سوال 1)

1) در حالت اول ما داده های تست و آموزش را 200 گرفتیم، در این حالت بعد از آموزش روی 200 داده آموزش، مدل 200 داده ی دیگر را حدس میزند و به دقت 80 رسیده است. در این حالت چون تعداد داده های تست برابر با داده های آموزش بوده است احتمال اینکه مدل overfit شده باشد بسیار پایین است و این دقت درواقع دقت درستی است و اگر تعداد داده ها بیشتری برای آموزش استفاده می کردیم احتمالا به دقت بالا تری می رسید و این روش نسبت به حالت دوم بهینه تر است

در حالت دوم فقط 20 داده تست داریم و مدل روی 380 داده ی آموزش ترین می شود در این حالت احتمال overfit بسیار زیاد است چرا که مدل بیشتر داده های آموزش را دیده است با اینکه دقت مدل 90 درصد است نمی توان به طور قطع گفت مدل درست کار می کند چرا که از تعداد داده های کمی برای تست استفاده شده است. و این حالت بهینه نیست به علت مشکل overfit.

حالت بهینه می تواند حالتی بین این دو باشد ک مثلا 20 تا 30 درصد را برای تست جدا کنیم در این صورت هم احتمال overfit کم می شود هم مدل می تواند به دقت مناسبی برسد.

2) inference به معنی پروسه ی نتیجه گیری و استنباط خواص یک جمعیت یا احتمال یک سری داده است. در این پروسه با استفاده از دانش های اولیه و دانش های کلی سعی می کنیم خواص یا لیبل یا خروجی یک سری داده را حدس بزنیم برای مثال داریم:

$$\left.\begin{matrix} All\ humans\ are\ mortal \\ All\ Greeks\ are\ humans \end{matrix}\right\} \Longrightarrow All\ Greeks\ are\ mortal$$

$$\left.\begin{matrix} All\ meat\ comes\ from\ animas \\ All\ beaf\ is\ meat \end{matrix}\right\} beaf\ comes\ from\ animals$$

سوال 2)

Gradient Decent یک الگوریتم بهینه سازی مرتبه تکراری مرتبه اول برای یافتن مقدار مینیموم محلی یک تابع دیفرانسیلی است. برای پیدا کردن مقدار کمینه ما از این روش در کم کردن تابع هزینه یمان استفاده می کنیم و در هر مرحله به سمت نقطه مینیموم که همان مقدار منفی gradient است حرکت می کنیم.

$$Cost\ Function: J(\theta) = \frac{1}{2}\sum_{i=1}^{q}\left(h_\theta\left(x^i\right) - y^i\right)^2$$

$$h(x) = \tanh(w^T x + b)$$

$$Gradient\ Function: \frac{\delta J(\theta)}{\delta w} = \frac{\delta\left(\frac{1}{2}(\tanh(w^T x+b)- y^i)^2\right)}{\delta w}$$

$$= \left(\tanh(w^T x + b) - y^i\right)\left(1 - tanh^2(w^T x + b)\right)\frac{\delta wx}{\delta w}$$

$$= \left(\tanh(w^T x + b) - y^i\right)\left(1 - tanh^2(w^T x + b)\right)x$$

$$\Delta W = -\eta\frac{\delta J}{\delta w}$$

$$= -\eta x\left(\tanh(w^T x + b) - y^i\right)\left(1 - tanh^2(w^T x + b)\right)$$

$$\Delta b = -\eta\frac{\delta J}{\delta b}$$

$$= -\eta\left(\tanh(w^T x + b) - y^i\right)\left(1 - tanh^2(w^T x + b)\right)$$

که دو فرمول بالا همان قانون آپدیت کردن وزن ها و بایاس است و در آن eta همان نرخ یادگیری ما هست.
همان طور که مشخص است نرخ یادگیری به صورت یک ضریب برای فرمول ما ظاهر شده است که این نشان می دهد هر چه مقدار نرخ ما بیشتر باشد اختلاف وزن جدید با قدیم بیشتر می شود در نتیجه تغییر بزرگ تری در تغییر وزن ها اتفاق می افتاد که ما را به سمت اکسترموم محلی حرکت می دهد و هر چه کمتر باشد با سرعت کمتری حرکت اتفاق می افتد که سریع رفتن ممکن است باعث پریدن از نقطه اکسترموم محلی شود و مدل همگرا نشود و با سرعت کم رفتن ممکن است با عث نرسیدن به نقطه و طولانی شدن زمان آموزش مدل شود، باید یک مقدار برای مدل که این 2 مشکل را نداشته باشد پیدا کنیم.

سوال 3)

ما در اینجا n+1 نقطه داریم برای اینکه از هر نقطه یک چند جمله ای داشته باشیم می توانیم انها را به صورت n معادله جدا از هم بنویسیم که داریم:

$$\begin{cases} a_n x_1^n + a_{n-1} x_1^{n-1} + \cdots + a_1 x_1 + a_0 = y_1 \\ \vdots \\ a_n x_n^n + a_{n-1} x_n^{n-1} + \cdots + a_1 x_n + a_0 = y_n \end{cases}$$

که این سیستم را می شود به صورت زیر نوشت:

$$\begin{pmatrix} x_1^n & x_1^{n-1} & \cdots & x_1 & 1 \\ & \vdots & \ddots & \vdots & \\ x_n^n & x_n^{n-1} & \cdots & x_n & 1 \end{pmatrix} \begin{pmatrix} a_n \\ \vdots \\ a_0 \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

چون ماتریس بالا singular نیست یعنی دترمینان آن مخالف 0 است حتما یک جواب یکتا و unique دارد (ماتریس بالا ماتریس vandermonde است که ویژگی ان دترمینان مخالف 0 است) این چند جمله حتما از درجه کمتر از n+1 است که می شود حتما یک چند جمله ای از درجه n و یا کمتر و این جواب یکتاست.

سوال 4)

Falat Pejvak Co.
Inspection & Certification

Subject: Date:

$\beta_0 = \bar{Y} - \beta_1 \bar{X} \qquad \beta_1 = \frac{\sum((X_i - \bar{X})(Y_i - \bar{Y}))}{\sum_i (X_i - \bar{X})}$

$SXX = \sum_{i=1}^{n} (x_i - \bar{x})^2 \qquad SYY = \sum_{i=1}^{n} (y_i - \bar{y})^2 \qquad SXY = \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$

$\Rightarrow Var(\beta_1) = Var\left(\frac{SXY}{SXX}\right) = Var\left(\frac{\sum(x_i - \bar{x}) y_i}{SXX}\right) = \frac{1}{SXX^2} \sum x_i - \bar{x}$

$= \frac{\sigma^2}{SXX}$

$Var(\beta_0) = Var(\bar{y} - \beta_1 \bar{x}) = Var\left(\sum\left(\frac{1}{n} - \frac{(x_i - \bar{x})\bar{x}}{SXX}\right) y_i\right)$

$= \sum\left(\frac{1}{n} - \frac{x_i - \bar{x}}{SXX}\bar{x}\right)^2 \sigma^2 = \sum\left[\frac{1}{n^2} + \frac{SXX + \bar{x}^2}{SXX^2} - \frac{2}{n}\frac{\bar{x}(x_i - \bar{x})}{SXX}\right]\sigma^2$

$= \left[\frac{1}{n} + \frac{n\bar{x}}{SXX}\right]\sigma^2 = \frac{\sum x_i^2}{n * SXX}\sigma^2$

$Cov(y_i, \beta_1) = \frac{x_i - \bar{x}}{SXX} Var(y_{01}) = \frac{x_i - \bar{x}}{SXX}\sigma^2$

$Cov(y_i, \beta_0) = Cov(y_{01}\bar{y} - \beta_1 \bar{x}) = \frac{\sigma^2}{n} + \bar{x}\frac{x_i - \bar{x}}{SXX}\sigma^2$

$\Rightarrow Cov(\beta_0, \beta_1) = Cov(\bar{y} - \beta_1 \bar{x}, \beta_1) = Cov\left(\frac{\sum y_i}{n} - \sum\frac{(x_i - \bar{x})\bar{x}}{SXX} y_i, \sum\frac{(x_i - \bar{x}) y_i}{SXX}\right)$

$= \sum\left(\frac{1}{n} - \frac{x_i - \bar{x}}{SXX}\bar{x}\right)\frac{x_i - \bar{x}}{SXX}\sigma^2 = -\frac{\bar{x}}{SXX}\sigma^2$

β_0, β_1 مستقل $\Rightarrow Cov(\beta_0, \beta_1) = 0 \Rightarrow \bar{x} = 0 \rightarrow$ مماسی باشد

سوال 5)

Falat Pejvak Co.
Inspection & Certification

Subject: Date:

$\bar{x} = \frac{\sum x_i}{8} = 2.25 \qquad \bar{y} = \frac{\sum y_i}{8} = 42.125$

$\beta_0 = \bar{y} - \beta_1 \bar{x} \qquad \beta_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$

$\beta_1 = \frac{5.75}{10.5} = 0.575 \qquad \beta_0 = 42.125 - 0.575 \times 2.25 = 40.83$

$\sigma_x^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1} = \frac{10.5}{7} = 1.5 \qquad S_y^2 = \frac{10.875}{7} = 1.553$

سوال 6)

در این سوال می خواهیم 2 تیم را با توجه به رنگ های انها از هم طبقه بندی کنیم برای اینکار داریم:

```
from collections import Counter

target_names = ['red', 'blue']
train = images
targets = labels
predict_targets = ['red' if average[0] > average[1] else 'blue' for average in averages]

print(len(train), len(target), len(predict_targets))
print(Counter(targets).keys(), Counter(targets).values())
print(Counter(predict_targets).keys(), Counter(predict_targets).values())
```

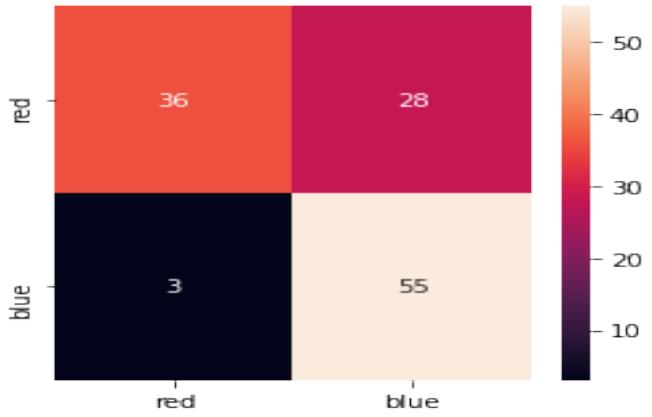
```
122 122 122
dict_keys(['blue', 'red']) dict_values([64, 58])
dict_keys(['blue', 'red']) dict_values([39, 83])
```

ابتدا تمامی عکس ها را خوانده و با توجه به تیم آن ها، آن ها را لیبل گزاری می کنیم

سپس هر عکس را با توجه به اینکه مقدار قرمز آن بیشتر از آبی است پیش بینی می کنیم که در کدام کلاس قرار دارند و خروجی ها را بدست می اوریم و داریم:

	precision	recall	f1-score	support
red	0.92	0.56	0.70	64
blue	0.66	0.95	0.78	58
accuracy			0.75	122
macro avg	0.79	0.76	0.74	122
weighted avg	0.80	0.75	0.74	122

همان طور که می بینیم دقت 75% درصد و پارامتر های دیگر نیز قابل مشاهده است. و ماتریس آن به شکل:



که همان طور که می بینیم 36 عدد از کلاس قرمز و 55 عدد از کلاس آبی به درستی تشخیص داده شده اند. 28 عدد از کلاس قرمز به اشتباه کلاس آبی و 3 عدد از کلاس آبی، قرمز تشخیص داده شده اند. طبق همین اعداد می بینیم که مقدار recall برای کلاس قرمز باید کم تر باشد چرا که تعداد زیادی از این کلاس به اشتباه طبقه بندی شده است و در عین حال precision کلاس آبی نیز باید کم باشد چرا که اکثر داده ها کلاس آبی تشخیص داده شده است که تعداد کمی از انها درست بوده است.

سوال 7)

در این سوال ابتدا مقادیر نقاط را به صورت رندوم بدست می اوریم و یک سری نقاط در نمودار بوجود می آوریم.

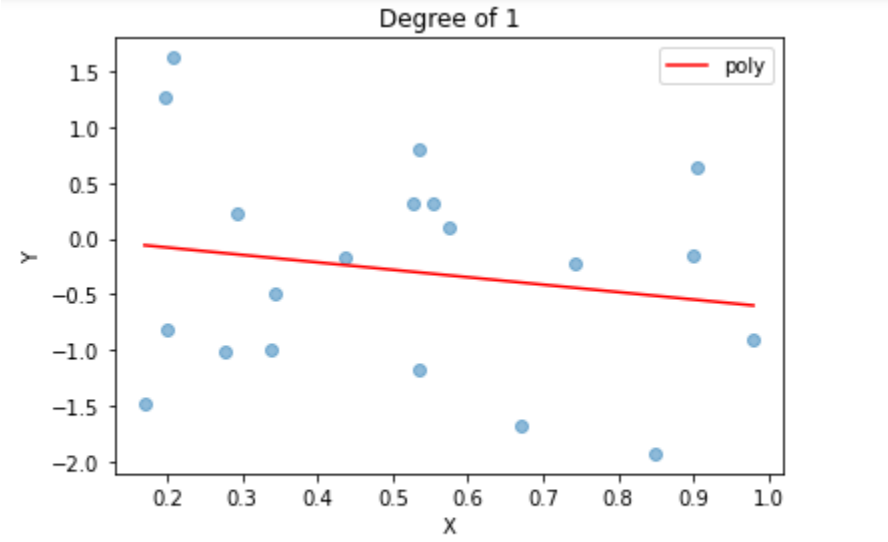
```
def f(x):
    return np.cos(2 * np.pi * x)
npoints = 20
x = np.random.rand(npoints)[:, np.newaxis]
y = f(x) + np.random.normal(0, 1, npoints)[:, np.newaxis]
```

که در اینجا به تعداد 20 نقطه درست کرده ایم.

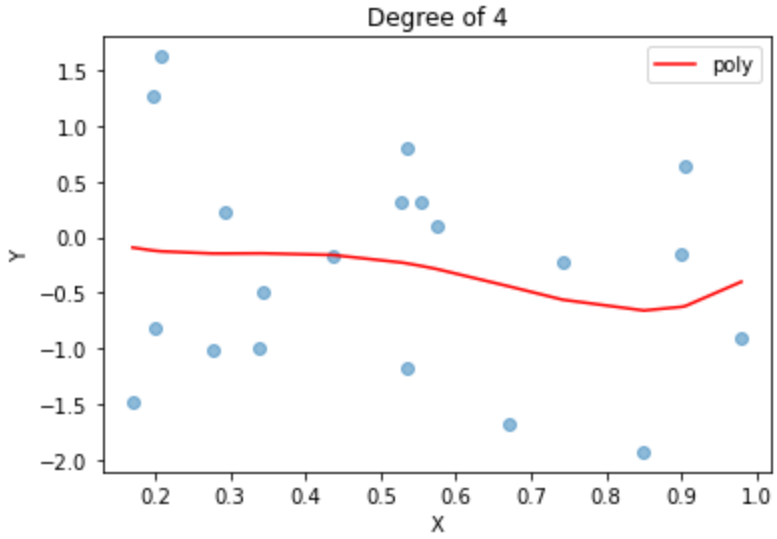
حال با استفاده از توابع سعی می کنیم خط هایی با درجه های مختلف بر روی این نقاط آموزش دهیم و داریم:

```
degrees = [1 ,4, 8 ,12, 15, 20]
for i, d in enumerate(degrees):
    poly_model = make_pipeline(PolynomialFeatures(degree=d), linear_model.LinearRegression())
#     mse, bias, var = bias_variance_decomp(poly_model, x_train, y_train, x_test, y_test, loss='
    poly_model.fit(x_train, y_train)
```

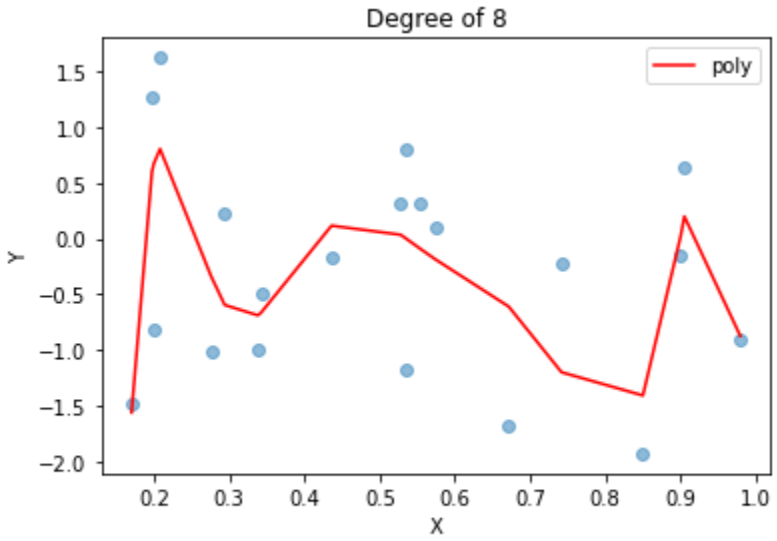
حال برای هر یک از درجه ها داریم:



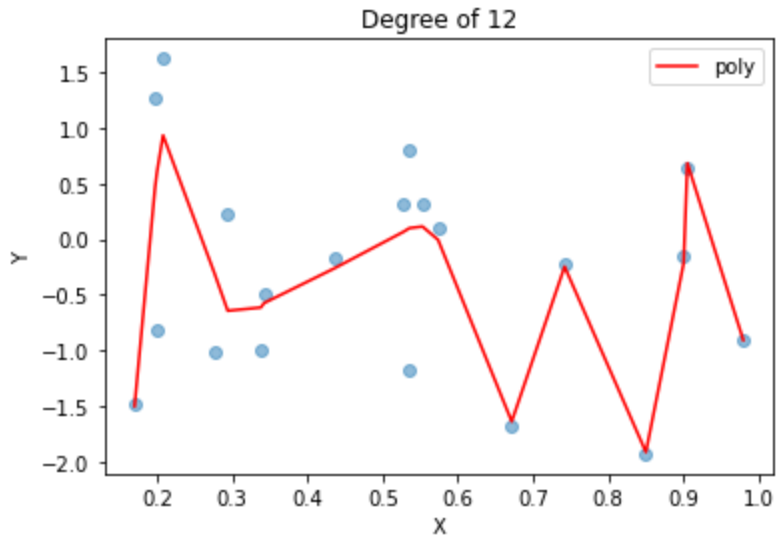

```
Mean Square Error for degree:  1  Is:  0.8619047330753362
```



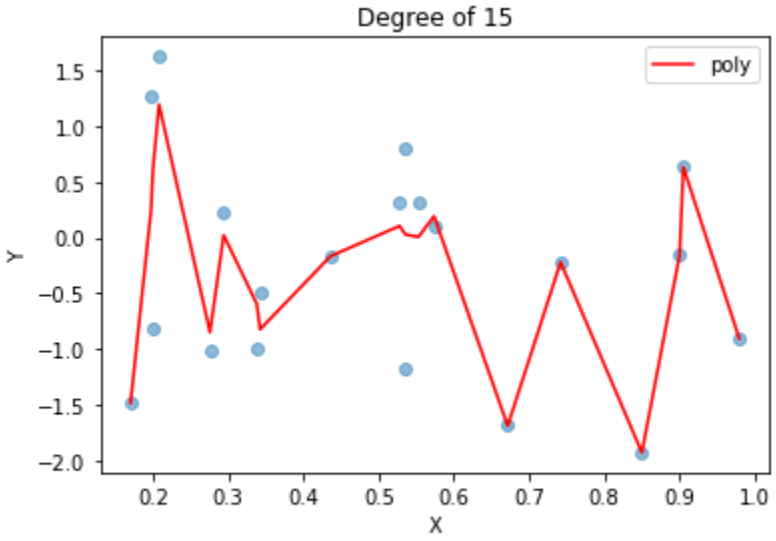

```
Mean Square Error for degree:  4  Is:  0.8558016355589079
```



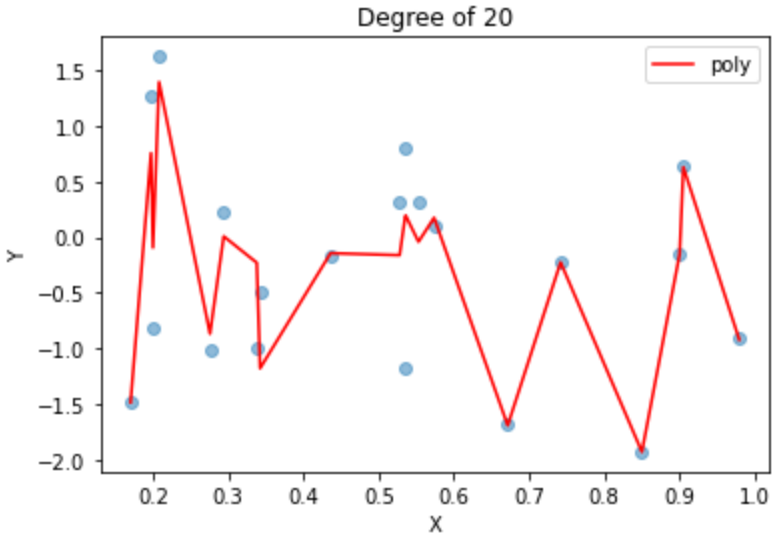

```
Mean Square Error for degree:  8  Is:  0.4777019042650602
```




```
Mean Square Error for degree:  12  Is:  0.3373775010911285
```




```
Mean Square Error for degree:  15  Is:  0.2869410787969532
```




```
Mean Square Error for degree:  20  Is:  0.2278920836431868
```

همان طور که می بینیم با زیاد کردن درجه تابع مقدار خطای ما کاهش می یابد چرا که خط بهتر از نقاط می تواند عبور کند ولی با این کار در واقع مدل ما در حال overfit شدن به داده های آموزش است اگر یک سری داده با همین طوزیع احتمالی به مدل بدهیم خطای درجه های بالا تر بیشتر می شود چرا که آن ها فقط برای داده های آموزش می توانند به خوبی عمل کند و در واقع overfit شده است.

در مدل در درجه های ابتدایی که خط در واقع underfit است مقدار بایاس مدل بالاست و در درجه های بیشتر مقدار variance مدل زیاد و bias آن کم می شود که نشان از overfit است.

سوال 8)

فرض می کنیم جدول رو به رو را داریم:

	True Condition	
Predicted Condition	True positive	False Positive
	False Negative	True Negative

مقادیر TP نشان می دهد چند عدد از هر کلاس به درستی پیش بینی شده است، مقدار TN نشان می دهد چند عدد از هرکلاس به درستی در کلاس پیش بینی نشده و دروتقع کلاس دیگری است، FP نشان می دهد چند عدد در هر کلاس به اشتباه در کلاس پیشبینی شده است در حالی که مقدار واقعی آن کلاس دیگری بوده است و FN نشان می دهد در هر کلاس جند عدد واقعا در آن کلاس بوده اند ولی به اشتباه در کلاس دیگر پیش بینی شده است

حال برای مقادیر recall و precision و accuracy داریم:

$$Precision = \frac{tp}{tp + fp}$$

$$Recall = \frac{tp}{tp + fn}$$

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn}$$

در نتیجه precision می شود تعداد داده های همان کلاس که به درستی پیش بینی شده تقسیم بر تعداد داده هایی که در آن کلاس پیشبینی شده.

Recall می شود تعداد داده های آن کلاس که به درستی پیش بینی شده تقسیم برا داده های های آن کلاس که به درستی در همان کلاس یا کلاس های دیگر پیشبینی شده و accuracy می شود تعداد درست پیشبینی تقسیم بر تعداد کل داده ها.

در بعضی مواقع مانند زمانی که تعداد داده های یک کلاس کمتر از دیگری باشد و یا پارامتر tp یا tn مهم تر باشد مانند کار های پزشکی ممکن است یک ویژگی بیشتر بدرد ما بخورد و ما مجبور شویم سراغ بقیه ویژگی ها برویم.