# Exercise “Regression with a Multi Layer Perceptron (MLP)” Part 1/3

## Prof. Dr.-Ing. Jürgen Brauer

**Introduction:**

In this and the next exercises we will do something incredible! We will predict the sale price of houses based on some data items describing the house and take part at the Kaggle competitions. So you will become a “Kaggler!”.

Kaggle is a website offering a lot of datasets and competitions related to these datasets. A lot of machine learning enthusiasts take part at these competitions and having a high rating at one of these competitions can be a plus in your CV when applying for a data scientist position.

**Exercise 1: Getting the data and learning to work with Pandas**

Go to the website of the competition “House Prices: Advanced Regression Techniques“ at Kaggle:
https://www.kaggle.com/c/house-prices-advanced-regression-techniques

Then download the dataset at
https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data

`train.csv` contains 1461 rows of 81 columns. In each row we find the data for one house. The last column was the real sale price of the house.

`test.cvv` contains 1460 rows of only 80 columns. The ‘SalePrice’ column is missing. Our model that we will use will be a MLP and has to predict the sale price just on the other data columns.

We will submit our predicted sale prices later for `test.csv` at Kaggle and they will tell us how good we are regarding the predicted sale price compared to other “Kagglers”.

1.1
Read in the data using Pandas. You will get a Pandas data frame.

1.2
Retrieve the number of rows and columns for the data frame that you just have read in.

1.3
Print the names of all columns and each corresponding data type.

1.4
Get a complete single row from the data frame and show it.

1.5
Show for some years the values “YearBuilt” and “SalePrice”.

1.6
Use matplotlib to plot a scatter plot of the sale price and the year the house was built.
Is there a good correlation between the year the house was built and the sale price?

1.7
Compute the Pearson correlation coefficient for the column vectors “YearBuilt” and “SalePrice”.

1.8
Now prepare a new data frame that only contains the numeric columns of the original data frame.

1.9
Print all the names of the columns that contain numeric values.

1.10
Compute the Pearson correlation coefficient for each numeric column and the “SalePrice” column in order to detect columns (values) that are correlated to the “SalePrice”.

1.11
Show the names of all columns that have a Pearson correlation coefficient with the “SalePrice” column above 0.6.

1.12
Print some examples of values from the columns that are highly correlated with the “SalePrice” column and also print the corresponding sale price of the house.

## The output of your program should be similar to this:

```
1. Reading in training data...

2. Shape of data frame is (1460, 81)

3. Here are all column names:
column # 0 : column name = Id        data type is int64
column # 1 : column name = MSSubClass         data type is int64
column # 2 : column name = MSZoning  data type is object
column # 3 : column name = LotFrontage        data type is float64
column # 4 : column name = LotArea   data type is int64
column # 5 : column name = Street    data type is object
column # 6 : column name = Alley     data type is object
column # 7 : column name = LotShape  data type is object
column # 8 : column name = LandContour        data type is object
column # 9 : column name = Utilities data type is object
column # 10 : column name = LotConfig         data type is object
column # 11 : column name = LandSlope         data type is object
column # 12 : column name = Neighborhood      data type is object
column # 13 : column name = Condition1        data type is object
column # 14 : column name = Condition2        data type is object
column # 15 : column name = BldgType data type is object
column # 16 : column name = HouseStyle        data type is object
column # 17 : column name = OverallQual       data type is int64
column # 18 : column name = OverallCond       data type is int64
column # 19 : column name = YearBuilt         data type is int64
column # 20 : column name = YearRemodAdd      data type is int64
column # 21 : column name = RoofStyle         data type is object
column # 22 : column name = RoofMatl data type is object
column # 23 : column name = Exterior1st       data type is object
column # 24 : column name = Exterior2nd       data type is object
column # 25 : column name = MasVnrType        data type is object
column # 26 : column name = MasVnrArea        data type is float64
column # 27 : column name = ExterQual         data type is object
column # 28 : column name = ExterCond         data type is object
column # 29 : column name = Foundation        data type is object
column # 30 : column name = BsmtQual data type is object
column # 31 : column name = BsmtCond data type is object
column # 32 : column name = BsmtExposure      data type is object
column # 33 : column name = BsmtFinType1      data type is object
column # 34 : column name = BsmtFinSF1        data type is int64
column # 35 : column name = BsmtFinType2      data type is object
column # 36 : column name = BsmtFinSF2        data type is int64
column # 37 : column name = BsmtUnfSF         data type is int64
column # 38 : column name = TotalBsmtSF       data type is int64
column # 39 : column name = Heating  data type is object
column # 40 : column name = HeatingQC         data type is object
column # 41 : column name = CentralAir        data type is object
column # 42 : column name = Electrical        data type is object
column # 43 : column name = 1stFlrSF data type is int64
column # 44 : column name = 2ndFlrSF data type is int64
column # 45 : column name = LowQualFinSF      data type is int64
column # 46 : column name = GrLivArea         data type is int64
column # 47 : column name = BsmtFullBath      data type is int64
column # 48 : column name = BsmtHalfBath      data type is int64
column # 49 : column name = FullBath data type is int64
column # 50 : column name = HalfBath data type is int64
column # 51 : column name = BedroomAbvGr      data type is int64
column # 52 : column name = KitchenAbvGr      data type is int64
column # 53 : column name = KitchenQual       data type is object
column # 54 : column name = TotRmsAbvGrd      data type is int64
column # 55 : column name = Functional        data type is object
column # 56 : column name = Fireplaces        data type is int64
column # 57 : column name = FireplaceQu       data type is object
column # 58 : column name = GarageType        data type is object
column # 59 : column name = GarageYrBlt       data type is float64
```

```
column # 60 : column name = GarageFinish      data type is object
column # 61 : column name = GarageCars        data type is int64
column # 62 : column name = GarageArea        data type is int64
column # 63 : column name = GarageQual        data type is object
column # 64 : column name = GarageCond        data type is object
column # 65 : column name = PavedDrive        data type is object
column # 66 : column name = WoodDeckSF        data type is int64
column # 67 : column name = OpenPorchSF       data type is int64
column # 68 : column name = EnclosedPorch     data type is int64
column # 69 : column name = 3SsnPorch         data type is int64
column # 70 : column name = ScreenPorch       data type is int64
column # 71 : column name = PoolArea data type is int64
column # 72 : column name = PoolQC   data type is object
column # 73 : column name = Fence    data type is object
column # 74 : column name = MiscFeature       data type is object
column # 75 : column name = MiscVal  data type is int64
column # 76 : column name = MoSold   data type is int64
column # 77 : column name = YrSold   data type is int64
column # 78 : column name = SaleType data type is object
column # 79 : column name = SaleCondition     data type is object
column # 80 : column name = SalePrice         data type is int64
```

**4. Here is the first row of the table:**

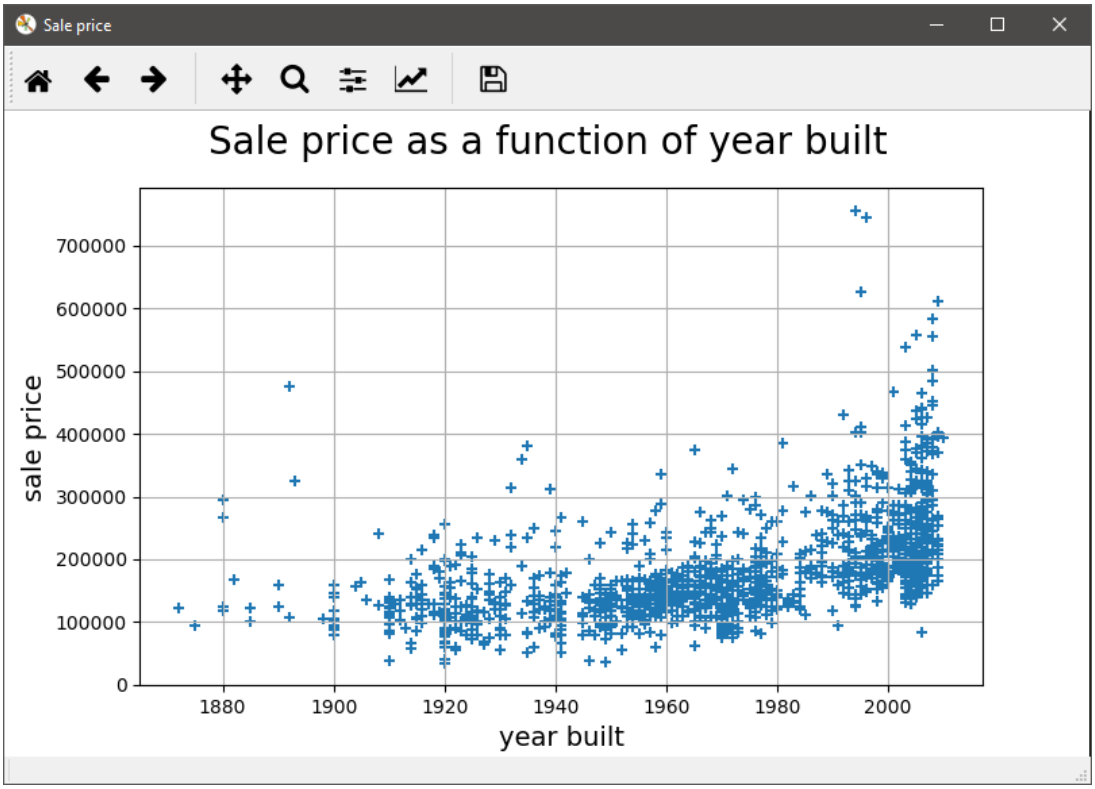
```
[1 60 'RL' 65.0 8450 'Pave' nan 'Reg' 'Lvl' 'AllPub' 'Inside' 'Gtl'
 'CollgCr' 'Norm' 'Norm' '1Fam' '2Story' 7 5 2003 2003 'Gable' 'CompShg'
 'VinylSd' 'VinylSd' 'BrkFace' 196.0 'Gd' 'TA' 'PConc' 'Gd' 'TA' 'No'
 'GLQ' 706 'Unf' 0 150 856 'GasA' 'Ex' 'Y' 'SBrkr' 856 854 0 1710 1 0 2 1
 3 1 'Gd' 8 'Typ' 0 nan 'Attchd' 2003.0 'RFn' 2 548 'TA' 'TA' 'Y' 0 61 0 0
 0 0 nan nan nan 0 2 2008 'WD' 'Normal' 208500]
```

**5. Now we show for the first 10 rows the year and the sales price:**

```
row nr # 0  : YearBuilt:  2003 --> SalePrice: 208500
row nr # 1  : YearBuilt:  1976 --> SalePrice: 181500
row nr # 2  : YearBuilt:  2001 --> SalePrice: 223500
row nr # 3  : YearBuilt:  1915 --> SalePrice: 140000
row nr # 4  : YearBuilt:  2000 --> SalePrice: 250000
row nr # 5  : YearBuilt:  1993 --> SalePrice: 143000
row nr # 6  : YearBuilt:  2004 --> SalePrice: 307000
row nr # 7  : YearBuilt:  1973 --> SalePrice: 200000
row nr # 8  : YearBuilt:  1931 --> SalePrice: 129900
row nr # 9  : YearBuilt:  1939 --> SalePrice: 118000
```

**6. Now we plot the sale price as afunction of the year built**

```
C:\Users\Juergen Brauer\AppData\Local\conda\conda\envs\env_mss\lib\site-
packages\matplotlib\backend_bases.py:2453: MatplotlibDeprecationWarning: Using default
event loop until function specific to this GUI is implemented  warnings.warn(str,
mplDeprecation)
```




**7. Pearson correlation coefficient of (year_built, sale price) vectors is : 0.5228973328794969**

**8. Preparing a new data frame with only numeric columns**

**9. Here are only the columns which contain numeric data**

```
There are  38 numeric columns
numeric column # 0 : column name = Id
numeric column # 1 : column name = MSSubClass
numeric column # 2 : column name = LotFrontage
numeric column # 3 : column name = LotArea
numeric column # 4 : column name = OverallQual
numeric column # 5 : column name = OverallCond
numeric column # 6 : column name = YearBuilt
numeric column # 7 : column name = YearRemodAdd
numeric column # 8 : column name = MasVnrArea
numeric column # 9 : column name = BsmtFinSF1
numeric column # 10 : column name = BsmtFinSF2
numeric column # 11 : column name = BsmtUnfSF
numeric column # 12 : column name = TotalBsmtSF
numeric column # 13 : column name = 1stFlrSF
numeric column # 14 : column name = 2ndFlrSF
numeric column # 15 : column name = LowQualFinSF
numeric column # 16 : column name = GrLivArea
numeric column # 17 : column name = BsmtFullBath
numeric column # 18 : column name = BsmtHalfBath
numeric column # 19 : column name = FullBath
numeric column # 20 : column name = HalfBath
numeric column # 21 : column name = BedroomAbvGr
numeric column # 22 : column name = KitchenAbvGr
numeric column # 23 : column name = TotRmsAbvGrd
numeric column # 24 : column name = Fireplaces
numeric column # 25 : column name = GarageYrBlt
numeric column # 26 : column name = GarageCars
numeric column # 27 : column name = GarageArea
numeric column # 28 : column name = WoodDeckSF
numeric column # 29 : column name = OpenPorchSF
numeric column # 30 : column name = EnclosedPorch
```

```
numeric column # 31 : column name = 3SsnPorch
numeric column # 32 : column name = ScreenPorch
numeric column # 33 : column name = PoolArea
numeric column # 34 : column name = MiscVal
numeric column # 35 : column name = MoSold
numeric column # 36 : column name = YrSold
numeric column # 37 : column name = SalePrice
```

**10. Here is the correlation of each numeric column with the 'SalesPrice' column:**

```
Pearson correlation of  Id and 'SalePrice' is -0.021916719443431106
Pearson correlation of  MSSubClass and 'SalePrice' is -0.08428413512659517
Pearson correlation of  LotFrontage and 'SalePrice' is nan
Pearson correlation of  LotArea and 'SalePrice' is 0.2638433538714056
Pearson correlation of  OverallQual and 'SalePrice' is 0.7909816005838051
Pearson correlation of  OverallCond and 'SalePrice' is -0.077855894048678
Pearson correlation of  YearBuilt and 'SalePrice' is 0.5228973328794969
Pearson correlation of  YearRemodAdd and 'SalePrice' is 0.5071009671113862
C:\Users\Juergen Brauer\AppData\Local\conda\conda\envs\env_mss\lib\site-
packages\scipy\stats\stats.py:5277: RuntimeWarning: invalid value encountered in less
  x = np.where(x < 1.0, x, 1.0)  # if x > 1 then return 1.0
Pearson correlation of  MasVnrArea and 'SalePrice' is nan
Pearson correlation of  BsmtFinSF1 and 'SalePrice' is 0.3864198062421533
Pearson correlation of  BsmtFinSF2 and 'SalePrice' is -0.01137812145021514
Pearson correlation of  BsmtUnfSF and 'SalePrice' is 0.2144791055469689
Pearson correlation of  TotalBsmtSF and 'SalePrice' is 0.6135805515591953
Pearson correlation of  1stFlrSF and 'SalePrice' is 0.6058521846919146
Pearson correlation of  2ndFlrSF and 'SalePrice' is 0.3193338028320678
Pearson correlation of  LowQualFinSF and 'SalePrice' is -0.025606130000679534
Pearson correlation of  GrLivArea and 'SalePrice' is 0.708624477612652
Pearson correlation of  BsmtFullBath and 'SalePrice' is 0.22712223313149424
Pearson correlation of  BsmtHalfBath and 'SalePrice' is -0.016844154297359016
Pearson correlation of  FullBath and 'SalePrice' is 0.560663762748446
Pearson correlation of  HalfBath and 'SalePrice' is 0.2841076755947825
Pearson correlation of  BedroomAbvGr and 'SalePrice' is 0.16821315430073996
Pearson correlation of  KitchenAbvGr and 'SalePrice' is -0.13590737084214122
Pearson correlation of  TotRmsAbvGrd and 'SalePrice' is 0.5337231555820281
Pearson correlation of  Fireplaces and 'SalePrice' is 0.4669288367515278
Pearson correlation of  GarageYrBlt and 'SalePrice' is nan
Pearson correlation of  GarageCars and 'SalePrice' is 0.640409197258352
Pearson correlation of  GarageArea and 'SalePrice' is 0.6234314389183616
Pearson correlation of  WoodDeckSF and 'SalePrice' is 0.3244134445681299
Pearson correlation of  OpenPorchSF and 'SalePrice' is 0.3158562271160552
Pearson correlation of  EnclosedPorch and 'SalePrice' is -0.12857795792595675
Pearson correlation of  3SsnPorch and 'SalePrice' is 0.0445836653357484
Pearson correlation of  ScreenPorch and 'SalePrice' is 0.11144657114291123
Pearson correlation of  PoolArea and 'SalePrice' is 0.09240354949187321
Pearson correlation of  MiscVal and 'SalePrice' is -0.021189579640303255
Pearson correlation of  MoSold and 'SalePrice' is 0.04643224522381935
Pearson correlation of  YrSold and 'SalePrice' is -0.028922585168730326
Pearson correlation of  SalePrice and 'SalePrice' is 1.0
```

**11. List of columns highly correlated with the sale price:**

```
Here highly correlated means, that the Pearsoncorrelation coefficient is above 0.6
['OverallQual', 'TotalBsmtSF', '1stFlrSF', 'GrLivArea', 'GarageCars', 'GarageArea']
```

**12. Here are some examples of the values in the columns that are highly correlated with the sale price:**

```
( OverallQual , 7 ) ( TotalBsmtSF , 856 ) ( 1stFlrSF , 856 ) ( GrLivArea , 1710 ) (
GarageCars , 2 ) ( GarageArea , 548 ) --> salesprice was 208500
( OverallQual , 6 ) ( TotalBsmtSF , 1262 ) ( 1stFlrSF , 1262 ) ( GrLivArea , 1262 ) (
GarageCars , 2 ) ( GarageArea , 460 ) --> salesprice was 181500
( OverallQual , 7 ) ( TotalBsmtSF , 920 ) ( 1stFlrSF , 920 ) ( GrLivArea , 1786 ) (
GarageCars , 2 ) ( GarageArea , 608 ) --> salesprice was 223500
( OverallQual , 7 ) ( TotalBsmtSF , 756 ) ( 1stFlrSF , 961 ) ( GrLivArea , 1717 ) (
GarageCars , 3 ) ( GarageArea , 642 ) --> salesprice was 140000
( OverallQual , 8 ) ( TotalBsmtSF , 1145 ) ( 1stFlrSF , 1145 ) ( GrLivArea , 2198 ) (
GarageCars , 3 ) ( GarageArea , 836 ) --> salesprice was 250000
```

( OverallQual , 5 ) ( TotalBsmtSF , 796 ) ( 1stFlrSF , 796 ) ( GrLivArea , 1362 ) ( GarageCars , 2 ) ( GarageArea , 480 ) --> salesprice was 143000
( OverallQual , 8 ) ( TotalBsmtSF , 1686 ) ( 1stFlrSF , 1694 ) ( GrLivArea , 1694 ) ( GarageCars , 2 ) ( GarageArea , 636 ) --> salesprice was 307000
( OverallQual , 7 ) ( TotalBsmtSF , 1107 ) ( 1stFlrSF , 1107 ) ( GrLivArea , 2090 ) ( GarageCars , 2 ) ( GarageArea , 484 ) --> salesprice was 200000
( OverallQual , 7 ) ( TotalBsmtSF , 952 ) ( 1stFlrSF , 1022 ) ( GrLivArea , 1774 ) ( GarageCars , 2 ) ( GarageArea , 468 ) --> salesprice was 129900
( OverallQual , 5 ) ( TotalBsmtSF , 991 ) ( 1stFlrSF , 1077 ) ( GrLivArea , 1077 ) ( GarageCars , 1 ) ( GarageArea , 205 ) --> salesprice was 118000