Report Machine Learning A1 – David Valensi 260873585

1 Sampling:

1.1 The pseudocode for sampling activities is :

```
dic= {}

for i in range(n):
    a= random.uniform(0,1)
    if(a< 0.2):
        dic["Movies"] +=1
    elif(a< 0.6):
        day_activity= "COMP-551"
    elif(a< 0.7):
        day_activity= "Playing"
    else:
        day_activity="Studying"
distribution[acitivity] =   dic[acitivity]/n
```

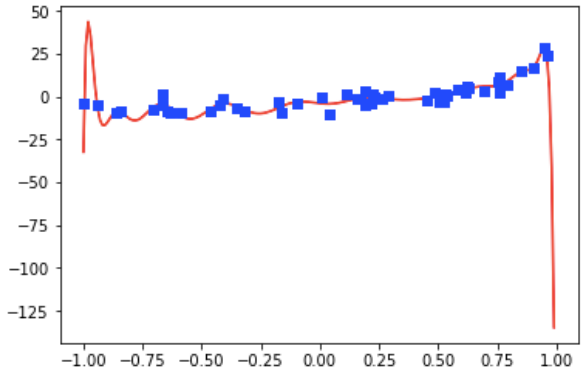
1.2 The fraction of days spent to each activity is closer to the true distribution when we sample over 1000 days than over 100 days The fractions tend to the real distribution:

```
Sampling over 100 random probabilities :
Movies 0.24
COMP-551 0.42
Playing 0.08
Studying 0.26
Sampling over 1000 random probabilities :
Movies 0.228
COMP-551 0.429
Playing 0.111
Studying 0.332
```

2 Model Selection

2.1 b) 20 degree polynomial fit:

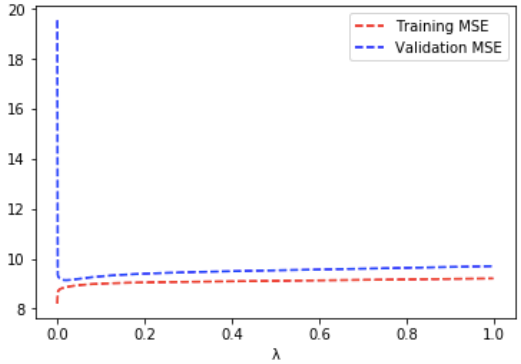
```
Train MSE :  6.474752064154863
MSE Validation:  1421.1635301853185
```



2.1 c) Since the validation MSE is much higher than the training MSE we can infer that there is obviously an overfitting. The reason to that is the learned polynom fits too much the training data and gets too high coefficients and thus fits worst the validation data.

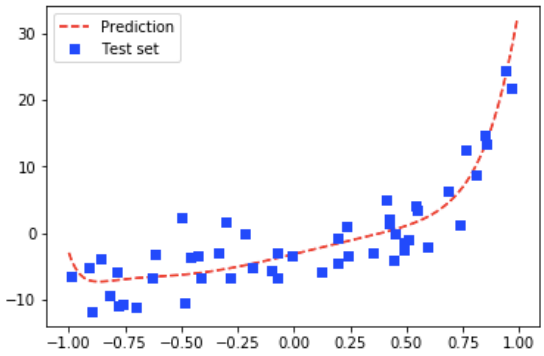
2.2 a) Comparing different lambda parameters, we find out what's the best lambda:

```
The optimal lambda : 0.02001
```



2.2 d) Testing MSE with L2 regularization. The model is not overfitting since the test MSE is not significantly higher than the training and validation MSE. Moreover this seems to be the best fit we can achieve on the test set.

```
Test MSE :  50.82444706148792
```

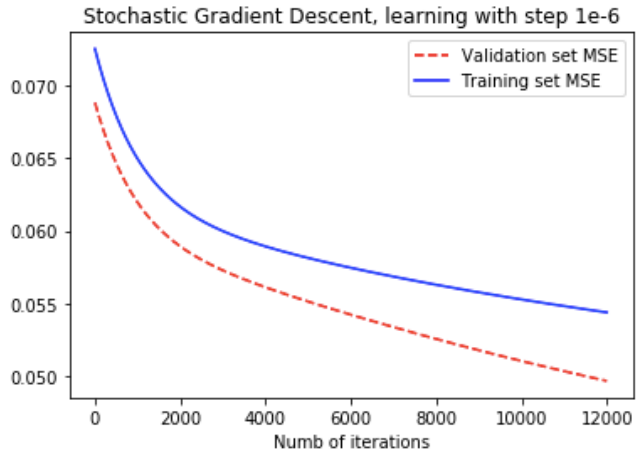


3. The source polynomial seems to be a degree 3 polynomial. The visualization helps us to tell that despite only test set is plotted.

3 Gradient Descent

3.1 Stochastic Gradient Descent with step of 1e-6, learning curve:

```
Validation performance :  0.04966162455023821
```

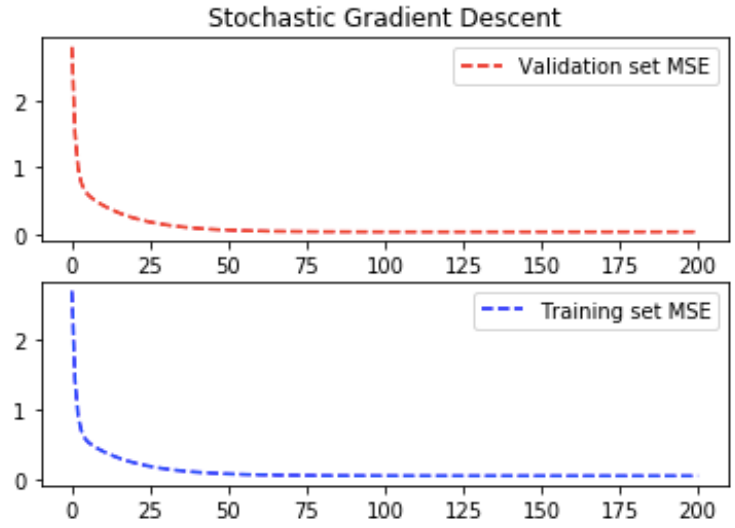


3.2 a)

Step size	Validation set MSE	Number of epochs (until learning saturation)	
1e-8	0.06784069173580148	10000	
1e-7	0.061960201961669334	10000	
1e-6	0.05103253263500413	10000	
1e-5	0.037679672794321044	8000	
1e-4	0.03703665362643937	3000	
1e-3	0.0369352162868476	200	
1e-2	0.03706912992056424	30	(Not very stable)

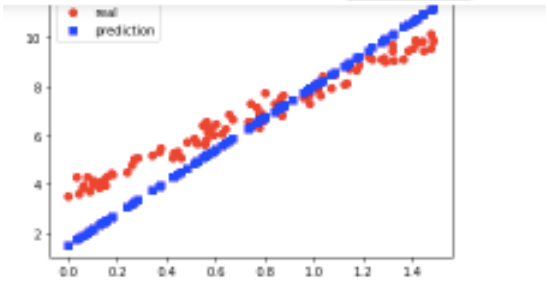
3.2 b) Test MSE of step size 1e-3 : 0.03436316716043181

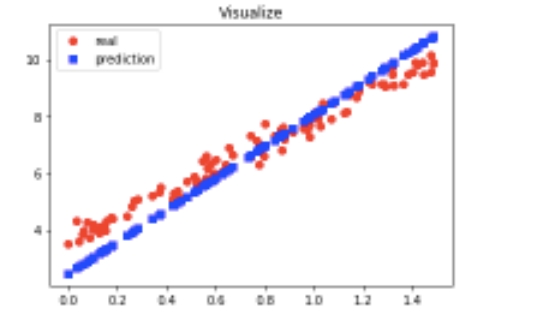
```
Validation performance :  0.036957126701812594
```

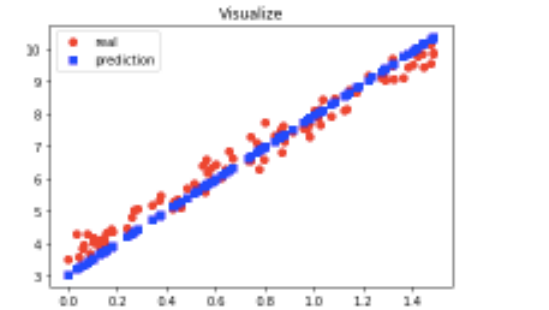


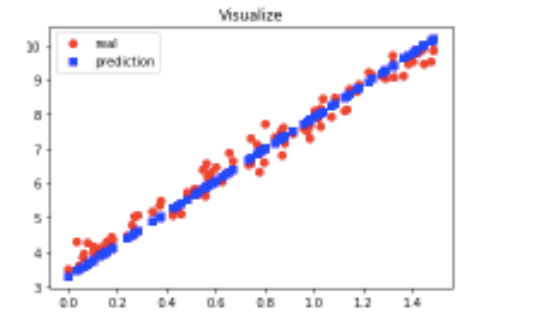

```
Test performance :  0.03467090862244922
```

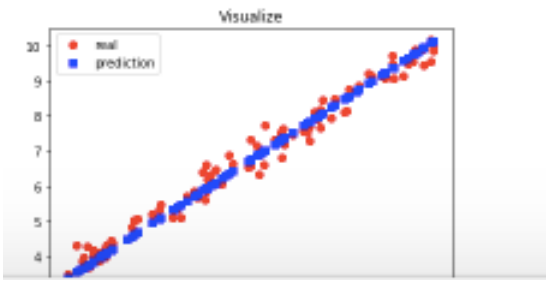
3.3 Visualization of the improvement of parameters with training (5 random epochs):











4 Real Life dataset

a) Filling the missing values with the mean of the column.

This is not a really good solution since the mean is highly influenced by extreme values and if there are a lot of missing values in a column, everyone is going to be replaced by the same mean value. It removes information about diversity and distribution about the original dataset.

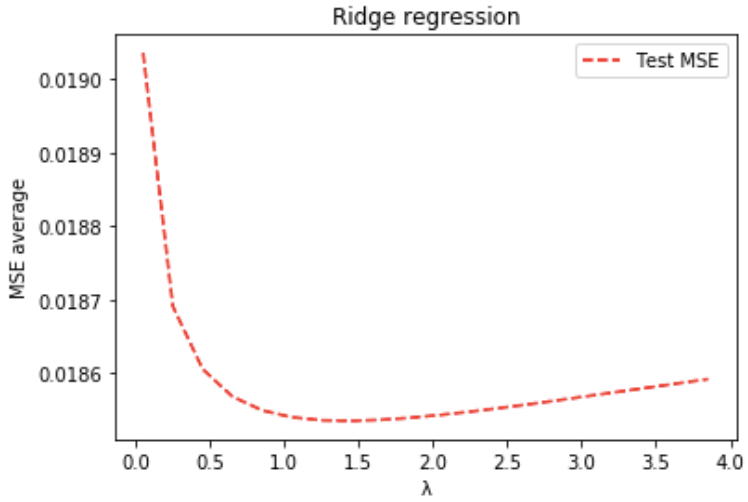
b) We could fill the missing values with other methods.

The median of each column for example, will be more representative of the majority of data than the mean. It may be that the mean and median are the same in some cases. We can also use a random value picked in the column to fill messing values. This will keep the diversity of original dataset but might alter the correlation between features and labels.

```
The averaged MSE over  5-fold cross-validation error is : 0.0194903025306669
```

4.3

a)




```
optimal_avg_mse= 0.018534912884940104    optimal_lambda= 1.4500000000000002
```

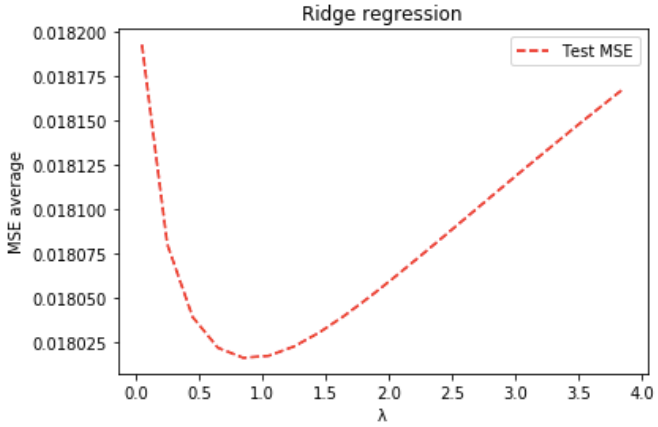
b) We could indeed use these results for **feature selection**. We learned that the Ridge regression is in fact constraining the coefficient in a multi-dimensional sphere of radius λ. Then, we can use this value to restrain some features which lead to too low coefficient is the W matrix. If we select only features which lead to consisting coefficients in W, we could select only meaningful features.

c) I set a minimal threshold to the coefficient matrix such that I selected only meaningful features.

The regular linear regression gives similar results than with the complete dataset:

```
The averaged MSE over  5-fold cross-validation error is : 0.018266029094430033
```

On the other side, Ridge regression, after trying several regularization rate, gives slightly better fit :
Avg MSE: 0.018016167942022557




```
optimal_avg_mse= 0.018016167942022557   optimal_lambda= 0.8500000000000001
```

d) We can achieve similar results, even maybe better ones by reducing the number of features. This is done by selecting meaningful features i.e. features which are multiplied by a consisting coefficient. This is possible since there was 123 features originally which contained highly non correlated features. These features didn't help us predicting the output, then we can do without.