KC House Sales Dataset

Prediction Sales Price

By Luigi Fiori

We need to answer 3 questions:

1. Which one is the variable that has a bigger impact on house pricing?

2. How positive is the presence of an additional bathroom in terms of potential overall price?

3. Is it the month a variable to take in consideration when selling a house?

Steps followed to answer the questions:

1. Data Mining: Get and Load the Data

2. Data Cleaning: Fix data inconsistencies and handle missing values

3. Data Visualization: Create data visualizations to understand our data and make necessary hypotheses

4. Predictive Modeling: train Models, evaluate their performance, and use them to create predictions

DATA MINING

We start by loading our dataset that can be found in the file "kc_house_data.csv", in the provided repository.

```
df = pd.read_csv('kc_house_data.csv')
df.head()
```

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade
0	7129300520	10/13/2014	221900.0	3	1.00	1180	5650	1.0	NaN	0.0	...	7
1	6414100192	12/9/2014	538000.0	3	2.25	2570	7242	2.0	0.0	0.0	...	7
2	5631500400	2/25/2015	180000.0	2	1.00	770	10000	1.0	0.0	0.0	...	6

We have a first look at the dataset and we already see that some of the data are missing.

In the next slide we will show how we dealt with it.

DATA CLEANING

The database presents some missing values and outliers:
We check for the total number of missing values using the built Python's function and we then deal with them in different ways based on the different circumstances:

```
df.isnull().sum()
```

id	0
date	0
price	0
bedrooms	0
bathrooms	0
sqft_living	0
sqft_lot	0
floors	0
waterfront	2376
view	63
condition	0
grade	0
sqft_above	0
sqft_basement	0
yr_built	0
yr_renovated	3842
zipcode	0
lat	0
long	0
sqft_living15	0
sqft_lot15	0
check_column	0

dtype: int64

I decide to fill in the missing values for 'waterfront': over 99% of data present value =0
Here the python code fills up in the 'waterfront' column the missing values with 0.0.

```
df['waterfront'] = df['waterfront'].fillna(value=0.0)
```

I drop the rows with missing values for 'view': less than 1% of data are lost.
The code drops the rows just for the 'view' column, inplace means definitively.

```
df.dropna(subset = ['view'], inplace = True)
```

I drop the all column for 'yr_renovated': 20% of missing values: not enough info about it.
The code below drops the all column 'yr_renov', axis refers the columns in the dataframe.

```
df = df.drop('yr_renovated', axis = 1)
```

After a deeper analysis 'sqft_basement' result to have idden missing values: we fill the missing data inserting the difference between 'sqft_living' and 'sqft_above'
Value count normalized counts the number of the values in %.

```
df['sqft_basement'].value_counts(normalize = True)
```

```
0.0        0.593906
?          0.021022
```

```
(df['sqft_living']) - (df['sqft_above'])
```

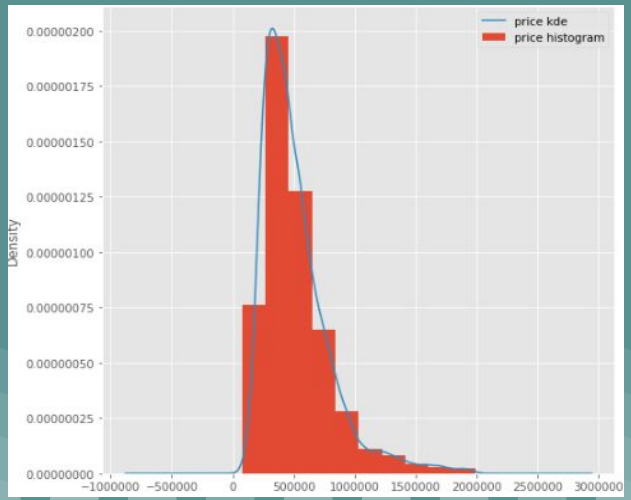
Data Visualization

Before fitting our model data visualization is a useful tool to have an idea of the distributions:

Histograms with KDE overlay to check the distribution

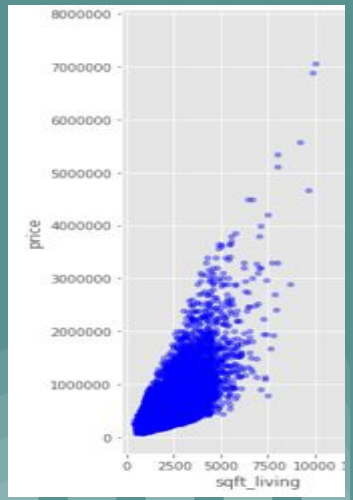
The distribution results normal or tend to be skewed?

Below we can see that the distribution is a bit skewed, so some outliers are present on the right side.



Scatter Plot

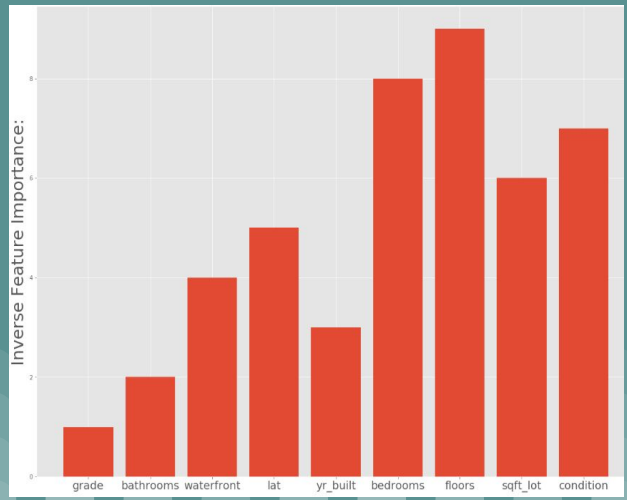
As we can see below there is linear relationship between the dependent variable(y) and the predictor(x).



Bar Plot for Features

Which is the most important feature?

The most important feature is grade followed by bathrooms and so on.



Predictive Model

OLS Regression Results

Dep. Variable:	price	R-squared:	0.583
Model:	OLS	Adj. R-squared:	0.583
Method:	Least Squares	F-statistic:	4937.
Date:	Sat, 06 Jul 2019	Prob (F-statistic):	0.00
Time:	11:47:08	Log-Likelihood:	-2.8616e+05
No. Observations:	21171	AIC:	5.723e+05
Df Residuals:	21164	BIC:	5.724e+05
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	5.578e+06	1.09e+05	51.335	0.000	5.37e+06	5.79e+06
grade	1.706e+05	1470.474	116.006	0.000	1.68e+05	1.73e+05
bathrooms	8.56e+04	2460.583	34.787	0.000	8.08e+04	9.04e+04
waterfront	4.249e+05	1.82e+04	23.325	0.000	3.89e+05	4.61e+05
yr_built	-3353.2809	55.687	-60.217	0.000	-3462.432	-3244.130
sqft_basement	51.9801	3.222	16.131	0.000	45.664	58.296
condition	1.499e+04	2054.312	7.297	0.000	1.1e+04	1.9e+04

Omnibus:	5426.476	Durbin-Watson:	1.947
Prob(Omnibus):	0.000	Jarque-Bera (JB):	22605.974
Skew:	1.211	Prob(JB):	0.00
Kurtosis:	7.445	Cond. No.	1.75e+05

The final model obtained present has an Adj. R-squared of 0.583 determining a decent goodness of fit of our model with the data, doing a good job of explaining changes in the dependent variable.

The 3 variables with an higher impact, as seen on the previous graph, on the overall house price are:

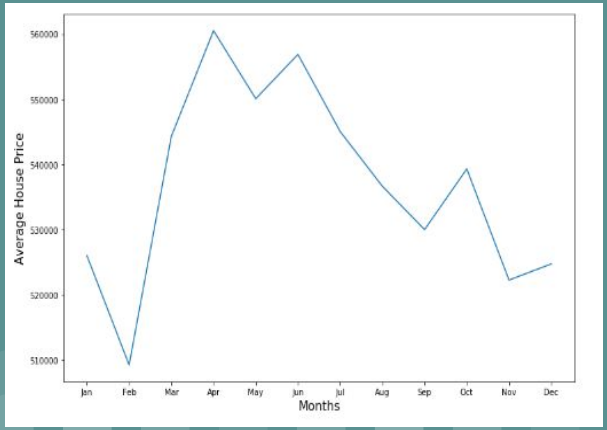
1. Grade-> an increase of 1 unit would increase the price of $170600
2. Bathrooms->an additional bathroom increase the price of $85600
3. Year Built->on average a 1 year older house it's worthed $3353 less

So in terms of house prices, when possible, are definitely these the variables to maximise.
The formula obtained is:
Price = 1.706e+05(grade) + 8.56e+04(bathrooms) + 4.249e+05(waterfront) - 3353(yr_built) + 52(sqft_basement) + 1.5e+04(condition) + 5.58e+06

P-values for these predictors are all statistically significant, if we check the corresponding values are all lower than 0.05.
This represents the probability that the coefficients are 0.

3 questions answered!

1. The variable that has a bigger impact on house pricing as shown from the graph above is grade.
2. The presence of an additional bathroom instead of a bedroom in terms of potential overall price is really positive. As we saw bathrooms is the second most important feature while bedrooms is the less important.
3. For our analysis let's have a look at the summary model:



As we can see from the model the month has an high influence on price depending on the month. February is the worst month while April is the best. So the answer at the 3rd question is definitely yes.

Future Work

There are some aspects of dataset that I would have like to explore further with more time available.

Firstly during the data cleaning process I filled up the missing values in the ‘waterfront’ column with 0 considering that more than the 99% of the data presented that value. Was definitely present a correlation between postcode and waterfall and with more time we could get more accurate results.

Secondly the ‘postcode’ column resulted to be really interesting in terms of house prices. Some postcodes are definitely more ‘premium’ than others an a deeper analysis could get valuable informations.

Lastly for the model, having more time, we could have tried more different combinations of predictors.

Said so we can feel satisfied with the result obtained.

Thanks everyone for your time!