main_auto

October 22, 2023

1 Advanced Java & Advanced Python Assignment

1.1 Deng Chuan Chang | Yasser El Karkouri | Julien Godfroy

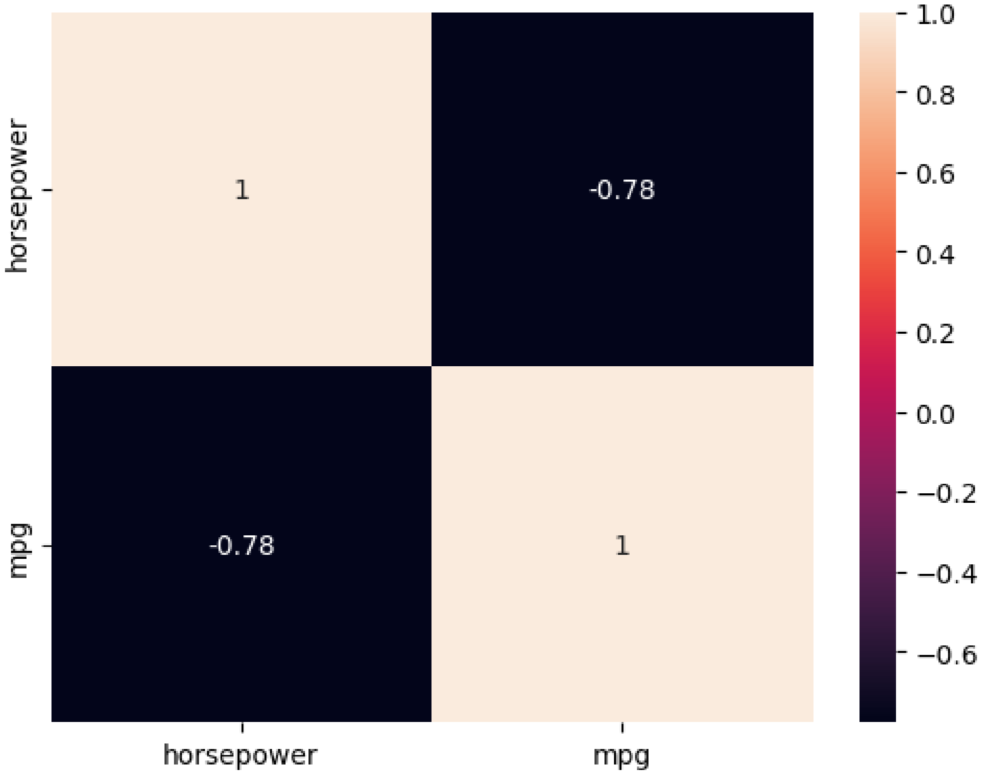
1.1.1 Auto Dataset

[ ]:
```
from  Class.ModelClass import * # Importing the Model class from ModelClass.py
from functions.utils import * # Importing the utils functions from utils.py
```

Let's start with a quick data analysis

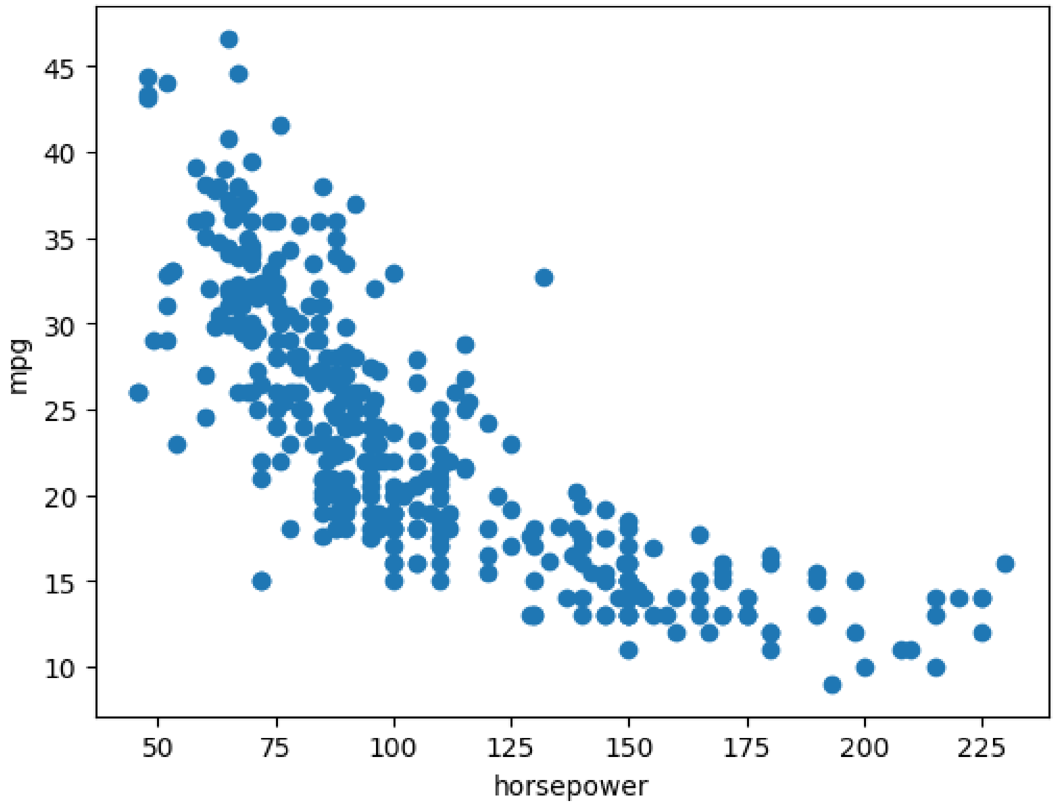
[ ]:
```
x, y, df = import_clean_data('./data/Auto.csv', input_list=['horsepower'],␣
  ↪output_list=['mpg'])
df.head()

#heatmap for the correlation coefficient between the variables
import seaborn as sns
sns.heatmap(df.corr(), annot=True)
```

[ ]:
```
<AxesSubplot: >
```



```
[ ]: #plot correlation mpg and horsepower
     plt.scatter(x, y)
     plt.xlabel('horsepower')
     plt.ylabel('mpg')
```

```
[ ]: Text(0, 0.5, 'mpg')
```



Import the data, clean it, and prepare the input and output vectors

```
[ ]: x, y, df = import_clean_data('./data/Auto.csv', ['horsepower'], ['mpg'])
     X, y = prepare_vectors(x, y)
```

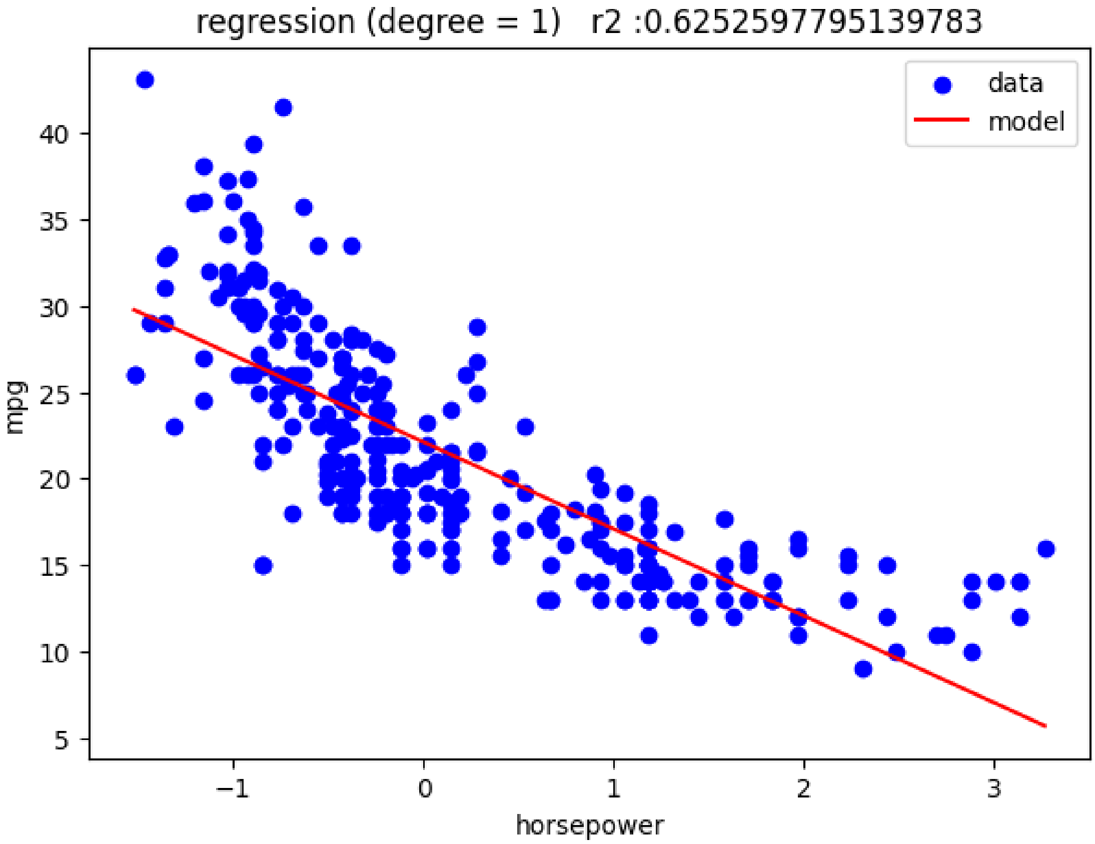
For each degree, compute the regression

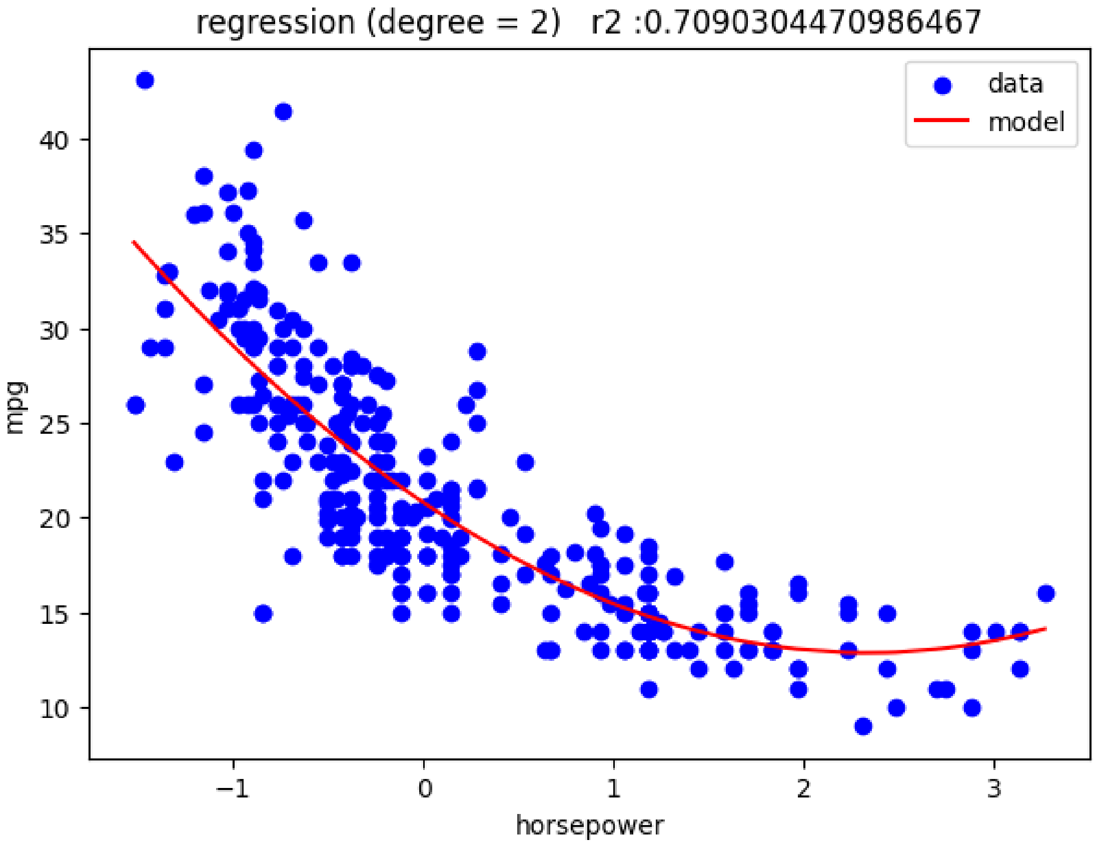
```
[ ]: degrees_list = [1, 2, 5]

     model_list = []
     for i, degree in enumerate(degrees_list):
         model = Model(X, y, learning_rate=0.0008, iterations=6000, test_size=0.2)  ␣
      ↪#Create the model with the chosen parameters
         model.transform(degree)              #IMPORTANT STEP : Transforming X vector␣
      ↪into polynomial vector
         model.compute_regression()
         model_list.append(model)
```

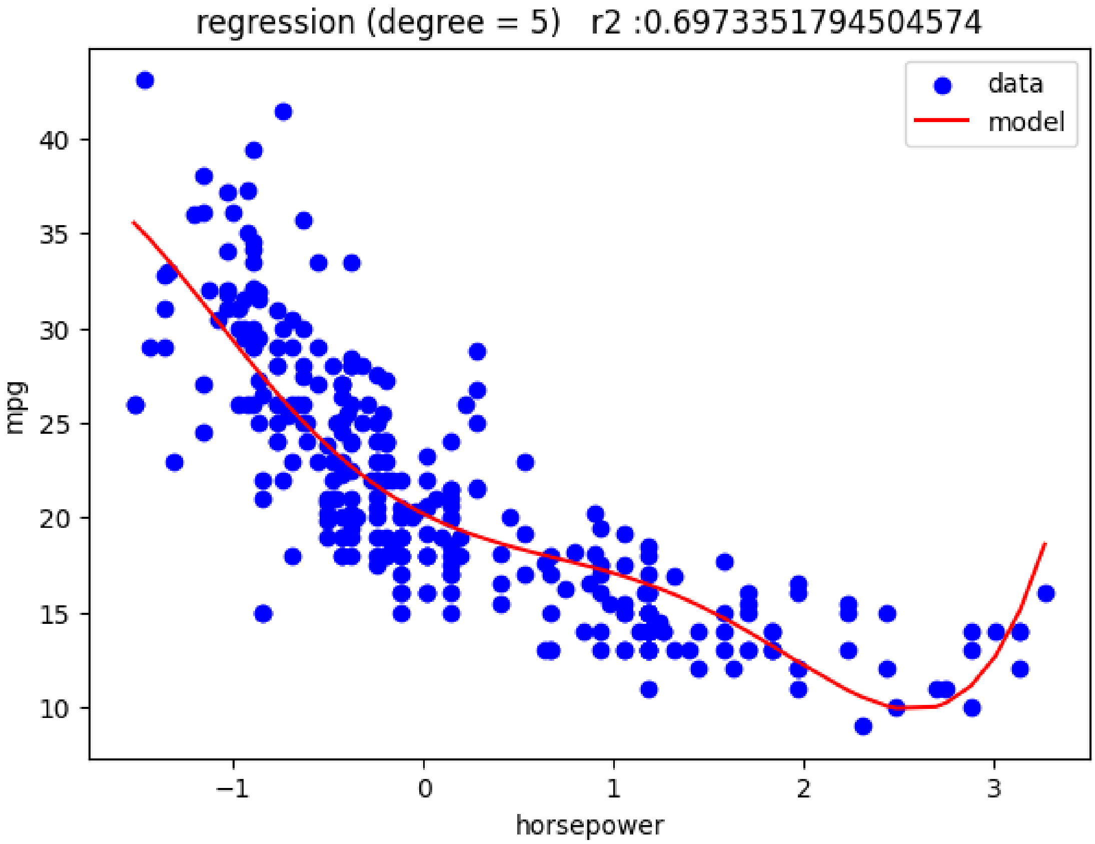
For each model, compute the regression and plot the results and r2 score

```
[ ]: for i, model in enumerate(model_list):
         model.compute_r_square()
```

```
    model.plot_regression_2D('horsepower', 'mpg', 'regression (degree = {})'.
↪format(degrees_list[i]))
```

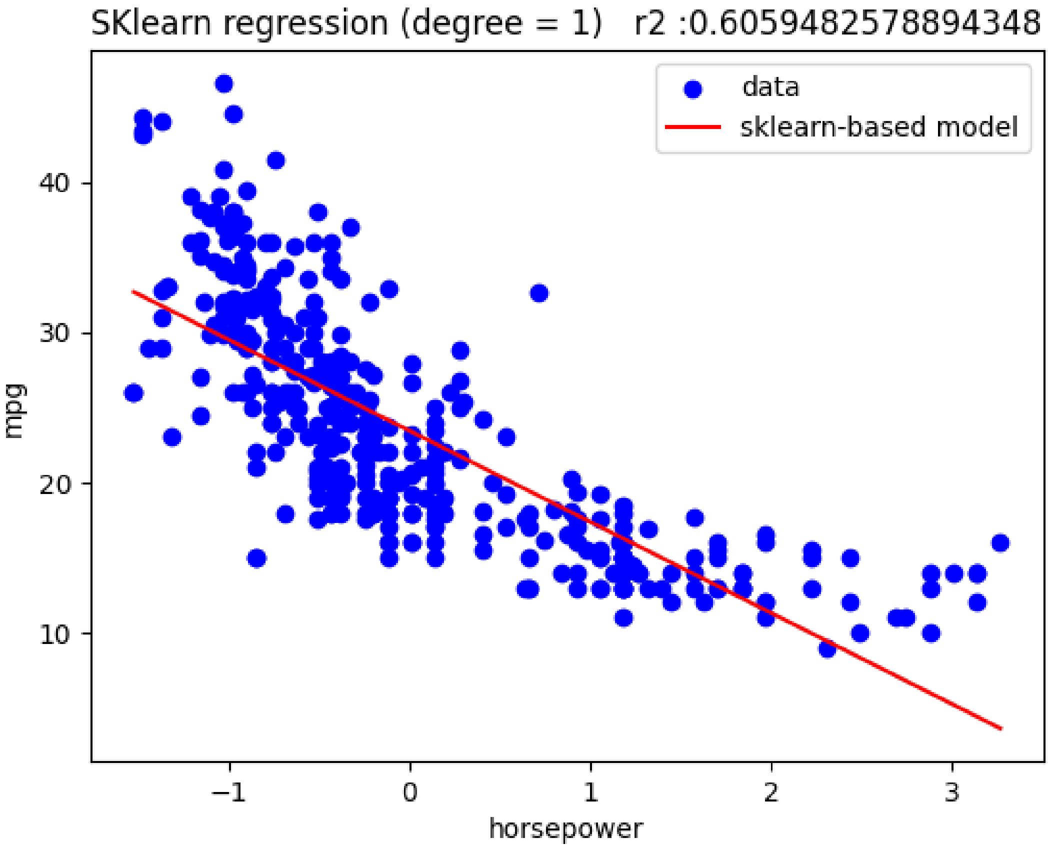


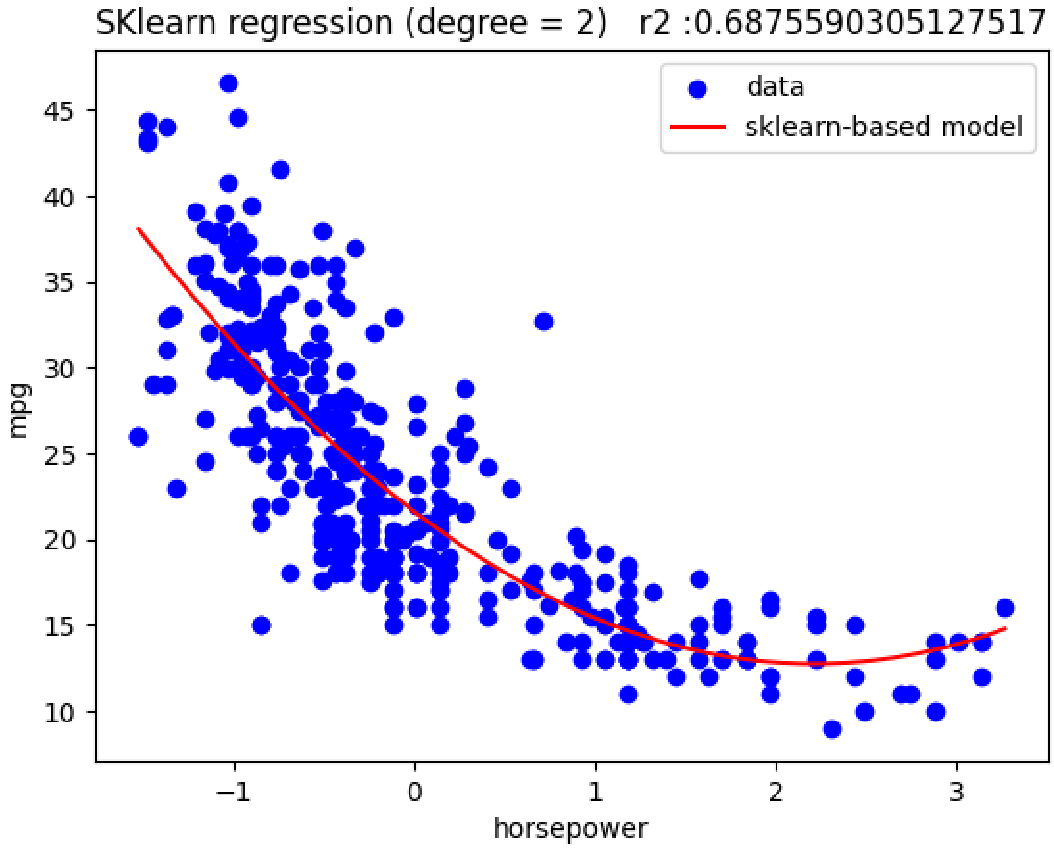

regression (degree = 2) r2 :0.7090304470986467
data
model
mpg
40
35
30
25
20
15
10
−1
0
1
2
3
horsepower

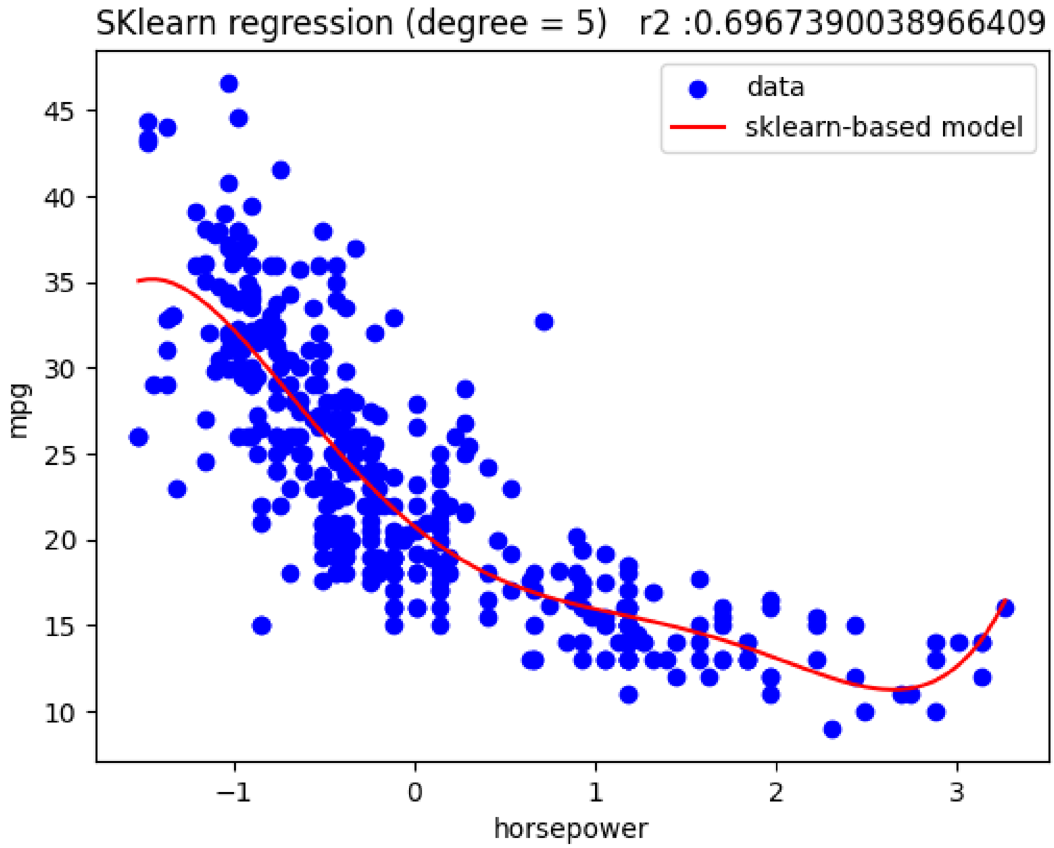


Check the previous results with the sklearn library

[ ]:
```
#Do the same with sklearn

for i, degree in enumerate(degrees_list):
    sk_compute_plot(X, y, degree, 'horsepower', 'mpg', 'SKlearn regression␣
 ↪(degree = {})'.format(degrees_list[i]))
```


SKlearn regression (degree = 1)   r2 :0.6059482578894348
data
sklearn-based model
mpg
40
30
20
10
−1
0
1
2
3
horsepower


SKlearn regression (degree = 2)   r2 :0.6875590305127517
data
sklearn-based model
mpg
45
40
35
30
25
20
15
10
−1
0
1
2
3
horsepower


SKlearn regression (degree = 5)   r2 :0.6967390038966409
data
sklearn-based model
mpg
45
40
35
30
25
20
15
10
−1
0
1
2
3
horsepower