Московский государственный технический университет им. Н. Э. Баумана

***Факультет** «Информатика и системы управления»*

***Кафедра** «Системы обработки информации и управления»*

Отчёт по лабораторной работе №3

*по курсу «**Введение в машинное обучение**»*

Исполнила: Алиева Д.Г., ИУ5-41
Проверил: Гапанюк Ю.Е.

Москва 2018 г.

Условие ЛР: Необходимо решить задачу предсказания стоимости дома в зависимости от его характеристик.

1. Провести предподготовку данных
2. Разделить данные
3. Обучить модель из sklearn
4. Реализовать линейную регрессию
5. Эксперименты с моделью

Код и результат работы программы:

In [16]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from scipy import stats
from sklearn.cross_validation import train_test_split
from sklearn import linear_model
from sklearn.metrics import mean_absolute_error, r2_score
%matplotlib inline
```

In [14]:
```
data = pd.read_csv('./KaggleLab3/train.csv')
```

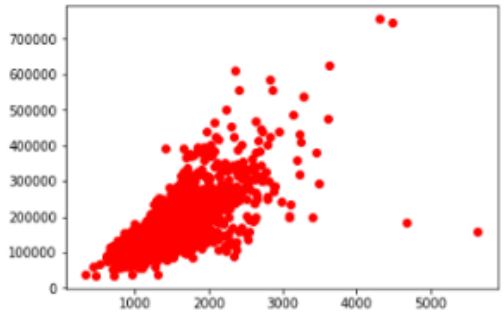
In [11]:
```
data
```

Out[11]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscV
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	
5	6	50	RL	85.0	14115	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	MnPrv	Shed	70
6	7	20	RL	75.0	10084	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	
7	8	60	RL	NaN	10382	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	Shed	35
8	9	50	RM	51.0	6120	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	
9	10	190	RL	50.0	7420	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	
10	11	20	RL	70.0	11200	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	

In [26]:
```
#распредеение площади домов и цен
plt.plot(data['GrLivArea'], data['SalePrice'], 'ro')
```

Out[26]:
```
[<matplotlib.lines.Line2D at 0x25a665920b8>]
```



In [31]:
```
def cleaning(data):
    categorical_columns = [c for c in data.columns if data[c].dtype.name == 'object']
    numerical_columns   = [c for c in data.columns if (data[c].dtype.name != 'object' and c != 'SalePrice')]
    answer_column = [c for c in data.columns if c == 'SalePrice']
    #заполняем пустые количественные медианным значением
    data = data.fillna(data.median(axis=0), axis=0)
    #заполняем пустые категориальные самым частым значением по признаку
    data_describe = data.describe(include=[object]) #получение сводной информации по таблице
    for c in categorical_columns:
        data[c] = data[c].fillna(data_describe[c]['top']) #fillna() - метод для замены отсутствующих значений на числовые
    #начинаем векторизацию - переводим категориальные признаки в количественные
    binary_columns    = [c for c in categorical_columns if data_describe[c]['unique'] == 2] #бинарные
    nonbinary_columns = [c for c in categorical_columns if data_describe[c]['unique'] > 2] #небинарные
    for c in binary_columns:
        top = data_describe[c]['top']
        top_items = data[c] == top
        data.loc[top_items, c] = 0
        data.loc[np.logical_not(top_items), c] = 1
        data_nonbinary = pd.get_dummies(data[nonbinary_columns]) #возврат нового столбца для каждого элемента
    #начинаем нормализацию количественных признаков
    data_numerical = data[numerical_columns]
    data_numerical = (data_numerical - data_numerical.mean()) / data_numerical.std()
    data_answer = data[answer_column] #не требуется нормализация
    #соединяем всё в таблицу
    data = pd.concat((data_numerical, data[binary_columns], data_nonbinary, data_answer), axis=1)
    data = pd.DataFrame(data, dtype=float)
    return data
```

In [41]:
```
data.corr()
```

Out[41]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	...	Sale
Id	1.000000	0.011156	-0.009921	-0.033226	-0.028365	0.012609	-0.012713	-0.021998	-0.051071	-0.005024	...	
MSSubClass	0.011156	1.000000	-0.356718	-0.139781	0.032628	-0.059316	0.027850	0.040581	0.023573	-0.069836	...	
LotFrontage	-0.009921	-0.356718	1.000000	0.304522	0.234812	-0.053281	0.116685	0.083348	0.178469	0.214367	...	
LotArea	-0.033226	-0.139781	0.304522	1.000000	0.105806	-0.005636	0.014228	0.013788	0.103321	0.214103	...	
OverallQual	-0.028365	0.032628	0.234812	0.105806	1.000000	-0.091932	0.572323	0.550684	0.407252	0.239666	...	
OverallCond	0.012609	-0.059316	-0.053281	-0.005636	-0.091932	1.000000	-0.375983	0.073741	-0.125694	-0.046231	...	
YearBuilt	-0.012713	0.027850	0.116685	0.014228	0.572323	-0.375983	1.000000	0.592855	0.311600	0.249503	...	
YearRemodAdd	-0.021998	0.040581	0.083348	0.013788	0.550684	0.073741	0.592855	1.000000	0.176529	0.128451	...	
MasVnrArea	-0.051071	0.023573	0.178469	0.103321	0.407252	-0.125694	0.311600	0.176529	1.000000	0.261256	...	
BsmtFinSF1	-0.005024	-0.069836	0.214367	0.214103	0.239666	-0.046231	0.249503	0.128451	0.261256	1.000000	...	
BsmtFinSF2	-0.005968	-0.065649	0.042463	0.111170	-0.059119	0.040229	-0.049107	-0.067759	-0.071330	-0.050117	...	
BsmtUnfSF	-0.007940	-0.140759	0.124098	-0.002618	0.308159	-0.136841	0.149040	0.181133	0.113862	-0.495251	...	
TotalBsmtSF	-0.015415	-0.238518	0.363472	0.260833	0.537808	-0.171098	0.391452	0.291066	0.360067	0.522396	...	
1stFlrSF	0.010496	-0.251758	0.413773	0.299475	0.476224	-0.144203	0.281986	0.240379	0.339850	0.445863	...	
2ndFlrSF	0.005590	0.307886	0.072388	0.050986	0.295493	0.028942	0.010308	0.140024	0.173800	-0.137079	...	
LowQualFinSF	-0.044230	0.046474	0.037469	0.004779	-0.030429	0.025494	-0.183784	-0.062419	-0.068628	-0.064503	...	
GrLivArea	0.008273	0.074853	0.368007	0.263116	0.593007	-0.079686	0.199010	0.287389	0.388052	0.208171	...	
BsmtFullBath	0.002289	0.003491	0.090343	0.158155	0.111098	-0.054942	0.187599	0.119470	0.083010	0.649212	...	
BsmtHalfBath	-0.020155	-0.002333	-0.006979	0.048046	-0.040150	0.117821	-0.038162	-0.012337	0.027403	0.067418	...	
FullBath	0.005587	0.131608	0.180534	0.126031	0.550600	-0.194149	0.468271	0.439046	0.272999	0.058543	...	
HalfBath	0.006784	0.177354	0.047222	0.014259	0.273458	-0.060769	0.242656	0.183331	0.199108	0.004262	...	
BedroomAbvGr	0.037719	-0.023438	0.236840	0.119690	0.101676	0.012980	-0.070651	-0.040581	0.102775	-0.107355	...	
KitchenAbvGr	0.002951	0.281721	-0.004905	-0.017784	-0.183882	-0.087001	-0.174800	-0.149598	-0.038450	-0.081007	...	
TotRmsAbvGrd	0.027239	0.040380	0.320518	0.190015	0.427452	-0.057583	0.095589	0.191740	0.279568	0.044316	...	
Fireplaces	-0.019772	-0.045569	0.233221	0.271364	0.396765	-0.023820	0.147716	0.112581	0.247015	0.260011	...	
GarageYrBlt	-0.000122	0.081396	0.062996	-0.025865	0.514231	-0.306276	0.777182	0.616444	0.244444	0.148782	...	
GarageCars	0.016570	-0.040110	0.269539	0.154871	0.600671	-0.185758	0.537850	0.420622	0.361945	0.224054	...	

In [42]:
```
data.corr()['SalePrice'].abs().sort_values(ascending=False)
```

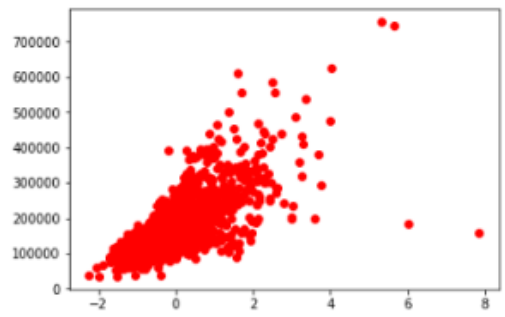
Out[42]:
```
SalePrice              1.000000
OverallQual            0.790982
GrLivArea              0.708624
GarageCars             0.640409
GarageArea             0.623431
TotalBsmtSF            0.613581
1stFlrSF               0.605852
ExterQual_TA           0.589044
FullBath               0.560664
BsmtQual_Ex            0.553105
TotRmsAbvGrd           0.533723
YearBuilt              0.522897
KitchenQual_TA         0.519298
GarageFinish_Unf       0.513906
YearRemodAdd           0.507101
KitchenQual_Ex         0.504094
BsmtQual_TA            0.498545
Foundation_PConc       0.497734
MasVnrArea             0.472614
Fireplaces             0.466929
GarageYrBlt            0.466754
ExterQual_Gd           0.452466
ExterQual_Ex           0.451164
BsmtFinType1_GLQ       0.434597
HeatingQC_Ex           0.434543
GarageFinish_Fin       0.419678
Neighborhood_NridgHt   0.402149
BsmtFinSF1             0.386420
MasVnrType_None        0.367456
SaleType_New           0.357509
                          ...
Functional_Sev         0.017116
BsmtHalfBath           0.016844
Exterior2nd_Stone      0.016754
RoofStyle_Flat         0.016433
PoolQC_Fa              0.016127
Functional_Mod         0.016073
SaleCondition_Alloca   0.015525
Neighborhood_SawyerW   0.014560
Condition2_RRAn        0.014510
```

In [43]:
```
data['GrLivArea'].corr(data['TotalBsmtSF'])
```

Out[43]:
```
0.45486820254790294
```

In [47]:
```
data = data[['OverallQual','GrLivArea','GarageCars','TotalBsmtSF','ExterQual_TA','FullBath','BsmtQual_Ex',
'TotRmsAbvGrd','YearBuilt','KitchenQual_TA','GarageFinish_Unf','KitchenQual_Ex','SalePrice']]
plt.plot(data['GrLivArea'], data['SalePrice'], 'ro')
```

Out[47]:
```
[<matplotlib.lines.Line2D at 0x25a68390cf8>]
```



In [49]:
```
# Разделить данные

x = data.drop(('SalePrice'), axis=1) # входные фичи

y = data['SalePrice'] # ответ

x_train, x_valid, y_train, y_valid = train_test_split(x, y, test_size = 0.25, random_state = 11)

#Обучить модель из sklearn, реализовать линейную регрессию

regr = linear_model.LinearRegression(fit_intercept=True)

regr.fit(x_train, y_train)

y_valid_predict = regr.predict(x_valid)

print('Коэффициенты: \n', regr.coef_)

print("MAE(средний модуль ошибки): %.2f" % mean_absolute_error(y_valid, y_valid_predict))

print('Оценка дисперсии: %.2f' % r2_score(y_valid, y_valid_predict))
```

```
Коэффициенты:
 [ 20040.67868727  21282.06721516  10568.03838394   8025.47815134
  -5591.26260915  -1630.22122407  33991.18852775   2996.11055473
   6258.80121185  -9984.24646655  -7607.93499391  29703.13546797]
MAE(средний модуль ошибки): 21247.26
Оценка дисперсии: 0.83
```

In [51]:
```
#прогон модели по тестовой выборке

x_test = pd.read_csv('./KaggleLab3/test.csv')

y_test = pd.read_csv('./KaggleLab3/sample_submission.csv')

x_test = cleaning(x_test)

x_test = x_test[['OverallQual','GrLivArea','GarageCars','TotalBsmtSF','ExterQual_TA','FullBath','BsmtQual_Ex','TotRmsAbvGrd','Yea

y_test = y_test[['SalePrice']]

#предсказание

y_test_predict = regr.predict(x_test)

print('Коэффициенты: \n', regr.coef_)

print("MAE: %.2f" % mean_absolute_error(y_test, y_test_predict))

print('Оценка дисперсии: %.2f' % r2_score(y_test, y_test_predict))
```

```
Коэффициенты:
 [ 20040.67868727  21282.06721516  10568.03838394   8025.47815134
  -5591.26260915  -1630.22122407  33991.18852775   2996.11055473
   6258.80121185  -9984.24646655  -7607.93499391  29703.13546797]
MAE: 52785.16
Оценка дисперсии: -15.53
```