305578195154965
```

8 Saving the Model

[13]:
```
import joblib
filename = 'regression.sav'
joblib.dump(model, filename)
```

[13]:
```
['regression.sav']
```

9 Training the Model: Deep Learning

[14]:
```
#!pip install tensorflow
```

[15]:
```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.layers import Dense
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.callbacks import EarlyStopping
```

[16]:
```
label_encoder = LabelEncoder()
label_encoder.fit(y_train)
encoded_y_train = label_encoder.transform(y_train)
encoded_y_test = label_encoder.transform(y_test)

y_train_categorical = to_categorical(encoded_y_train)
y_test_categorical = to_categorical(encoded_y_test)
```

[17]:
```
y_train_categorical.shape
```

[17]:
```
(5243, 3)
```

[18]:
```
model2 = Sequential()
model2.add(Dense(units=100, activation='relu', input_dim=40))
model2.add(Dense(units=100, activation='relu'))
model2.add(Dense(units=3, activation='softmax'))
```

[19]:
```
# Compiling and fitting the model
model2.compile(optimizer='adam',
               loss='categorical_crossentropy',
               metrics=['accuracy'])
```

[20]:
```
model2.summary()
```

```
Model: "sequential"

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 100)               4100

_________________________________________________________________
dense_1 (Dense)              (None, 100)               10100

_________________________________________________________________
dense_2 (Dense)              (None, 3)                 303
=================================================================
Total params: 14,503
Trainable params: 14,503
Non-trainable params: 0

_________________________________________________________________
```

[21]:
```
# set early stopping as callback
callbacks = [EarlyStopping(monitor='val_loss', patience=2)]
model2.fit(
    X_train_scaled,
    y_train_categorical,
    callbacks=callbacks,
    epochs=60,
    shuffle=True,
    verbose=2
)
```

```
Epoch 1/60
164/164 - 3s - loss: 0.5399 - accuracy: 0.7389
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 2/60
164/164 - 1s - loss: 0.3655 - accuracy: 0.8110
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 3/60
164/164 - 2s - loss: 0.3510 - accuracy: 0.8209
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 4/60
164/164 - 2s - loss: 0.3445 - accuracy: 0.8247
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
```

```
available. Available metrics are: loss,accuracy
Epoch 5/60
164/164 - 2s - loss: 0.3316 - accuracy: 0.8375
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 6/60
164/164 - 1s - loss: 0.3310 - accuracy: 0.8320
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 7/60
164/164 - 2s - loss: 0.3231 - accuracy: 0.8432
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 8/60
164/164 - 2s - loss: 0.3204 - accuracy: 0.8489
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 9/60
164/164 - 2s - loss: 0.3139 - accuracy: 0.8510
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 10/60
164/164 - 1s - loss: 0.3154 - accuracy: 0.8535
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 11/60
164/164 - 1s - loss: 0.3028 - accuracy: 0.8611
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 12/60
164/164 - 2s - loss: 0.3055 - accuracy: 0.8608
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 13/60
164/164 - 1s - loss: 0.3039 - accuracy: 0.8562
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 14/60
164/164 - 1s - loss: 0.2988 - accuracy: 0.8611
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 15/60
164/164 - 1s - loss: 0.2939 - accuracy: 0.8644
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 16/60
164/164 - 1s - loss: 0.2988 - accuracy: 0.8604
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
```

```
available. Available metrics are: loss,accuracy
Epoch 17/60
164/164 - 1s - loss: 0.2950 - accuracy: 0.8646
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 18/60
164/164 - 2s - loss: 0.2839 - accuracy: 0.8716
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 19/60
164/164 - 2s - loss: 0.2850 - accuracy: 0.8705
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 20/60
164/164 - 1s - loss: 0.2838 - accuracy: 0.8735
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 21/60
164/164 - 2s - loss: 0.2830 - accuracy: 0.8728
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 22/60
164/164 - 2s - loss: 0.2796 - accuracy: 0.8781
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 23/60
164/164 - 2s - loss: 0.2868 - accuracy: 0.8703
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 24/60
164/164 - 1s - loss: 0.2742 - accuracy: 0.8785
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 25/60
164/164 - 1s - loss: 0.2743 - accuracy: 0.8768
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 26/60
164/164 - 1s - loss: 0.2734 - accuracy: 0.8764
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 27/60
164/164 - 1s - loss: 0.2791 - accuracy: 0.8755
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 28/60
164/164 - 1s - loss: 0.2682 - accuracy: 0.8852
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
```

```
available. Available metrics are: loss,accuracy
Epoch 29/60
164/164 - 1s - loss: 0.2703 - accuracy: 0.8774
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 30/60
164/164 - 1s - loss: 0.2704 - accuracy: 0.8766
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 31/60
164/164 - 1s - loss: 0.2668 - accuracy: 0.8827
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 32/60
164/164 - 3s - loss: 0.2706 - accuracy: 0.8796
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 33/60
164/164 - 3s - loss: 0.2663 - accuracy: 0.8848
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 34/60
164/164 - 2s - loss: 0.2664 - accuracy: 0.8808
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 35/60
164/164 - 2s - loss: 0.2564 - accuracy: 0.8875
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 36/60
164/164 - 2s - loss: 0.2592 - accuracy: 0.8863
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 37/60
164/164 - 1s - loss: 0.2561 - accuracy: 0.8880
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 38/60
164/164 - 2s - loss: 0.2553 - accuracy: 0.8861
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 39/60
164/164 - 1s - loss: 0.2604 - accuracy: 0.8827
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 40/60
164/164 - 3s - loss: 0.2551 - accuracy: 0.8837
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
```

```
available. Available metrics are: loss,accuracy
Epoch 41/60
164/164 - 1s - loss: 0.2522 - accuracy: 0.8898
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 42/60
164/164 - 2s - loss: 0.2514 - accuracy: 0.8903
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 43/60
164/164 - 2s - loss: 0.2492 - accuracy: 0.8949
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 44/60
164/164 - 1s - loss: 0.2500 - accuracy: 0.8909
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 45/60
164/164 - 1s - loss: 0.2488 - accuracy: 0.8848
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 46/60
164/164 - 1s - loss: 0.2497 - accuracy: 0.8917
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 47/60
164/164 - 1s - loss: 0.2460 - accuracy: 0.8932
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 48/60
164/164 - 1s - loss: 0.2448 - accuracy: 0.8980
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 49/60
164/164 - 1s - loss: 0.2460 - accuracy: 0.8920
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 50/60
164/164 - 1s - loss: 0.2490 - accuracy: 0.8913
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 51/60
164/164 - 1s - loss: 0.2438 - accuracy: 0.8947
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 52/60
164/164 - 1s - loss: 0.2463 - accuracy: 0.8917
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
```

```
available. Available metrics are: loss,accuracy
Epoch 53/60
164/164 - 1s - loss: 0.2375 - accuracy: 0.8943
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 54/60
164/164 - 1s - loss: 0.2361 - accuracy: 0.8974
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 55/60
164/164 - 1s - loss: 0.2356 - accuracy: 0.8993
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 56/60
164/164 - 2s - loss: 0.2381 - accuracy: 0.8947
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 57/60
164/164 - 2s - loss: 0.2445 - accuracy: 0.8899
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 58/60
164/164 - 2s - loss: 0.2408 - accuracy: 0.8928
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 59/60
164/164 - 1s - loss: 0.2415 - accuracy: 0.8932
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
Epoch 60/60
164/164 - 1s - loss: 0.2404 - accuracy: 0.8947
WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which is not
available. Available metrics are: loss,accuracy
```

[21]:
```
<keras.callbacks.History at 0x15e1b450f88>
```

[22]:
```
model_loss, model_accuracy = model2.evaluate(
    X_test_scaled, y_test_categorical, verbose=2)
print(
    f"Normal Neural Network - Loss: {model_loss}, Accuracy: {model_accuracy}")
```

```
55/55 - 1s - loss: 0.2933 - accuracy: 0.8787
Normal Neural Network - Loss: 0.29330772161483765, Accuracy: 0.8787185549736023
```

[25]:
```
encoded_predictions = model2.predict(X_test_scaled[:5])
prediction_labels = np.argmax(encoded_predictions,axis=1)
```

```
<IPython.core.display.Javascript object>
```

[26]:
```
print(f"Predicted classes: {prediction_labels}")
print(f"Actual Labels: {list(y_test[:5])}")
```

```
Predicted classes: [0 2 2 0 2]
Actual Labels: ['CANDIDATE', 'False_Positive', 'False_Positive', 'CANDIDATE',
'False_Positive']
```

10 Saving the Model

[28]:
```
filename2 = 'deep_learning.sav'
#joblib.dump(model2, filename2)
```

11 Training the Model: SVM

[29]:
```
# Support vector machine linear classifier
from sklearn.svm import SVC
model3 = SVC(kernel='linear')
model3.fit(X_train_scaled, y_train)
```

[29]:
```
SVC(kernel='linear')
```

[30]:
```
print(f"Training Data Score: {model3.score(X_train_scaled, y_train)}")
print(f"Testing Data Score: {model3.score(X_test_scaled, y_test)}")
```

```
Training Data Score: 0.8439824527942018
Testing Data Score: 0.8415331807780321
```

[31]:
```
# Hyperparameter Tuning
param_grid2 = {'C': [1, 5, 10],
               'gamma': [0.0001, 0.001, 0.01]}
grid2 = GridSearchCV(model3, param_grid2, verbose=3)
```

[32]:
```
grid2.fit(X_train_scaled, y_train)
```

```
Fitting 5 folds for each of 9 candidates, totalling 45 fits
[CV] C=1, gamma=0.0001 …

[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.

[CV] … C=1, gamma=0.0001, score=0.856, total=   3.6s
[CV] C=1, gamma=0.0001 …

[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    3.6s remaining:    0.0s

[CV] … C=1, gamma=0.0001, score=0.846, total=   3.5s
[CV] C=1, gamma=0.0001 …

[Parallel(n_jobs=1)]: Done   2 out of   2 | elapsed:    7.1s remaining:    0.0s
```

```
[CV] … C=1, gamma=0.0001, score=0.839, total=   4.4s
[CV] C=1, gamma=0.0001 …
[CV] … C=1, gamma=0.0001, score=0.841, total=   4.7s
[CV] C=1, gamma=0.0001 …
[CV] … C=1, gamma=0.0001, score=0.825, total=   3.3s
[CV] C=1, gamma=0.001 …
[CV] … C=1, gamma=0.001, score=0.856, total=   3.0s
[CV] C=1, gamma=0.001 …
[CV] … C=1, gamma=0.001, score=0.846, total=   3.9s
[CV] C=1, gamma=0.001 …
[CV] … C=1, gamma=0.001, score=0.839, total=   3.3s
[CV] C=1, gamma=0.001 …
[CV] … C=1, gamma=0.001, score=0.841, total=   3.3s
[CV] C=1, gamma=0.001 …
[CV] … C=1, gamma=0.001, score=0.825, total=   2.9s
[CV] C=1, gamma=0.01 …
[CV] … C=1, gamma=0.01, score=0.856, total=   3.1s
[CV] C=1, gamma=0.01 …
[CV] … C=1, gamma=0.01, score=0.846, total=   6.3s
[CV] C=1, gamma=0.01 …
[CV] … C=1, gamma=0.01, score=0.839, total=   6.2s
[CV] C=1, gamma=0.01 …
[CV] … C=1, gamma=0.01, score=0.841, total=   2.8s
[CV] C=1, gamma=0.01 …
[CV] … C=1, gamma=0.01, score=0.825, total=   2.6s
[CV] C=5, gamma=0.0001 …
[CV] … C=5, gamma=0.0001, score=0.882, total=   3.8s
[CV] C=5, gamma=0.0001 …
[CV] … C=5, gamma=0.0001, score=0.868, total=   4.5s
[CV] C=5, gamma=0.0001 …
[CV] … C=5, gamma=0.0001, score=0.852, total=   6.7s
[CV] C=5, gamma=0.0001 …
[CV] … C=5, gamma=0.0001, score=0.869, total=   4.3s
[CV] C=5, gamma=0.0001 …
[CV] … C=5, gamma=0.0001, score=0.852, total=   4.6s
[CV] C=5, gamma=0.001 …
[CV] … C=5, gamma=0.001, score=0.882, total=   4.4s
[CV] C=5, gamma=0.001 …
[CV] … C=5, gamma=0.001, score=0.868, total=   5.3s
[CV] C=5, gamma=0.001 …
[CV] … C=5, gamma=0.001, score=0.852, total=   4.8s
[CV] C=5, gamma=0.001 …
[CV] … C=5, gamma=0.001, score=0.869, total=   5.7s
[CV] C=5, gamma=0.001 …
[CV] … C=5, gamma=0.001, score=0.852, total=   4.1s
[CV] C=5, gamma=0.01 …
[CV] … C=5, gamma=0.01, score=0.882, total=   4.3s
[CV] C=5, gamma=0.01 …
```

```
[CV] … C=5, gamma=0.01, score=0.868, total=   4.6s
[CV] C=5, gamma=0.01 …
[CV] … C=5, gamma=0.01, score=0.852, total=   5.7s
[CV] C=5, gamma=0.01 …
[CV] … C=5, gamma=0.01, score=0.869, total=   3.3s
[CV] C=5, gamma=0.01 …
[CV] … C=5, gamma=0.01, score=0.852, total=   2.6s
[CV] C=10, gamma=0.0001 …
[CV] … C=10, gamma=0.0001, score=0.881, total=   3.4s
[CV] C=10, gamma=0.0001 …
[CV] … C=10, gamma=0.0001, score=0.870, total=   3.0s
[CV] C=10, gamma=0.0001 …
[CV] … C=10, gamma=0.0001, score=0.856, total=   3.3s
[CV] C=10, gamma=0.0001 …
[CV] … C=10, gamma=0.0001, score=0.873, total=   2.4s
[CV] C=10, gamma=0.0001 …
[CV] … C=10, gamma=0.0001, score=0.860, total=   2.6s
[CV] C=10, gamma=0.001 …
[CV] … C=10, gamma=0.001, score=0.881, total=   2.8s
[CV] C=10, gamma=0.001 …
[CV] … C=10, gamma=0.001, score=0.870, total=   3.0s
[CV] C=10, gamma=0.001 …
[CV] … C=10, gamma=0.001, score=0.856, total=   3.2s
[CV] C=10, gamma=0.001 …
[CV] … C=10, gamma=0.001, score=0.873, total=   2.6s
[CV] C=10, gamma=0.001 …
[CV] … C=10, gamma=0.001, score=0.860, total=   2.7s
[CV] C=10, gamma=0.01 …
[CV] … C=10, gamma=0.01, score=0.881, total=   2.8s
[CV] C=10, gamma=0.01 …
[CV] … C=10, gamma=0.01, score=0.870, total=   2.9s
[CV] C=10, gamma=0.01 …
[CV] … C=10, gamma=0.01, score=0.856, total=   3.1s
[CV] C=10, gamma=0.01 …
[CV] … C=10, gamma=0.01, score=0.873, total=   2.5s
[CV] C=10, gamma=0.01 …
[CV] … C=10, gamma=0.01, score=0.860, total=   2.7s
[Parallel(n_jobs=1)]: Done  45 out of  45 | elapsed:  2.8min finished
```

[32]:
```
GridSearchCV(estimator=SVC(kernel='linear'),
             param_grid={'C': [1, 5, 10], 'gamma': [0.0001, 0.001, 0.01]},
             verbose=3)
```

[33]:
```
print(grid.best_params_)
print(grid.best_score_)
```

```
{'C': 1, 'penalty': 'l2'}
0.6305578195154965
```

[34]:
```
filename3 = 'svm.sav'
joblib.dump(grid2, filename3)
```

[34]:
```
['svm.sav']
```

12 Testing

[35]:
```
testing_data = pd.read_csv('test_data.csv')
# Dropping the null columns where all values are null
testing_data = testing_data.dropna(axis='columns', how='all')
# Dropping the null rows
testing_data = testing_data.dropna()
# Cleaning
mask = testing_data["koi_disposition"] == "FALSE POSITIVE"
testing_data.loc[mask, "koi_disposition"] = "False_Positive"
testing_data.head()
```

[35]:

	koi_disposition	koi_fpflag_nt	koi_fpflag_ss	koi_fpflag_co	koi_fpflag_ec	\
0	CANDIDATE	0	0	0	0	
1	CONFIRMED	0	0	0	0	
2	False_Positive	0	0	1	1	
3	False_Positive	1	0	0	0	
4	CONFIRMED	0	0	0	0	

	koi_period	koi_period_err1	koi_period_err2	koi_time0bk	\
0	10.191547	0.000078	-0.000078	136.69755	
1	8.308653	0.000010	-0.000010	138.10788	
2	9.942219	0.000059	-0.000059	133.84961	
3	0.948952	0.000010	-0.000010	131.85772	
4	14.613478	0.000156	-0.000156	140.41368	

	koi_time0bk_err1	…	koi_steff_err2	koi_slogg	koi_slogg_err1	\
0	0.00647	…	-74	4.049	0.013	
1	0.00102	…	-80	4.485	0.077	
2	0.00532	…	-155	4.434	0.126	
3	0.00821	…	-232	3.666	0.304	
4	0.00701	…	-136	4.242	0.125	

	koi_slogg_err2	koi_srad	koi_srad_err1	koi_srad_err2	ra	\
0	-0.015	1.684	0.107	-0.063	281.46173	
1	-0.033	0.849	0.040	-0.066	291.64090	
2	-0.273	0.917	0.269	-0.124	298.64505	
3	-0.076	3.067	0.461	-1.077	299.12161	
4	-0.125	1.314	0.244	-0.200	290.99527	

	dec	koi_kepmag

```
0  46.789894      11.740
1  41.241039      14.817
2  42.089851      14.129
3  41.424450      12.669
4  45.192211      13.223

[5 rows x 41 columns]
```

[36]:
```
# Splitting the test data into X and y
y_test = testing_data["koi_disposition"]
X_test = testing_data.drop(columns=["koi_disposition"])
```

[37]:
```
# loading model
filename4 = 'svm.sav'
loaded_model = joblib.load(filename3)
result = loaded_model.score(X_test, y_test)
print(result)
```

```
0.5139760410724472
```

[40]:
```
# loading
# filename5 = 'deep_learning.sav'
# loaded_model = joblib.load(filename5)
loaded_model = model2

label_encoder2 = LabelEncoder()
label_encoder2.fit(y_test)
encoded_y_test2 = label_encoder2.transform(y_test)

y_test_categorical2 = to_categorical(encoded_y_test2)

encoded_predictions2 = loaded_model.predict(X_test[:10])
prediction_labels2 = np.argmax(encoded_predictions2,axis=1)

# Take number correct over total to get "score" for grading
print(f"Predicted classes: {prediction_labels2}")
print(f"Actual Labels: {list(y_test[:10])}")
```

```
<IPython.core.display.Javascript object>

Predicted classes: [2 2 2 2 2 2 2 2 2 2]
Actual Labels: ['CANDIDATE', 'CONFIRMED', 'False_Positive', 'False_Positive',
'CONFIRMED', 'CONFIRMED', 'CONFIRMED', 'False_Positive', 'False_Positive',
'False_Positive']
```