CSE400 Assignment 3 report

Huahao Shang
hushang@syr.edu
SUID: hushang

1(a).

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	EnvironmentSatisfaction	JobSatisfaction	PerformanceRating	WorkLifeBalance	Y
0	41	Yes	Travel_Rarely	1102	Sales	1	2	4	3	1	
1	49	No	Travel_Frequently	279	Research & Development	8	3	2	4	3	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	4	3	3	3	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	3	3	3	
4	27	No	Travel_Rarely	591	Research & Development	2	1	2	3	3	

```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'EnvironmentSatisfaction', 'JobSatisfaction',
       'PerformanceRating', 'WorkLifeBalance', 'YearsAtCompany'],
      dtype='object')
```

JobSatisfaction	PerformanceRating	WorkLifeBalance	YearsAtCompany
4	3	1	6
2	4	3	10
3	3	3	0
3	3	3	8
2	3	3	2

```
ment',
ction',
```

(b) There is no NaN values

```
dfl.isna().any()
```

```
Age                       False
Attrition                 False
BusinessTravel            False
DailyRate                 False
Department                False
DistanceFromHome          False
Education                 False
EducationField            False
EmployeeCount             False
EmployeeNumber            False
EnvironmentSatisfaction   False
Gender                    False
HourlyRate                False
JobInvolvement            False
JobLevel                  False
JobRole                   False
JobSatisfaction           False
MaritalStatus             False
MonthlyIncome             False
MonthlyRate               False
NumCompaniesWorked        False
Over18                    False
OverTime                  False
PercentSalaryHike         False
PerformanceRating         False
RelationshipSatisfaction  False
StandardHours             False
StockOptionLevel          False
TotalWorkingYears         False
```

```
PerformanceRating         False
RelationshipSatisfaction  False
StandardHours             False
StockOptionLevel          False
TotalWorkingYears         False
TrainingTimesLastYear     False
WorkLifeBalance           False
YearsAtCompany            False
YearsInCurrentRole        False
YearsSinceLastPromotion   False
YearsWithCurrManager      False
dtype: bool
```

(c)

Yes, The “Attrition” is set to 0 or 1 based on Yes or No

(4)

	Age	Attrition	DailyRate	DistanceFromHome	EnvironmentSatisfaction	JobSatisfaction	PerformanceRating	WorkLifeBalance	YearsAtCompany	x0_Travel_Freq
0	41	1	1102	1	2	4	3	1	6	
1	49	0	279	8	3	2	4	3	10	
2	37	1	1373	2	4	3	3	3	0	
3	33	0	1392	3	4	3	3	3	8	
4	27	0	591	2	1	2	3	3	2	

nceRating	WorkLifeBalance	YearsAtCompany	x0_Travel_Frequently	x0_Travel_Rarely	x0_Research & Development	x0_Sales
3	1	6	0	1	0	1
4	3	10	1	0	1	0
3	3	0	0	1	1	0
3	3	8	1	0	1	0
3	3	2	0	1	1	0

2.

(a) train data: 1176 test data: 294
Index:285 Index:721

(b)
The performance is fine, with around 0.85 accuracy
score: 0.845

Beta features:

```
array([[-3.18147741e-02, -2.62857411e-04,  2.17927310e-02,
        -2.78447164e-01, -2.61351594e-01,  5.28027676e-01,
        -2.52099456e-01, -5.86571245e-02,  5.26873313e-01,
        -9.91669493e-02, -2.76229119e-01,  3.86154826e-01]])
```

(c)

Increase Iteration to 1000, the algorithm won't converge.
Use scaler, the algorithm will converge
Changing solver, newton-cg and lbfgs, it also won't converge.

(d) the accuracy: 0.836

(e)

With the converged algorithm, changing L1, L2 penalty and C = 1.0 or 10.0 won't affect much of the accuracy. The accuracy is always around 0.836.

solver='liblinear', multi_class = 'auto', penalty = 'l1', C = 1

	pred_Attrition	Attrition
7	0	0
14	0	1
17	0	0
18	0	0
20	0	0

```
Percentage of correct predictions is
0.8367346938775511
```

solver='liblinear', multi_class = 'auto', penalty = 'l1', C = 10

	pred_Attrition	Attrition
7	0	0
14	0	1
17	0	0
18	0	0
20	0	0

```
Percentage of correct predictions is
0.8367346938775511
```

solver='liblinear', multi_class = 'auto', penalty = 'l2', C = 10

	pred_Attrition	Attrition
7	0	0
14	0	1
17	0	0
18	0	0
20	0	0

```
Percentage of correct predictions is
0.8367346938775511
```

solver='liblinear', multi_class = 'auto', penalty = 'l2', C = 1

	pred_Attrition	Attrition
7	0	0
14	0	1
17	0	0
18	0	0
20	0	0

```
Percentage of correct predictions is
0.8367346938775511
```

3
(a)

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	EnvironmentSatisfaction	JobSatisfaction	PerformanceRating	WorkLifeBalance	Y
0	41	Yes	Travel_Rarely	1102	Sales	1	2	4	3	1	
1	49	No	Travel_Frequently	279	Research & Development	8	3	2	4	3	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	4	3	3	3	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	3	3	3	
4	27	No	Travel_Rarely	591	Research & Development	2	1	2	3	3	

ıg	WorkLifeBalance	YearsAtCompany	MaritalStatus
3	1	6	Single
4	3	10	Married
3	3	0	Single
3	3	8	Married
3	3	2	Married

(b)
Train length: 1176 Test Length: 294
Index: 285 Index: 721

(c)
Accuracy: 0.486. The accuracy is not very well. The algorithm doesn't converge also,

```
LogisticRegression(multi_class='ovr', penalty='none')

0.48299319727891155

array([[ 6.16011731e-03,  2.01925594e-04, -5.01327267e-03,
         1.68199242e-03, -6.10015731e-02, -3.03079587e-01,
        -5.03407810e-02,  6.58801095e-03, -1.40700475e-01,
        -2.09342154e-01,  5.07588692e-02, -2.73608731e-01,
        -5.48741761e-01],
       [ 1.22791790e-02,  1.82448865e-04,  1.30092561e-02,
        -2.79898272e-02, -3.21883575e-02, -1.88156799e-01,
         1.68843834e-02,  3.00646179e-04, -3.13039587e-02,
         1.58077825e-01, -1.79365773e-01, -2.72444530e-02,
        -4.69236738e-01],
       [-2.57341375e-02, -4.36499047e-04, -1.32900918e-02,
         2.64940213e-02,  7.64577049e-02,  8.16774931e-02,
        -8.20351708e-04, -5.56578571e-03,  7.98993177e-02,
        -1.18892923e-01,  2.54921125e-02,  1.96400607e-01,
         8.38995033e-01]])

array([-0.1198122 , -0.10557289,  0.03323385])
```

(d)
By increasing the iteration, the algorithm converges.

```
LogisticRegression(max_iter=1000, multi_class='ovr', penalty='none')

0.48639455782312924

array([[ 8.06179501e-03,  1.97346150e-04, -5.87966355e-03,
        -1.67074532e-02, -6.94243385e-02, -2.01517082e-01,
        -9.74078863e-02,  2.57054148e-03, -2.19569429e-01,
        -2.64960786e-01, -5.98956640e-02, -2.51044564e-01,
        -7.00819448e-01],
       [ 1.04686544e-02,  1.80352021e-04,  1.32474771e-02,
        -2.69902868e-02, -6.53969446e-02, -2.33555298e-02,
        -1.02664435e-02,  1.49822408e-03,  3.69906995e-01,
         4.20782998e-01, -4.76225352e-01, -3.44495573e-01,
        -5.92373068e-01],
       [-2.14001588e-02, -4.06761642e-04, -1.18032365e-02,
         2.94344481e-02,  1.24242176e-01, -9.27633885e-02,
         4.53480168e-02, -4.33122041e-03, -2.03281045e-01,
        -2.31179944e-01,  3.90428446e-01,  4.50462602e-01,
         9.93104141e-01]])

array([-0.12877401, -0.35767902, -0.09939082])
```

By using scaler, the algorithm also converges.

```
LogisticRegression(multi_class='ovr', penalty='none')

0.483843537414966

array([[ 0.07138442,  0.08237795, -0.04751135, -0.01965579, -0.07873187,
        -0.05216146, -0.07349446,  0.01983936, -0.11449575, -0.14744609,
        -0.09657201, -0.20706819, -0.24316969],
       [ 0.11147008,  0.08219233,  0.11113723, -0.02188601, -0.06179201,
         0.0277134 ,  0.01100717,  0.01036311,  0.13649332,  0.18214652,
        -0.19322897, -0.13349832, -0.20754517],
       [-0.19200763, -0.16480818, -0.09445363,  0.04145396,  0.1362615 ,
         0.00827789,  0.04692884, -0.0273917 , -0.06054792, -0.08390798,
         0.35001086,  0.3652269 ,  0.36644813]])

array([-1.30759213, -0.19329194, -0.75571357])
```

(e)
Accuracy of prediction: 0.472

	pred_MaritalStatus	MaritalStatus
7	Married	Divorced
14	Married	Single
17	Married	Divorced
18	Married	Married
20	Married	Divorced

```
Percentage of correct predictions is
0.47278911564625853
```

(f)

```
solver='newton-cg', multi_class = 'ovr', penalty = 'l2', max_iter =
1000, C = 1.0
```

	pred_MaritalStatus	MaritalStatus
7	Single	Divorced
14	Single	Single
17	Divorced	Divorced
18	Divorced	Married
20	Married	Divorced

```
Percentage of correct predictions is
0.673469387755102
```

```
solver='newton-cg', multi_class = 'ovr', penalty = 'l2', max_iter =
1000, C= 10
```

	pred_MaritalStatus	MaritalStatus
7	Single	Divorced
14	Single	Single
17	Divorced	Divorced
18	Divorced	Married
20	Married	Divorced

```
Percentage of correct predictions is
0.673469387755102
```

```
solver='liblinear', multi_class = 'ovr', penalty = 'l2', max_iter =
1000, C= 1
```

	pred_MaritalStatus	MaritalStatus
7	Single	Divorced
14	Single	Single
17	Divorced	Divorced
18	Divorced	Married
20	Married	Divorced

```
Percentage of correct predictions is
0.6700680272108843
```

```
solver='liblinear', multi_class = 'ovr', penalty = 'l2', max_iter =
1000, C= 10
```

	pred_MaritalStatus	MaritalStatus
7	Single	Divorced
14	Single	Single
17	Divorced	Divorced
18	Divorced	Married
20	Married	Divorced

```
Percentage of correct predictions is
0.673469387755102
```

solver='liblinear', multi_class = 'ovr', penalty = 'l1', max_iter = 1000, C= 10

	pred_MaritalStatus	MaritalStatus
7	Single	Divorced
14	Single	Single
17	Divorced	Divorced
18	Divorced	Married
20	Married	Divorced

```
Percentage of correct predictions is
0.673469387755102
```

solver='liblinear', multi_class = 'ovr', penalty = 'l1', max_iter = 1000, C= 1)

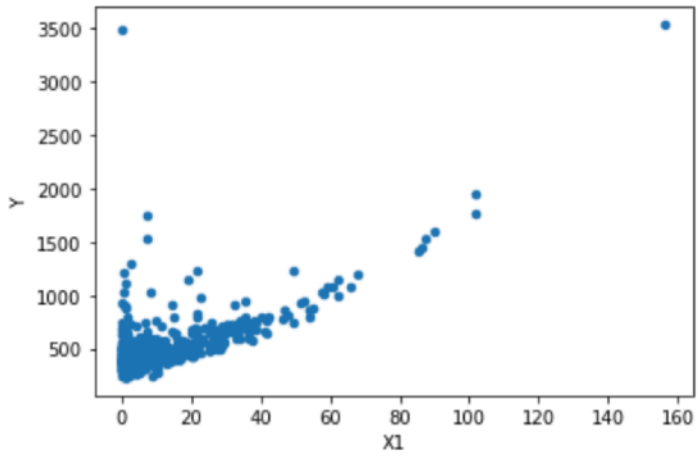
	pred_MaritalStatus	MaritalStatus
7	Single	Divorced
14	Single	Single
17	Divorced	Divorced
18	Divorced	Married
20	Married	Divorced

```
Percentage of correct predictions is
0.6700680272108843
```

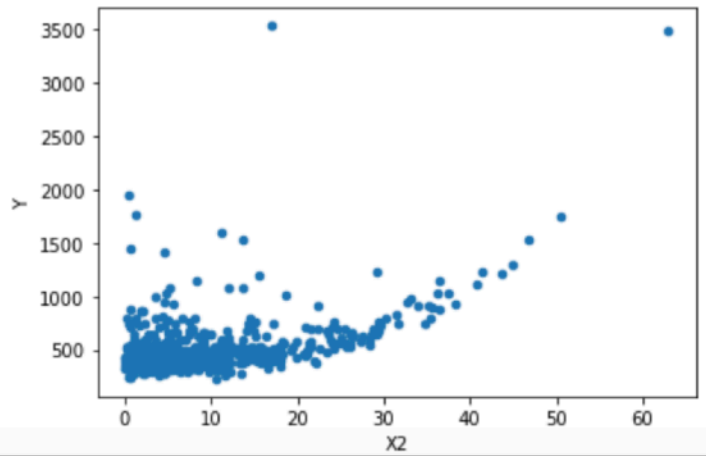
With penalty the prediction accuracy is higher, but L1 and L2 penalty don’t have significant difference in accuracy and also C values.

4.
(a)

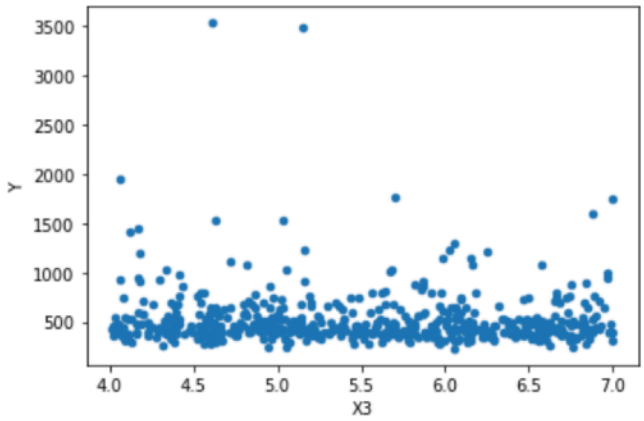
```
<AxesSubplot:xlabel='X1', ylabel='Y'>
```



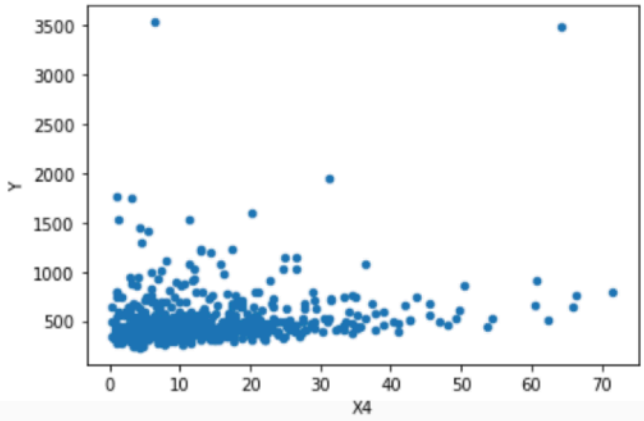

```
<AxesSubplot:xlabel='X2', ylabel='Y'>
```



4]: <AxesSubplot:xlabel='X3', ylabel='Y'>



4]: <AxesSubplot:xlabel='X4', ylabel='Y'>



(b)

Train length: 452 Test length:113

Index:363 Index:307

(c)
R^2. 0.7135

Beta1: 10.58 Beta2: 16.36 Beta3: 2.31 Beta4: 3.79

```
LinearRegression()

0.7135508460514826

array([10.58004134, 16.36289846,  2.31514097,  3.79524388])

173.4788609504679
```

(d)
Mean absolute error is
87.0872695766073
Fraction MAE is
0.16810263818922502

The prediction is pretty accurate. The value difference won't normally exceed 100.

(e)
The model prediction performs very well.
Compare to not using polynomials, the prediction is much more accurate. The correctness raise to 0.97, compare to 0.71 previously.

	predict Y	Y
363	365.08	388.71
176	436.22	505.72
192	349.74	366.29
77	603.99	661.84
320	412.65	473.04

```
Mean absolute error is
41.422900538736215
Fraction MAE is
0.07995771248616405
Percentage of correct predictions is
0.976505646183937
```

: LinearRegression()

: 0.9645039503735615

: array([4.86945856, 2.12144482, 0.09533217, 0.01034503, 0.10109484])

: 332.70512196470736