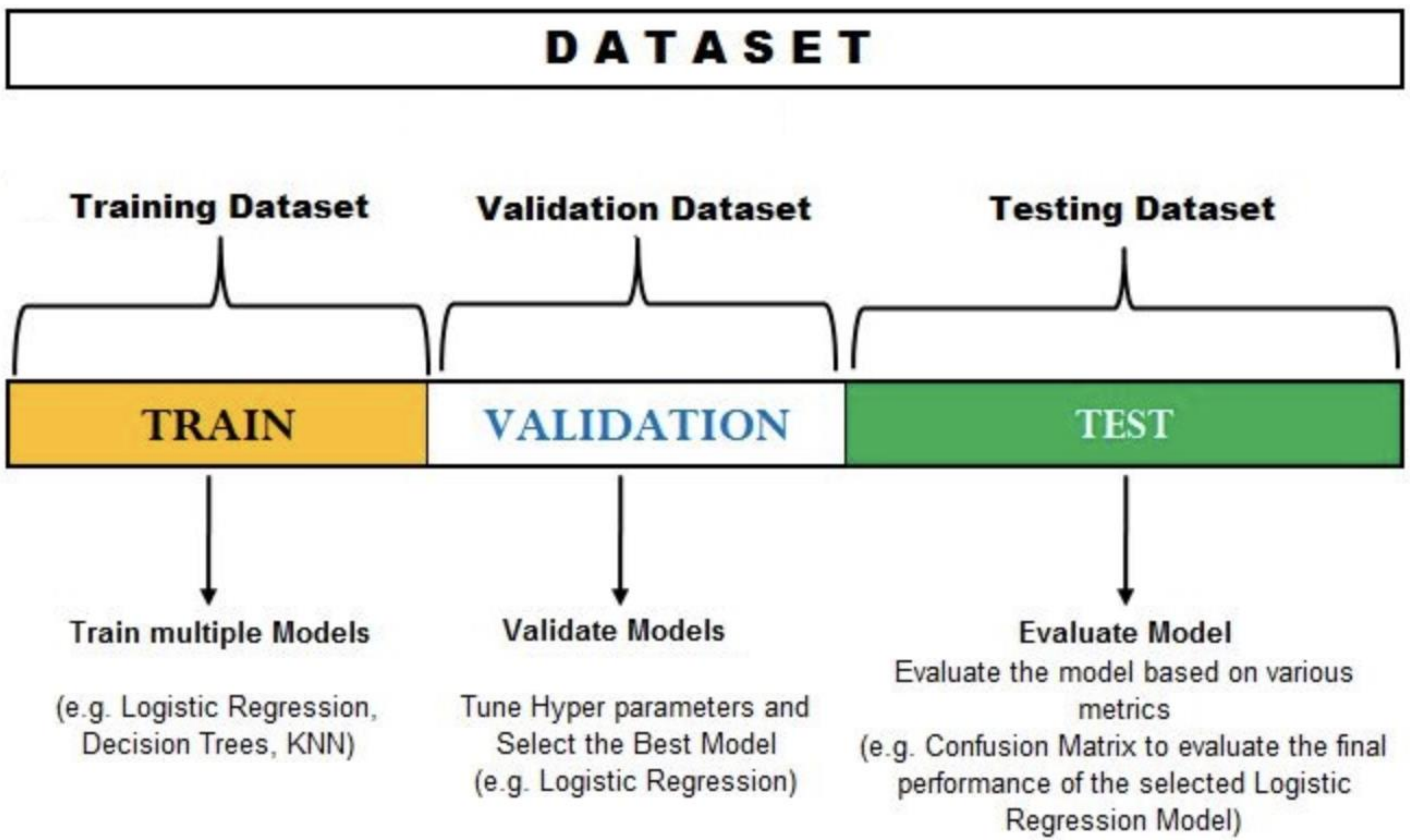
House Prices - Advanced Regression Techniques

Sanjin Jurić Fot

Josipa Radnić

Božidar Grgur Drmić


DATASET
Training Dataset
Validation Dataset
Testing Dataset
TRAIN
VALIDATION
TEST
Train multiple Models
(e.g. Logistic Regression, Decision Trees, KNN)
Validate Models
Tune Hyper parameters and Select the Best Model
(e.g. Logistic Regression)
Evaluate Model
Evaluate the model based on various metrics
(e.g. Confusion Matrix to evaluate the final performance of the selected Logistic Regression Model)

kaggle

PODACI	METRIKA
data_description.txt sample_submission.csv test.csv train.csv	$$RMSE_{\log} = \sqrt{\frac{\sum_{k=1}^{n} (\log \hat{y}_k - \log y_k)^2}{n}}$$ y_k = stvarna cijena kuće $\hat{y}_k$ = cijena koju je predvidio model **Logaritam!**

I. Opisna statistika

Ključni file nam je Train.csv

- 1480 redaka (kuća), 81 stupac
- 79 kovarijata, od čega 38 numeričkih
- Podjela: vrijeme, mjesto, kvaliteta i ostalo
- 5.8% podataka nedostaje

In [73]:
```
train.head(10)
```

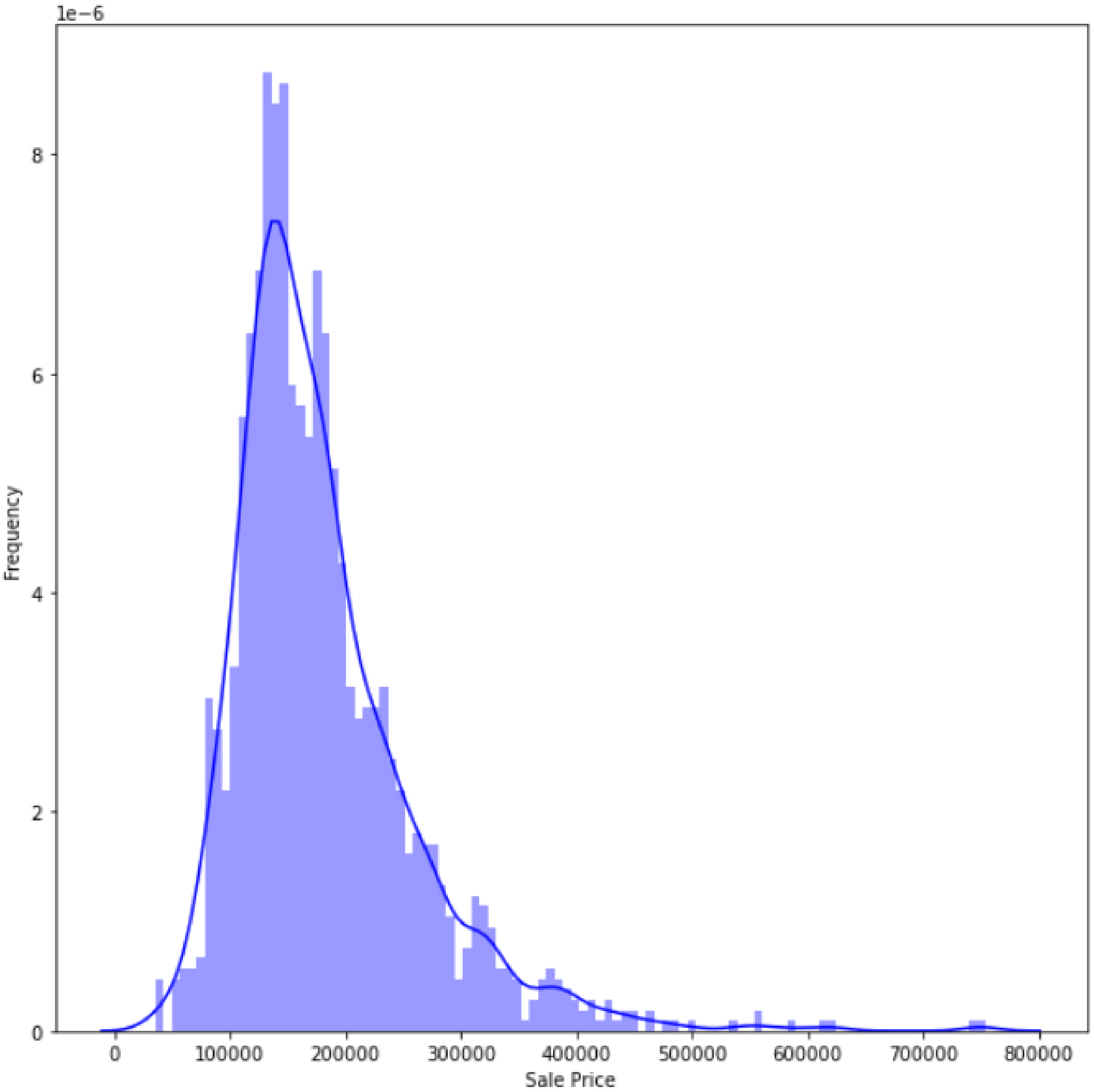
Out[73]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	Mo
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
5	6	50	RL	85.0	14115	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	MnPrv	Shed	700	
6	7	20	RL	75.0	10084	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
7	8	60	RL	NaN	10382	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	Shed	350	
8	9	50	RM	51.0	6120	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
9	10	190	RL	50.0	7420	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	

10 rows × 81 columns

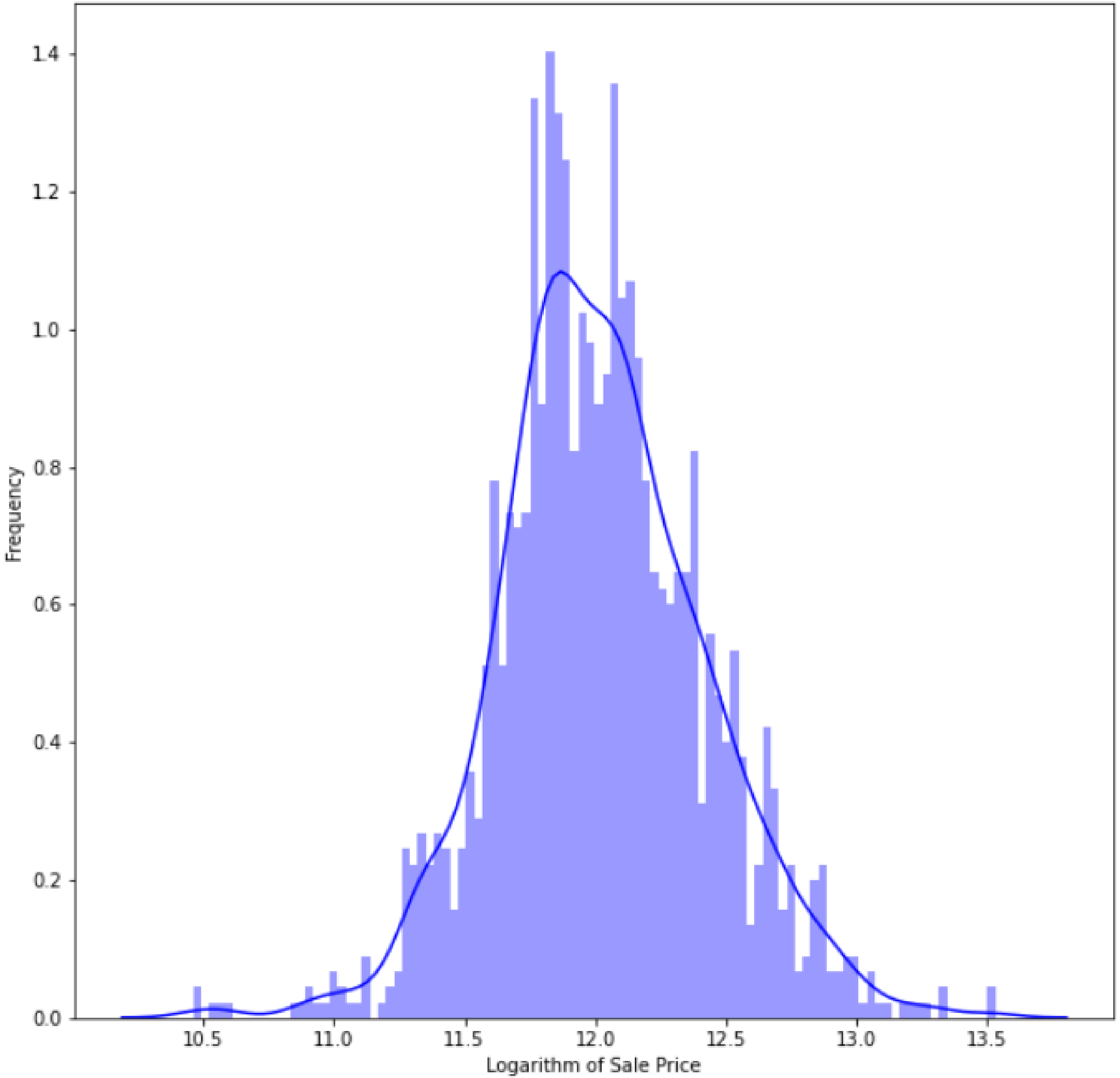
SalePrice je nagnuta ulijevo

```
In [85]: price.describe()

Out[85]: count      1460.000000
         mean     180921.195890
         std       79442.502883
         min       34900.000000
         25%      129975.000000
         50%      163000.000000
         75%      214000.000000
         max      755000.000000
```

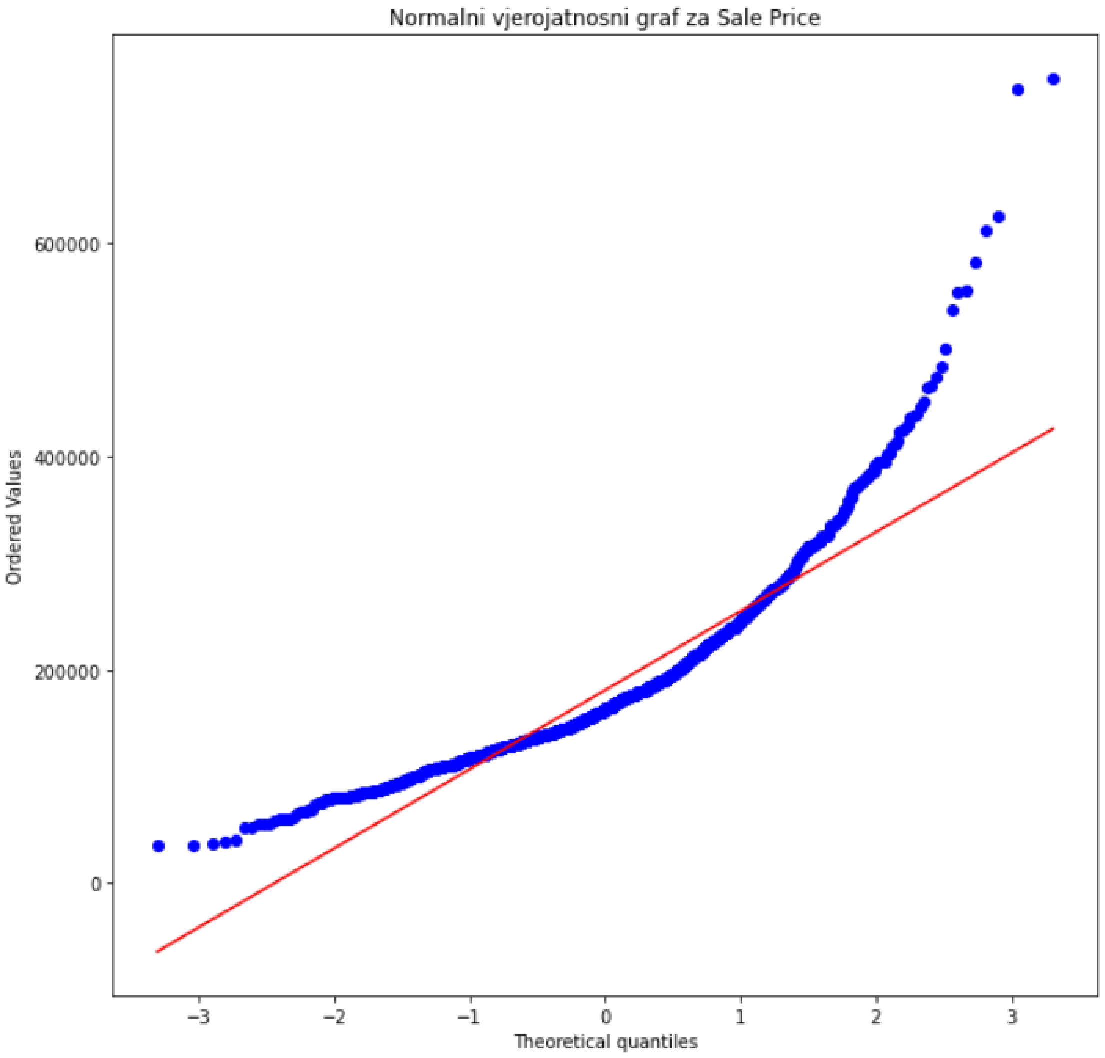


Njezin logaritam je više simetričan

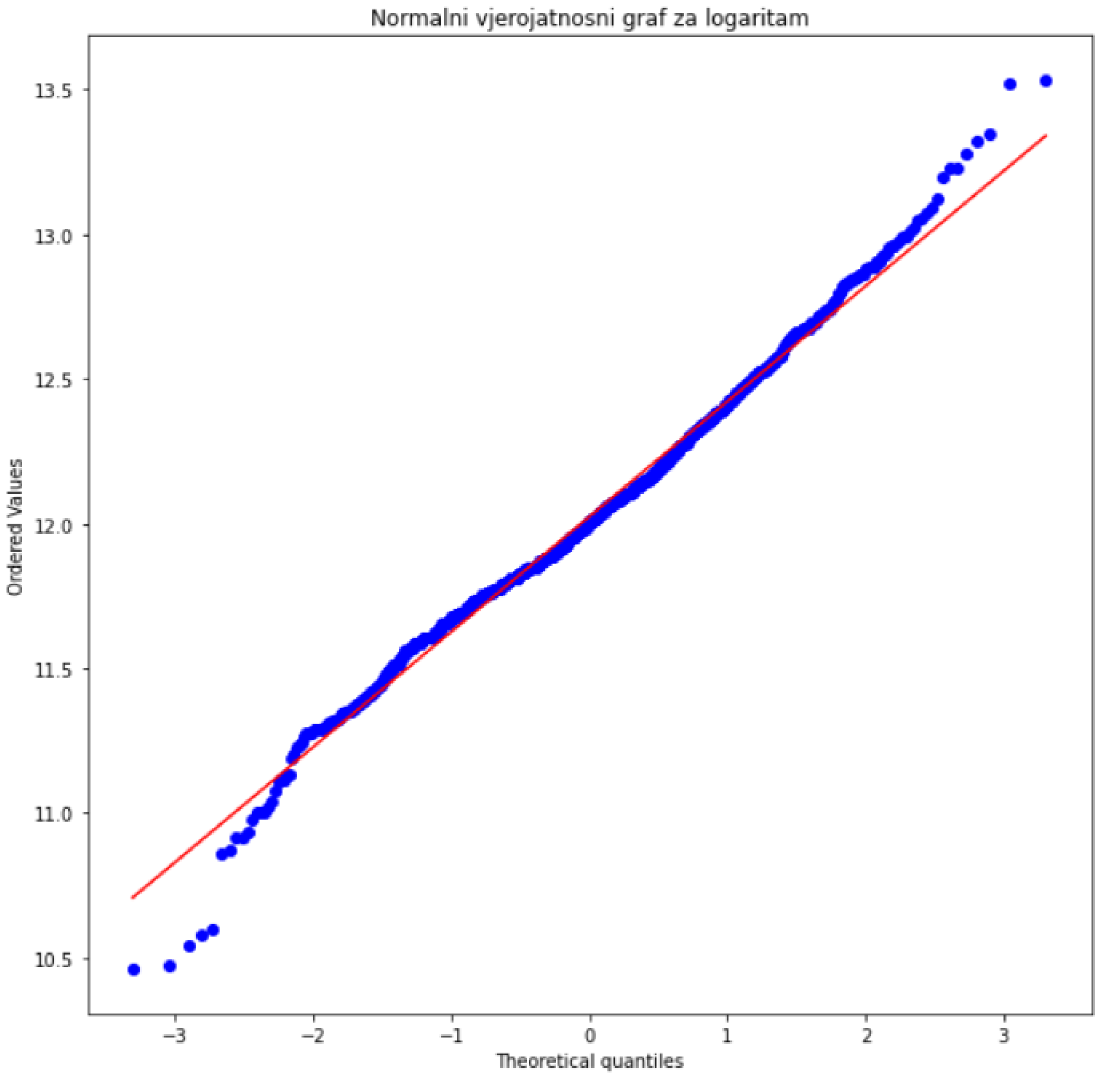
```
In [89]: log_price.describe()

Out[89]: count    1460.000000
         mean       12.024051
         std         0.399452
         min        10.460242
         25%        11.775097
         50%        12.001505
         75%        12.273731
         max        13.534473
```



Međutim, nitko ovdje nije normalan!



p=6.341849795213477e-61



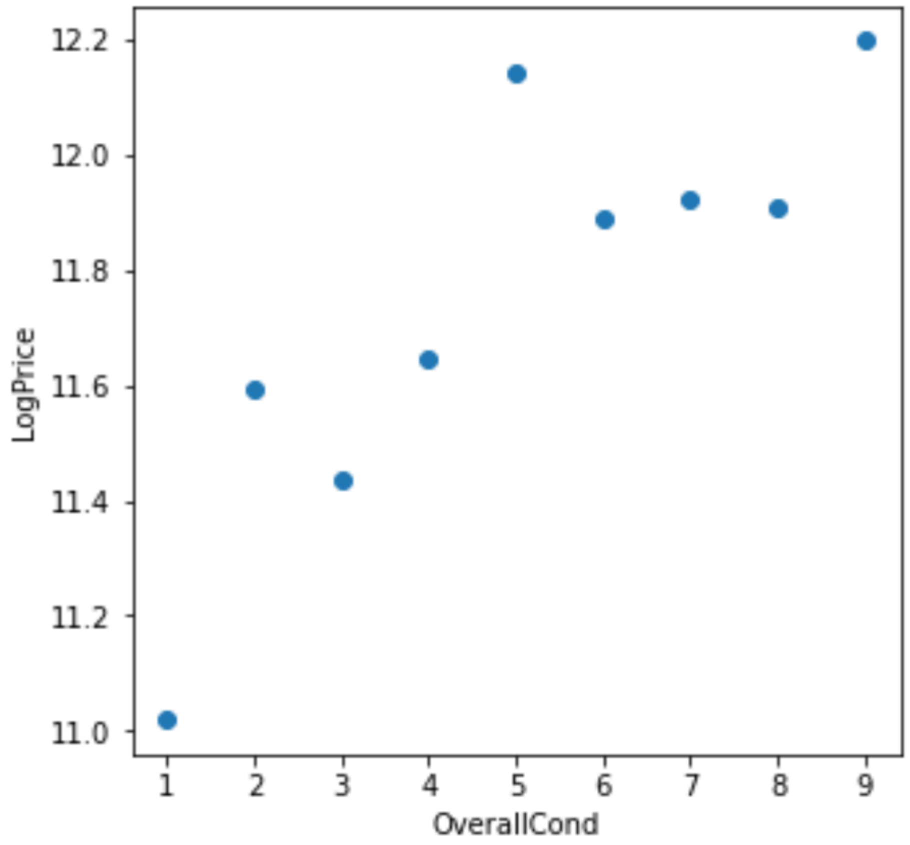
p=5.683759591984467e-06

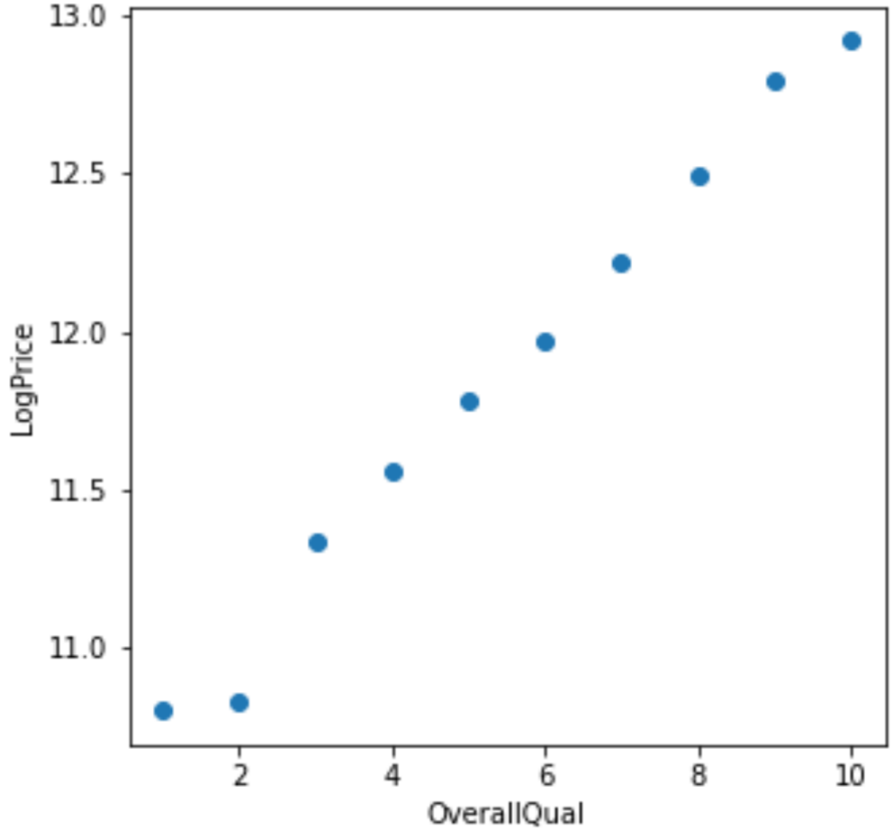
Podijelimo kovarijate:

- Vrijeme: *YearBuilt, YearRemodAdd, YrSold, MoSold*
- Mjesto: *MSZoning, Neighborhood, Condition*
- Kvaliteta: *„sve živo”, OverallQual, OverallCond*
- Ostalo: *SaleType, SaleCondition*

Sve živo:

Analizirajmo subjektivne procjene kuća

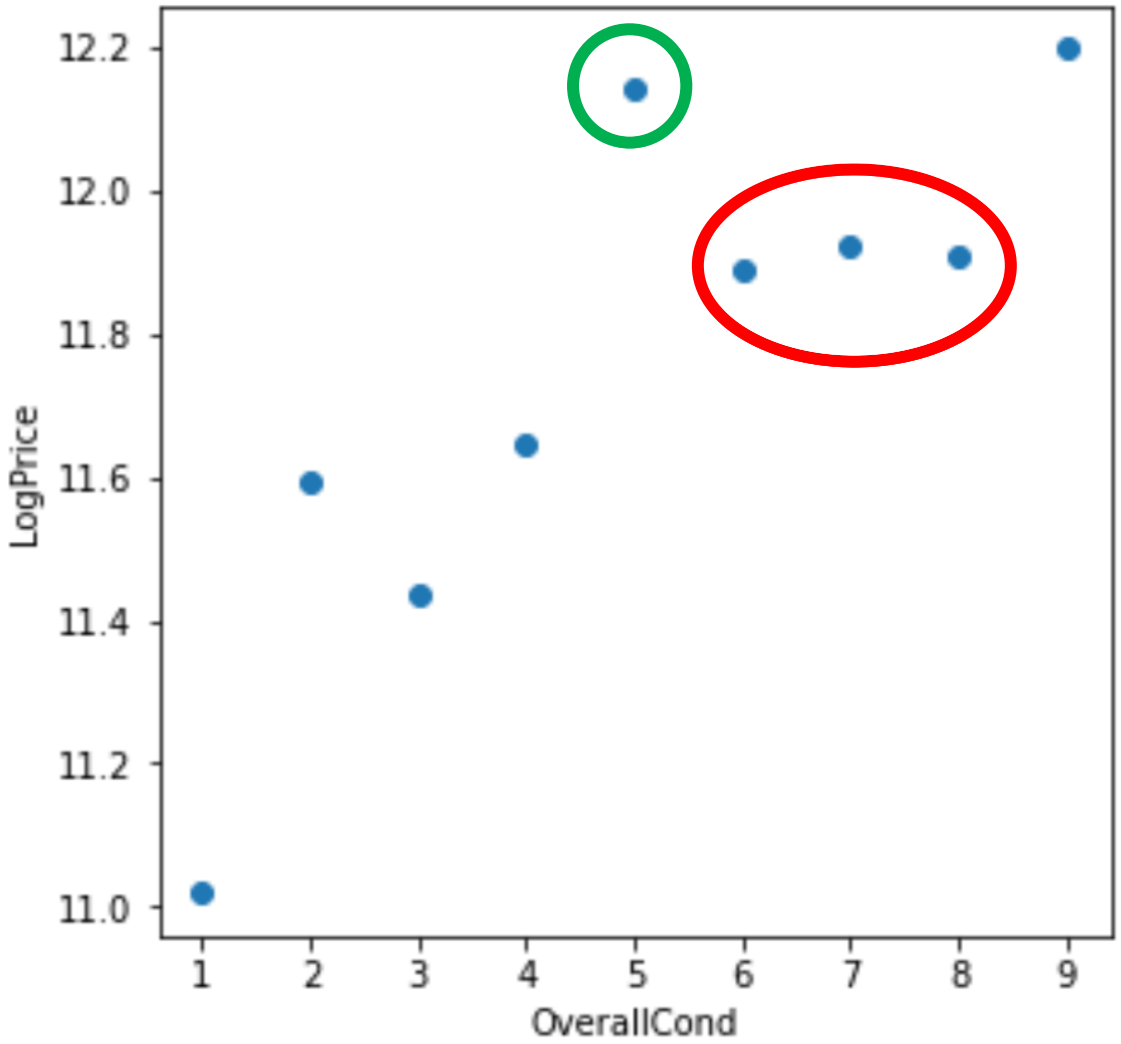


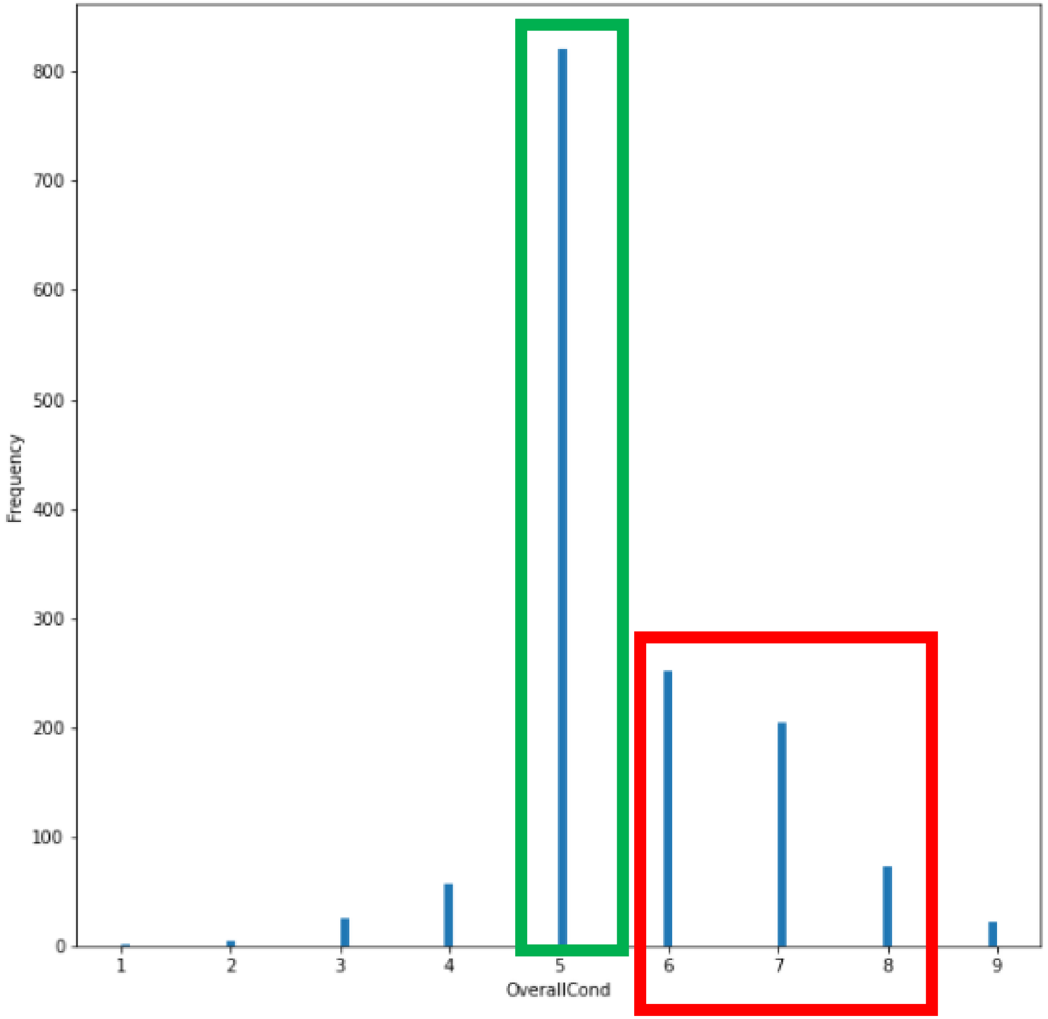


Corr(*OverallCond, OverallQual*) = -0.09 < 0

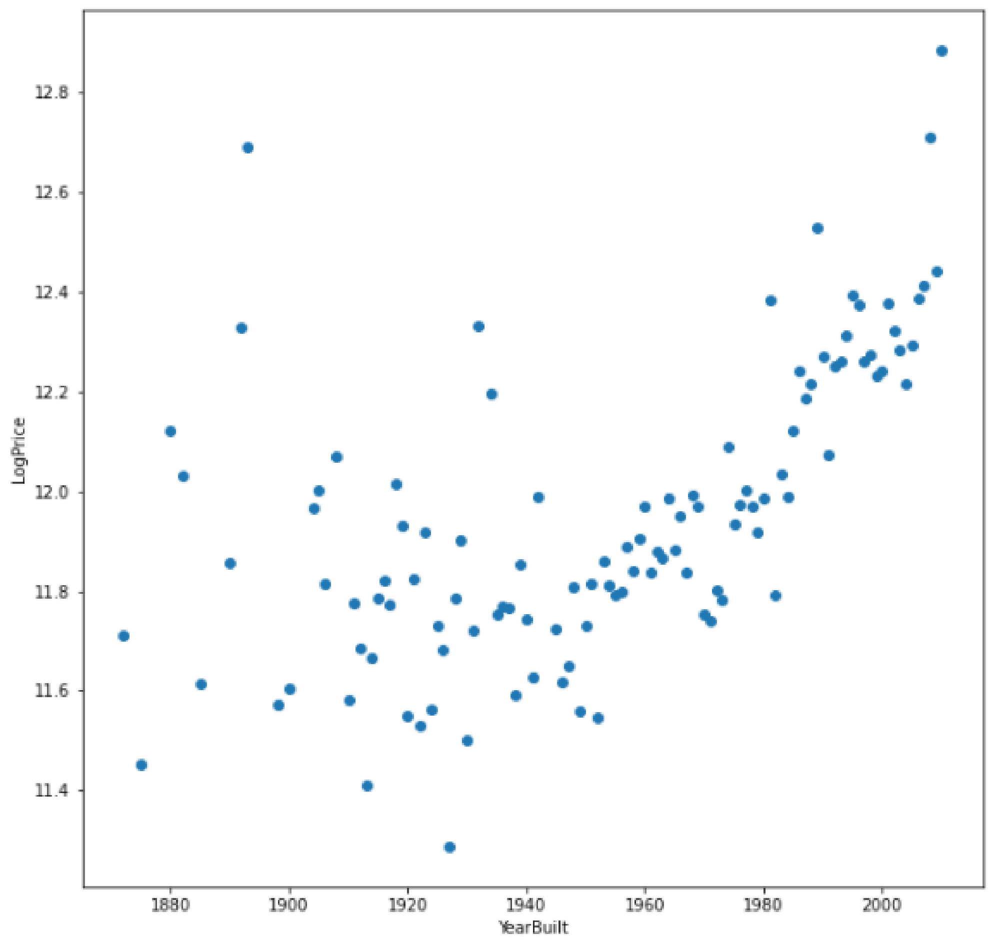
Corr(*OverAllCond, LogPrice*) = -0.03 < 0

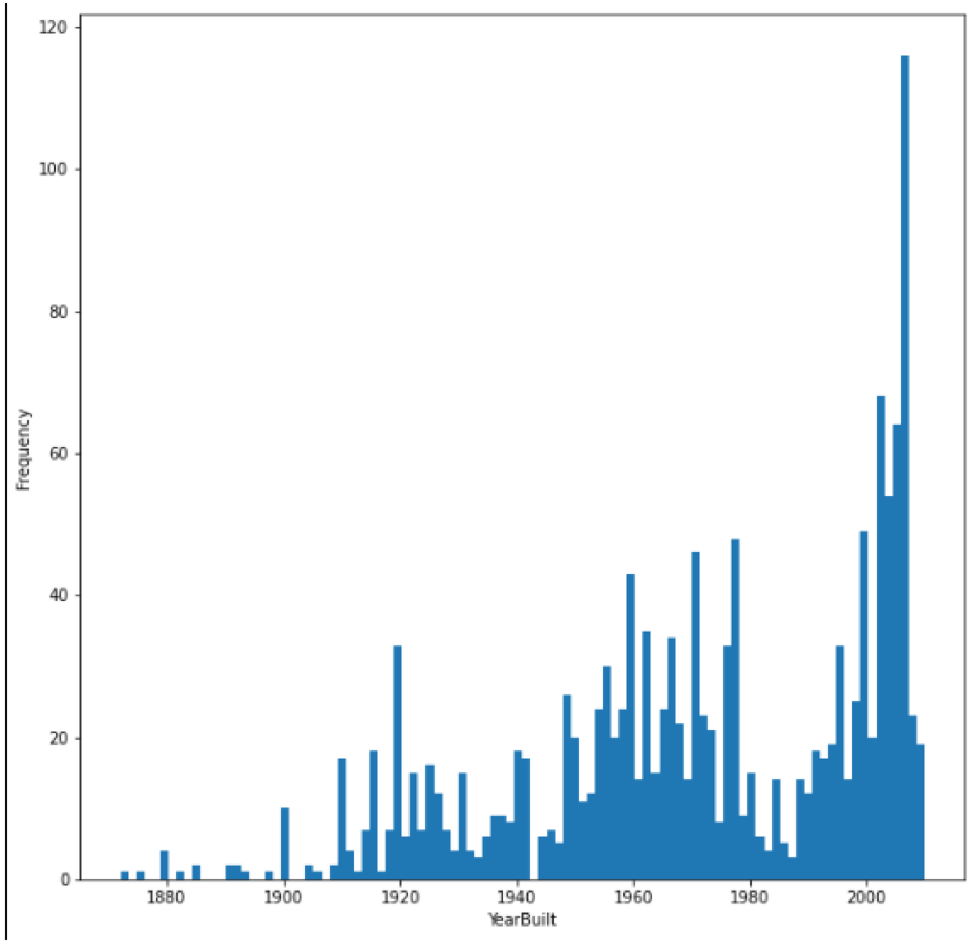
Koje je objašnjenje za to?



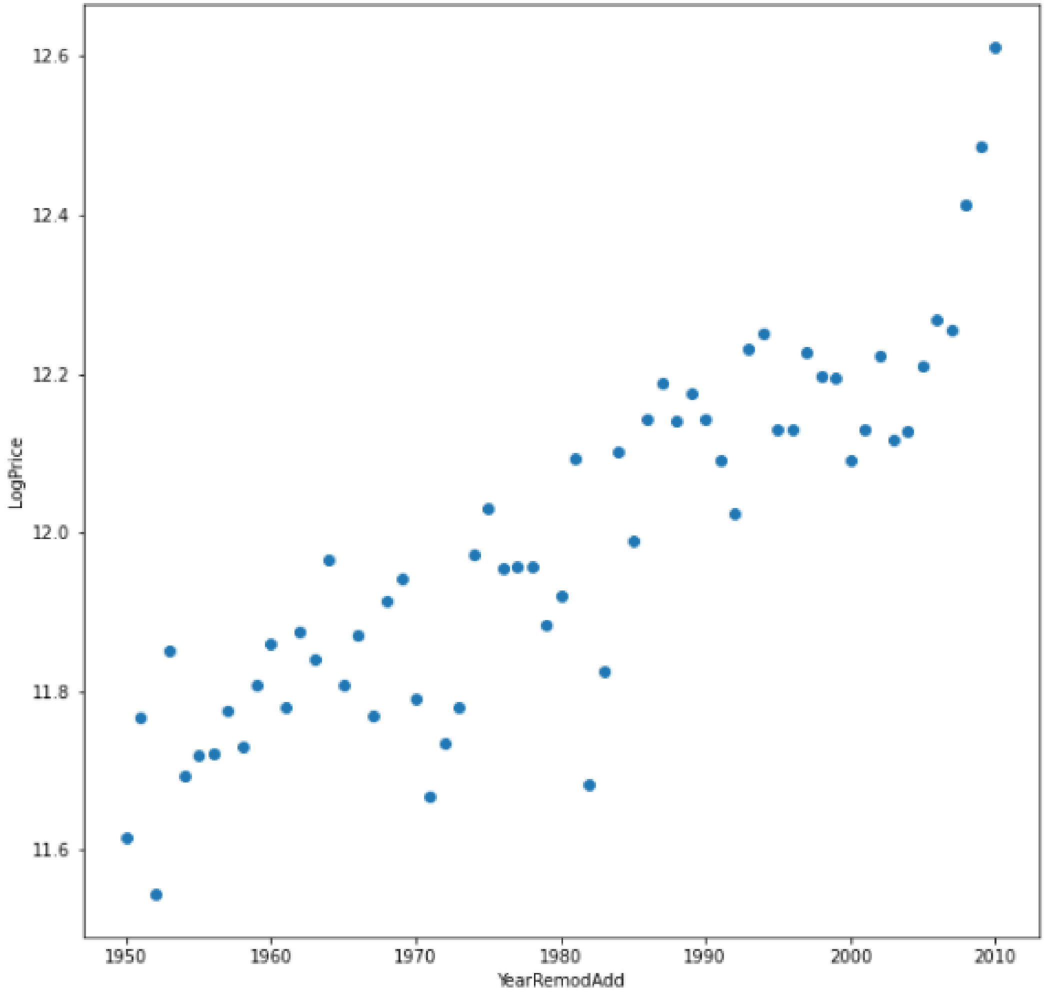


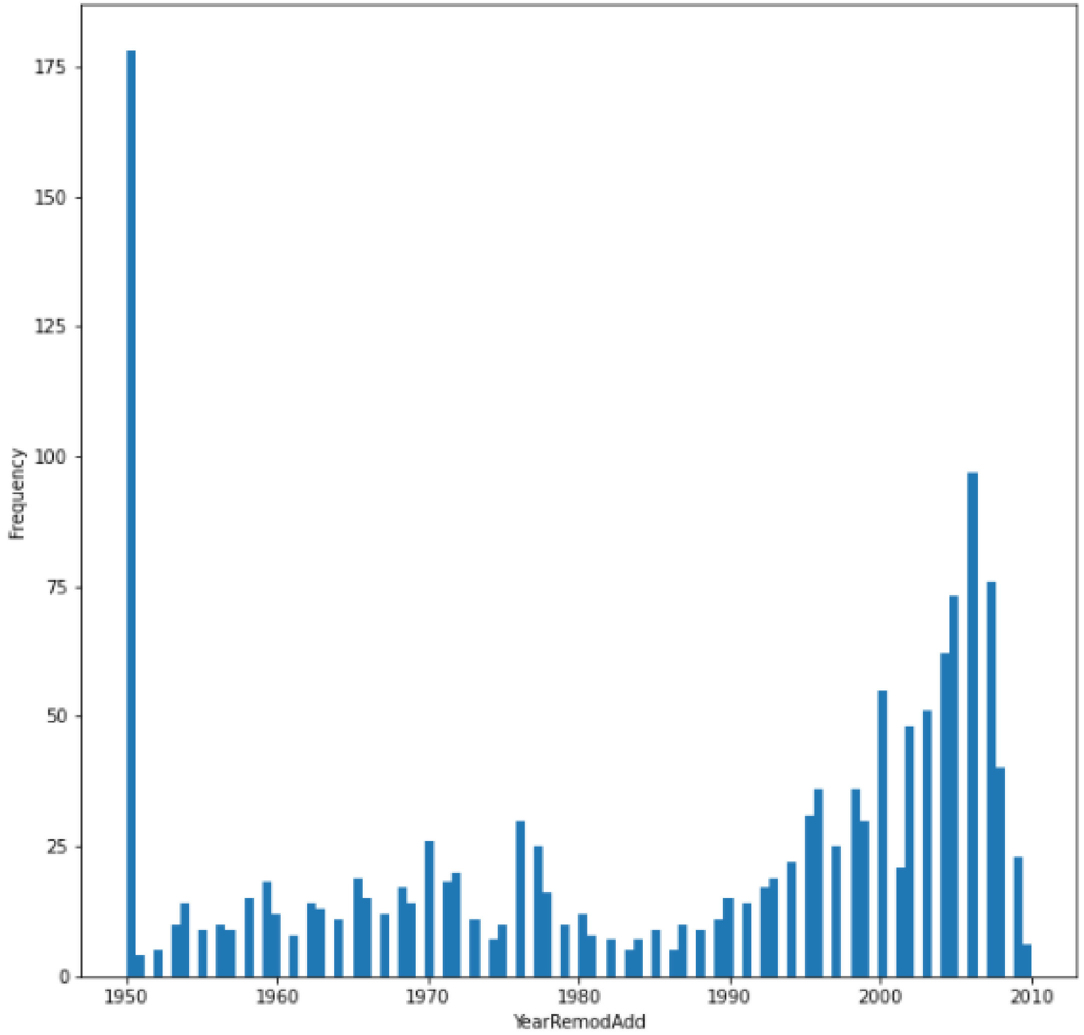
Čini se da su novije kuće poželjnije





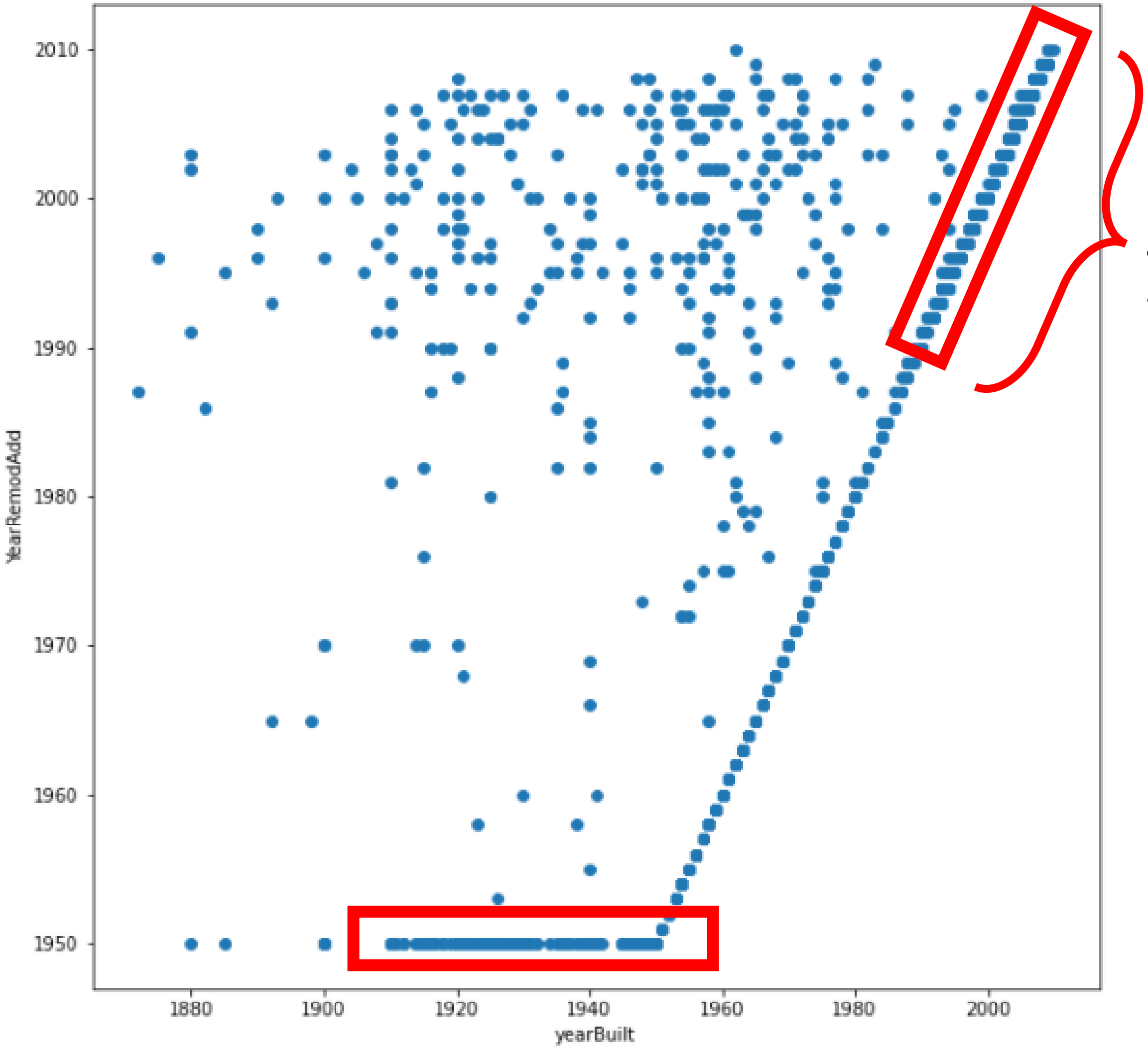
Kao i one nedavno obnavljane





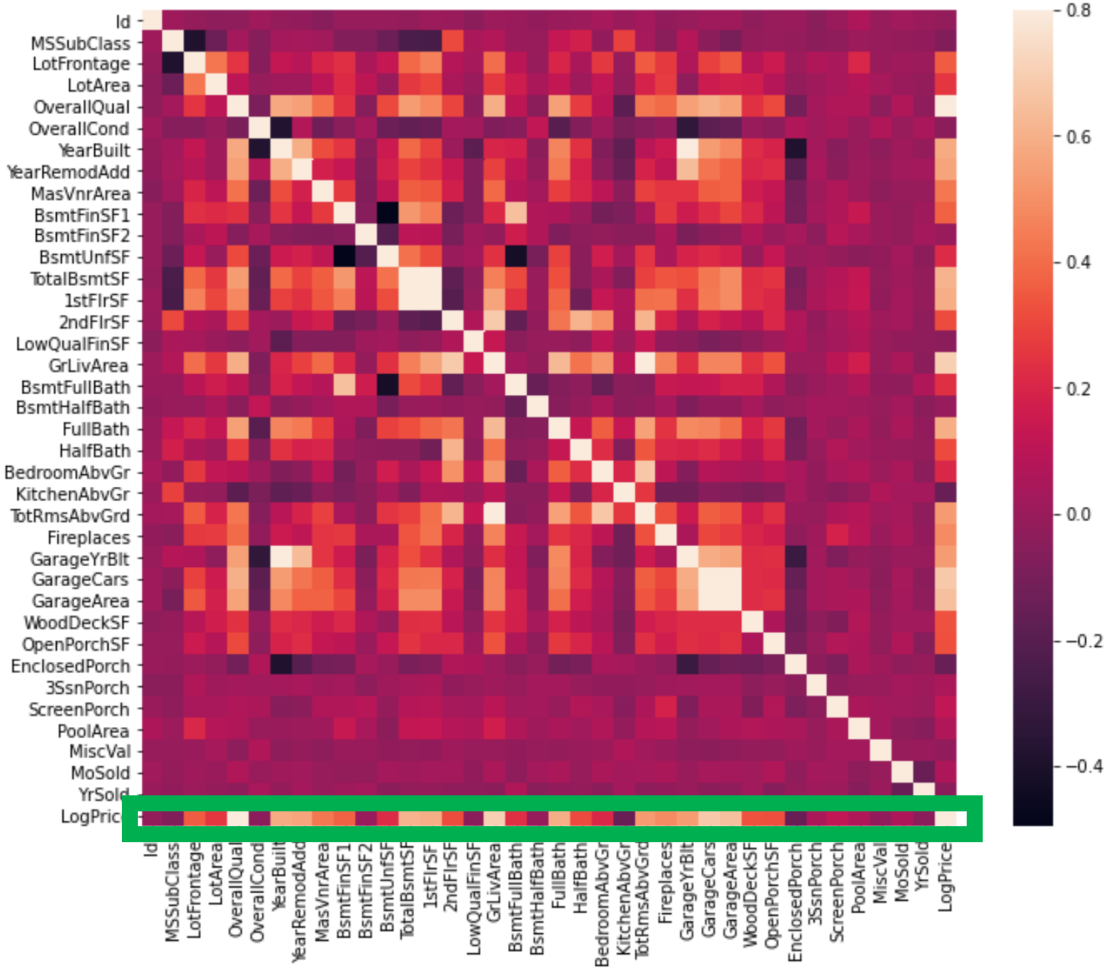
Zanima nas koliko je kuća uopće obnavljano i kada

47% sveukupno

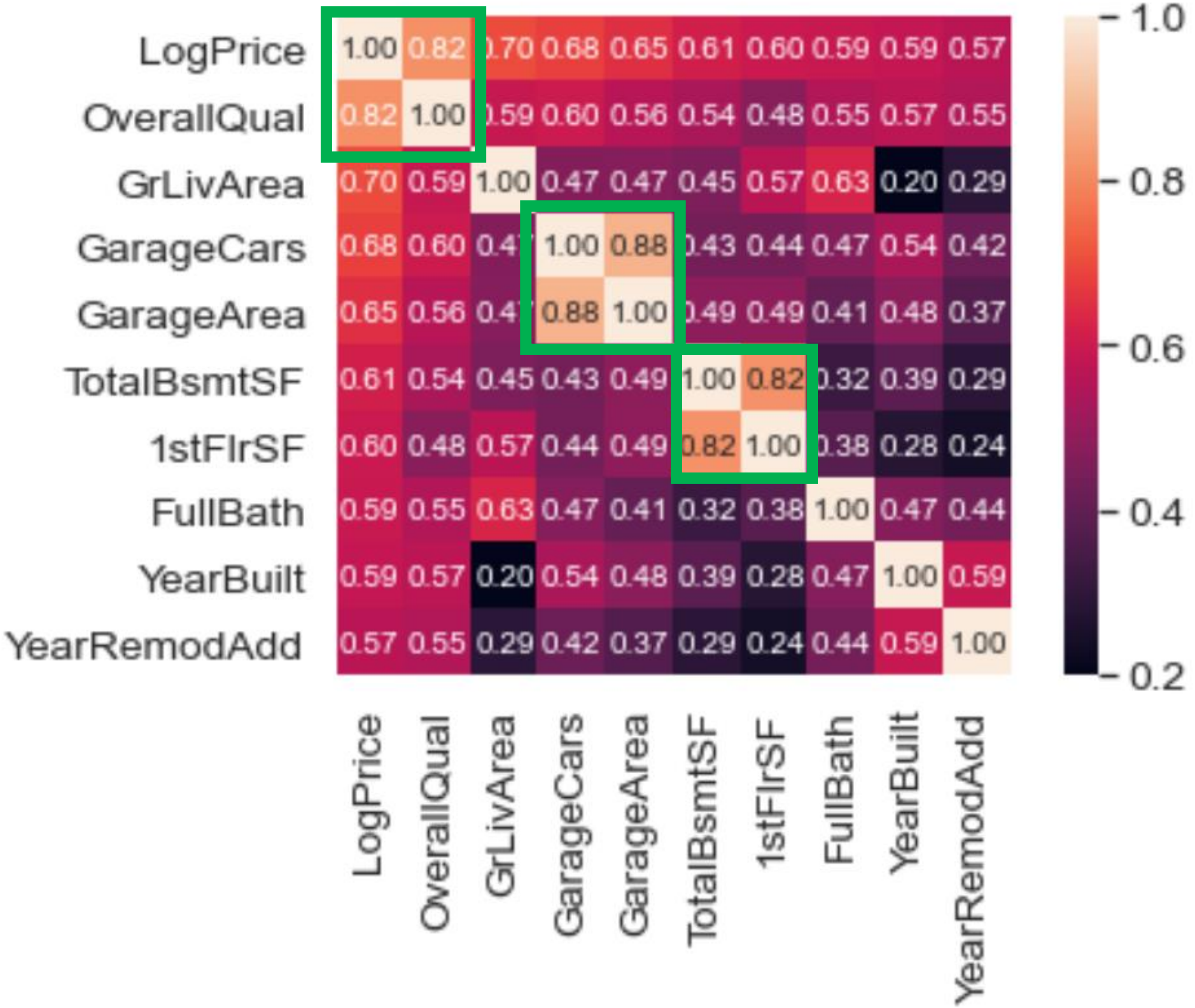


15% godinu dana nakon gradnje

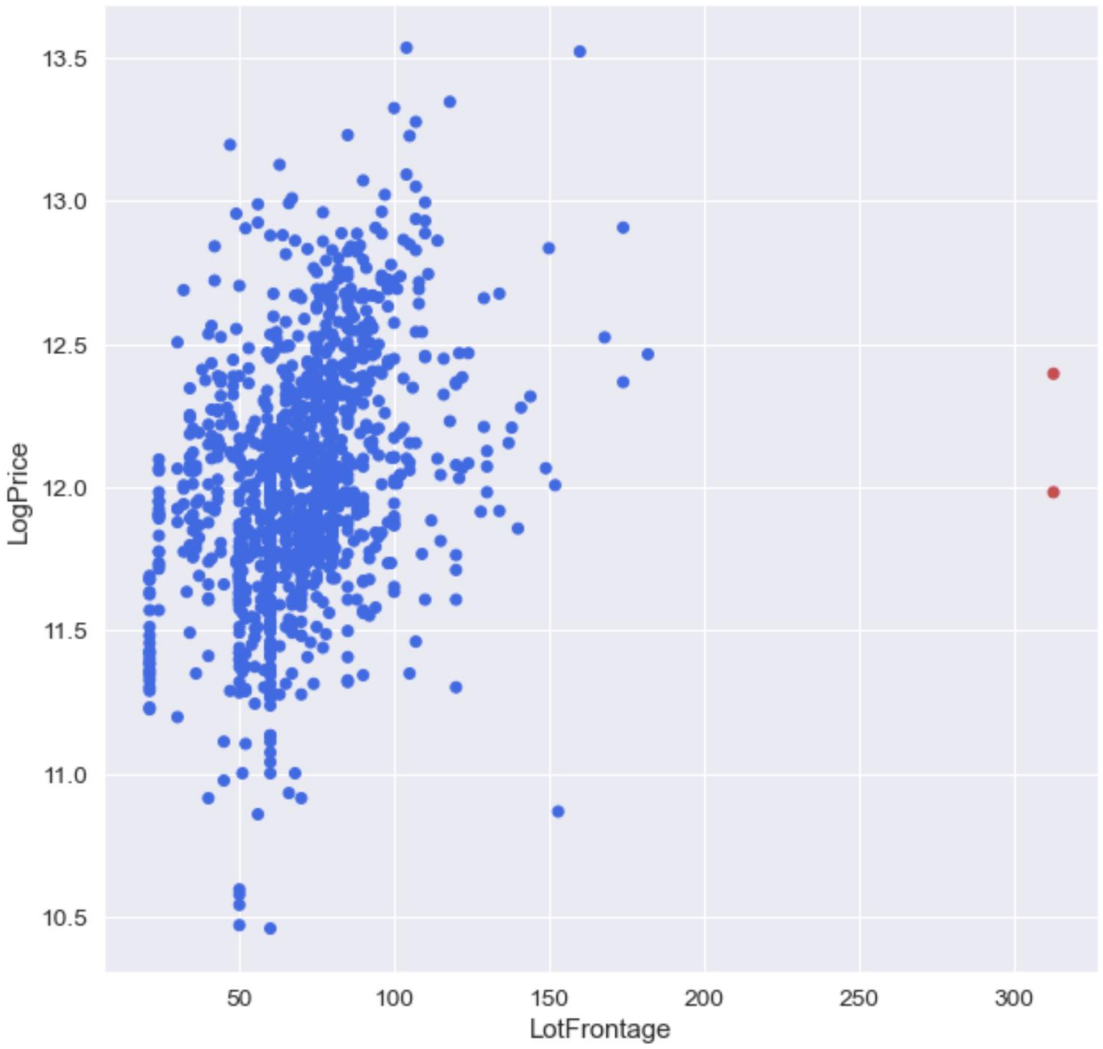
Korelacijska mapa



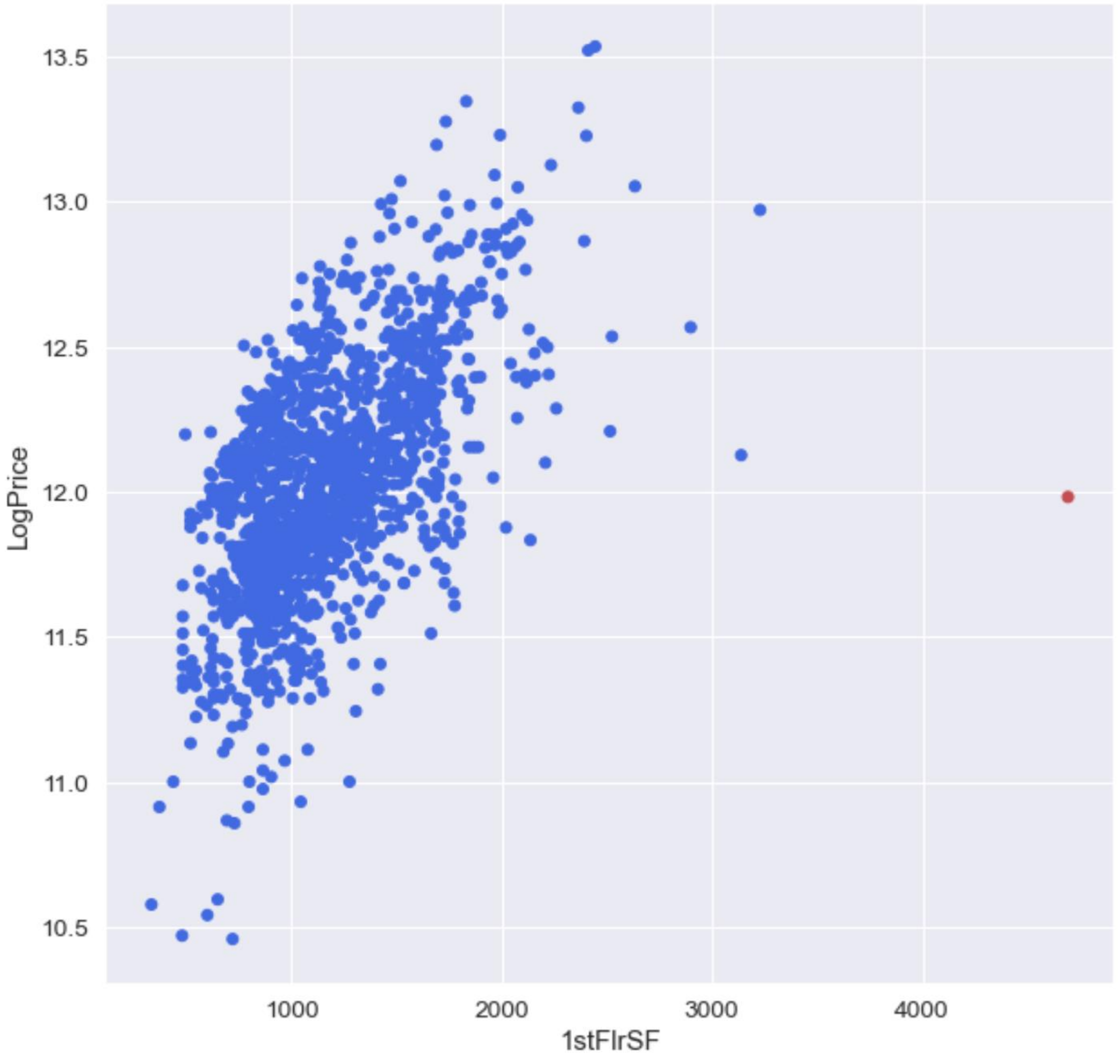
10 najbitnijih?



Promotrimo potencijalne outliere




LogPrice
TotalBsmtSF
13.5
13.0
12.5
12.0
11.5
11.0
10.5
0
1000
2000
3000
4000
5000
6000


LogPrice
1stFlrSF
13.5
13.0
12.5
12.0
11.5
11.0
10.5
1000
2000
3000
4000

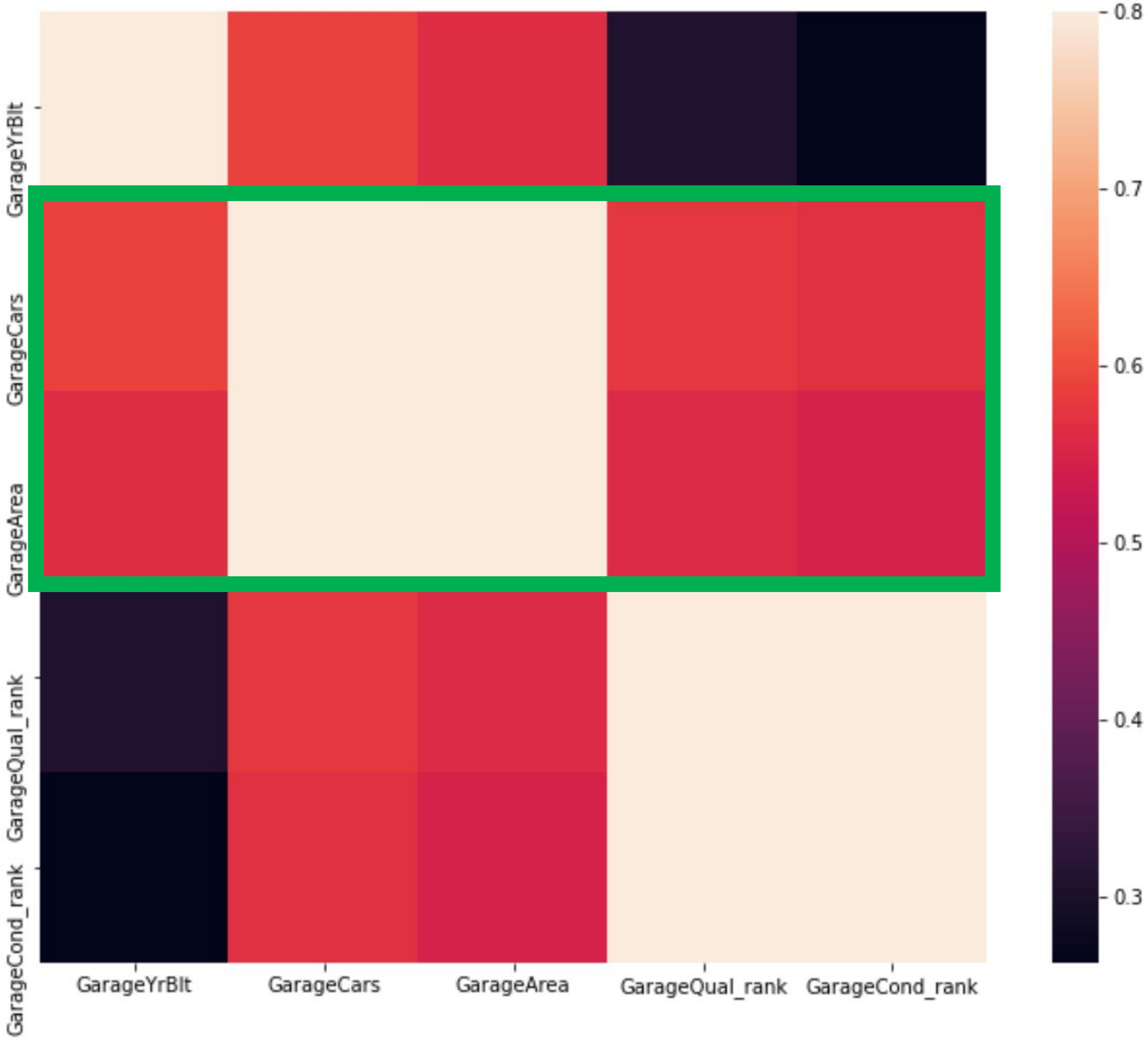
Grupirajmo Missing values:

	Total	Percent
PoolQC	1451	0.996566
MiscFeature	1402	0.962912
Alley	1365	0.937500
Fence	1176	0.807692
FireplaceQu	690	0.473901
LotFrontage	259	0.177885
GarageYrBlt	81	0.055632
GarageFinish	81	0.055632
GarageQual	81	0.055632
GarageCond	81	0.055632
GarageType	81	0.055632
BsmtExposure	38	0.026099
BsmtFinType2	38	0.026099
BsmtCond	37	0.025412
BsmtQual	37	0.025412
BsmtFinType1	37	0.025412
MasVnrArea	8	0.005495
MasVnrType	8	0.005495
Electrical	1	0.000687

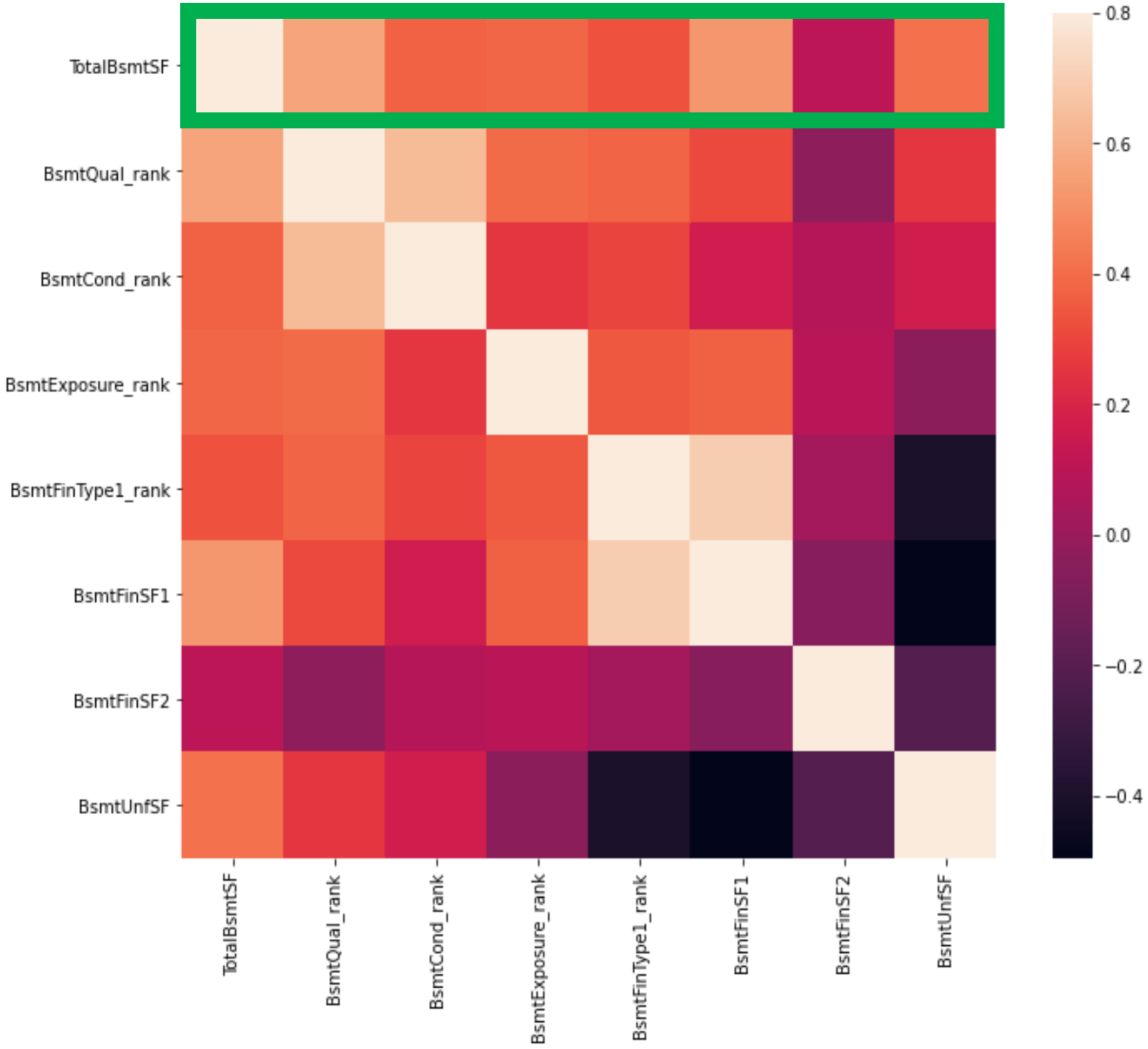
GARAŽA

PODRUM

Površina garaže nam je dovoljna!



Isto vrijedi i za podrum!



Izbacimo Missing Values:

	Total	Percent
PoolQC	1451	0.996566
MiscFeature	1402	0.962912
Alley	1365	0.937500
Fence	1176	0.807692
FireplaceQu	690	0.473901
LotFrontage	259	0.177885
GarageYrBlt	81	0.055632
GarageFinish	81	0.055632
GarageQual	81	0.055632
GarageCond	81	0.055632
GarageType	81	0.055632
BsmtExposure	38	0.026099
BsmtFinType2	38	0.026099
BsmtCond	37	0.025412
BsmtQual	37	0.025412
BsmtFinType1	37	0.025412
MasVnrArea	8	0.005495
MasVnrType	8	0.005495
Electrical	1	0.000687

GARAŽA

PODRUM

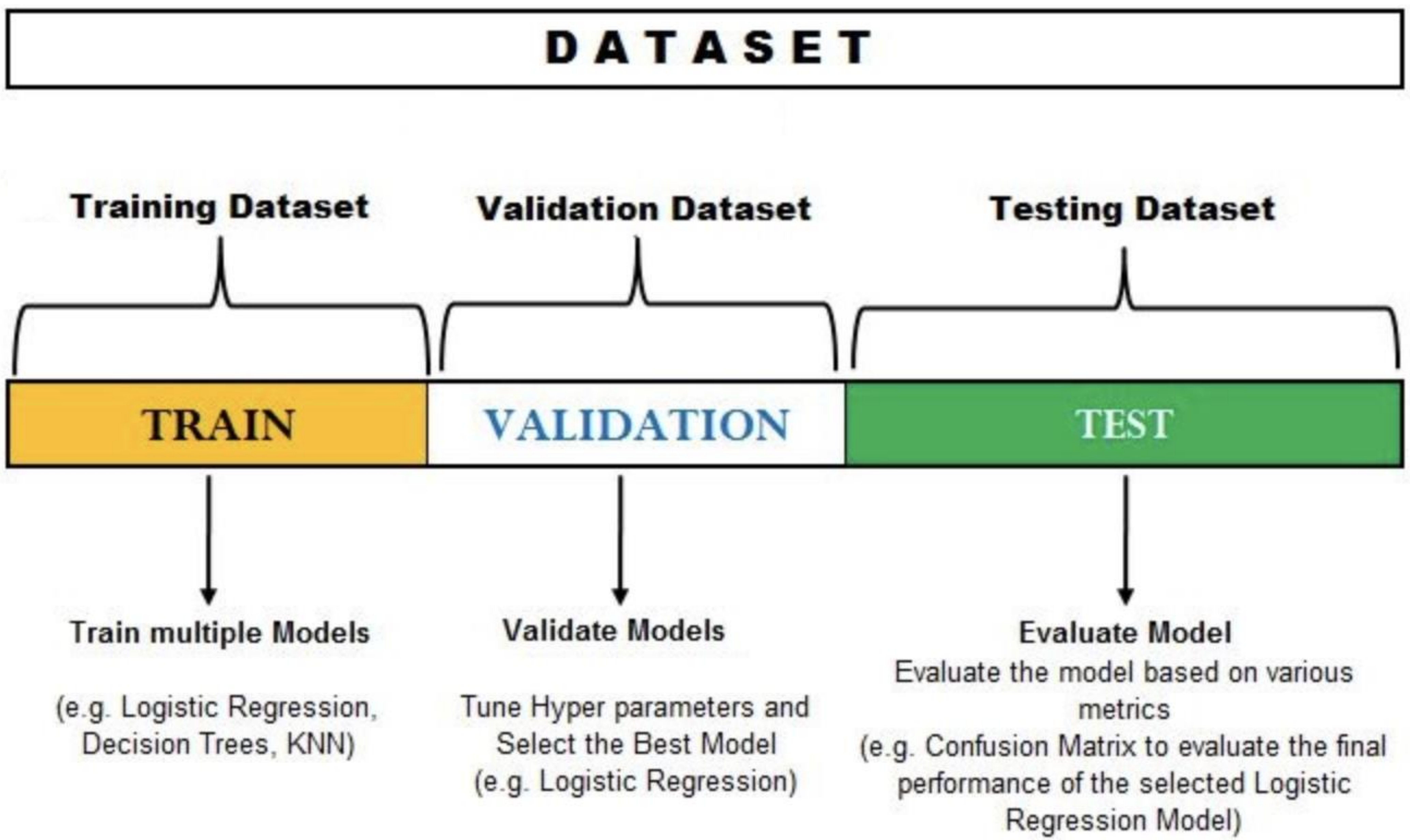
II. Linearna regresija

SADRŽAJ

- Uvod u problem
- Rekapitulacija opisne statistike
- Linearna regresija – uvod
- Regresija – modeli

SADRŽAJ

- Uvod u problem
- Rekapitulacija opisne statistike
- Linearna regresija – uvod
- Regresija – modeli


DATASET
Training Dataset
Validation Dataset
Testing Dataset
TRAIN
VALIDATION
TEST
Train multiple Models
(e.g. Logistic Regression, Decision Trees, KNN)
Validate Models
Tune Hyper parameters and Select the Best Model
(e.g. Logistic Regression)
Evaluate Model
Evaluate the model based on various metrics
(e.g. Confusion Matrix to evaluate the final performance of the selected Logistic Regression Model)

kaggle™

PODACI

- data_description.txt
- sample_submission.csv
- test.csv
- train.csv

METRIKA

$$RMSE_{\log} = \sqrt{\frac{\sum_{k=1}^{n} (\log \hat{y}_k - \log y_k)^2}{n}}$$

y_k = stvarna cijena kuće

$\hat{y}_k$ = cijena koju je predvidio model

Logaritam!

```
In [5]: train
```

Out[5]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1455	1456	60	RL	62.0	7917	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0
1456	1457	20	RL	85.0	13175	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0
1457	1458	70	RL	66.0	9042	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	GdPrv	Shed	2500
1458	1459	20	RL	68.0	9717	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0
1459	1460	20	RL	75.0	9937	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0

1460 rows × 81 columns

PODACI

- 1460 kuća
- 79 nezavisnih varijabli: 33 numeričke i 46 kategorijskih

SADRŽAJ

- Uvod u problem
- Rekapitulacija opisne statistike
- Linearna regresija – uvod
- Regresija – modeli

Izbacimo Missing Values:

	Total	Percent
PoolQC	1451	0.996566
MiscFeature	1402	0.962912
Alley	1365	0.937500
Fence	1176	0.807692
FireplaceQu	690	0.473901
BsmtFinType1	37	0.025412
MasVnrArea	8	0.005495
MasVnrType	8	0.005495
Electrical	1	0.000687

NE!!!

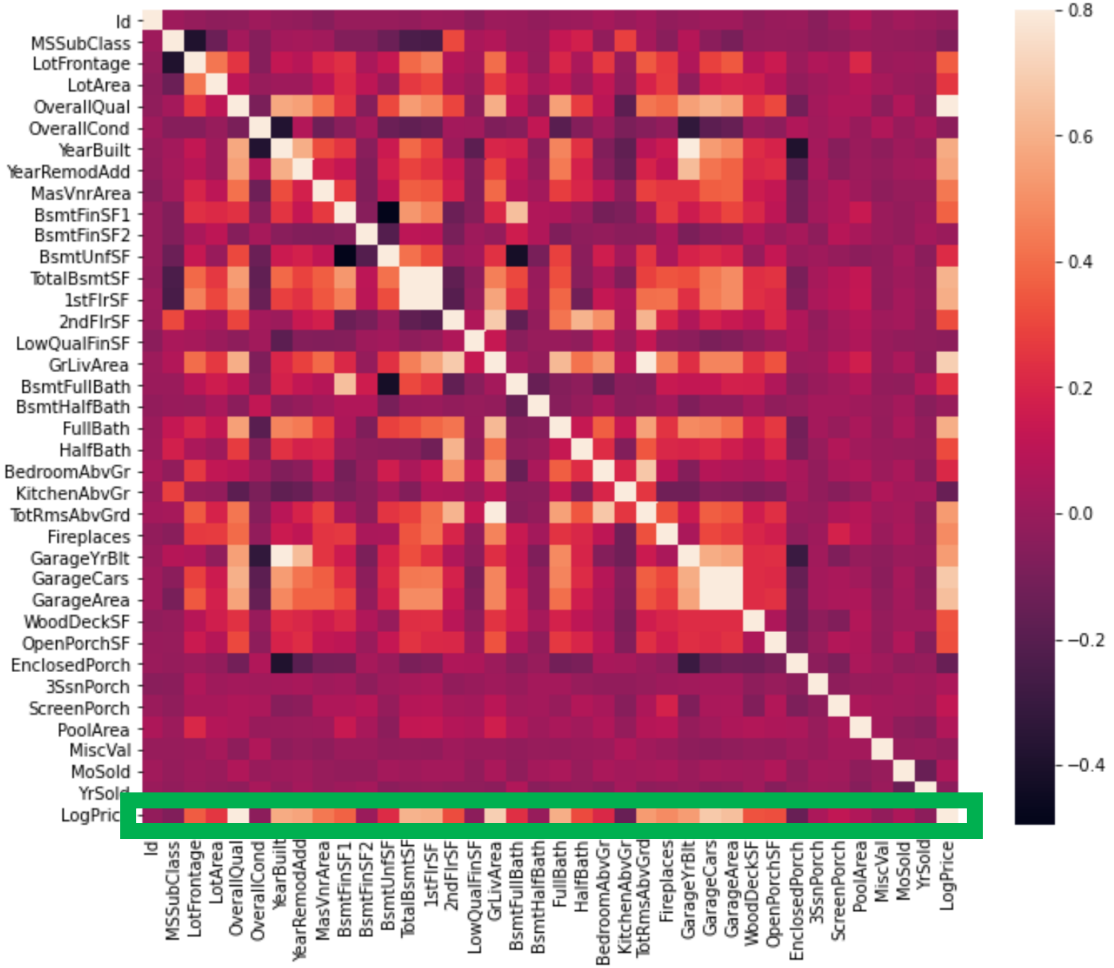
MISSING VALUES

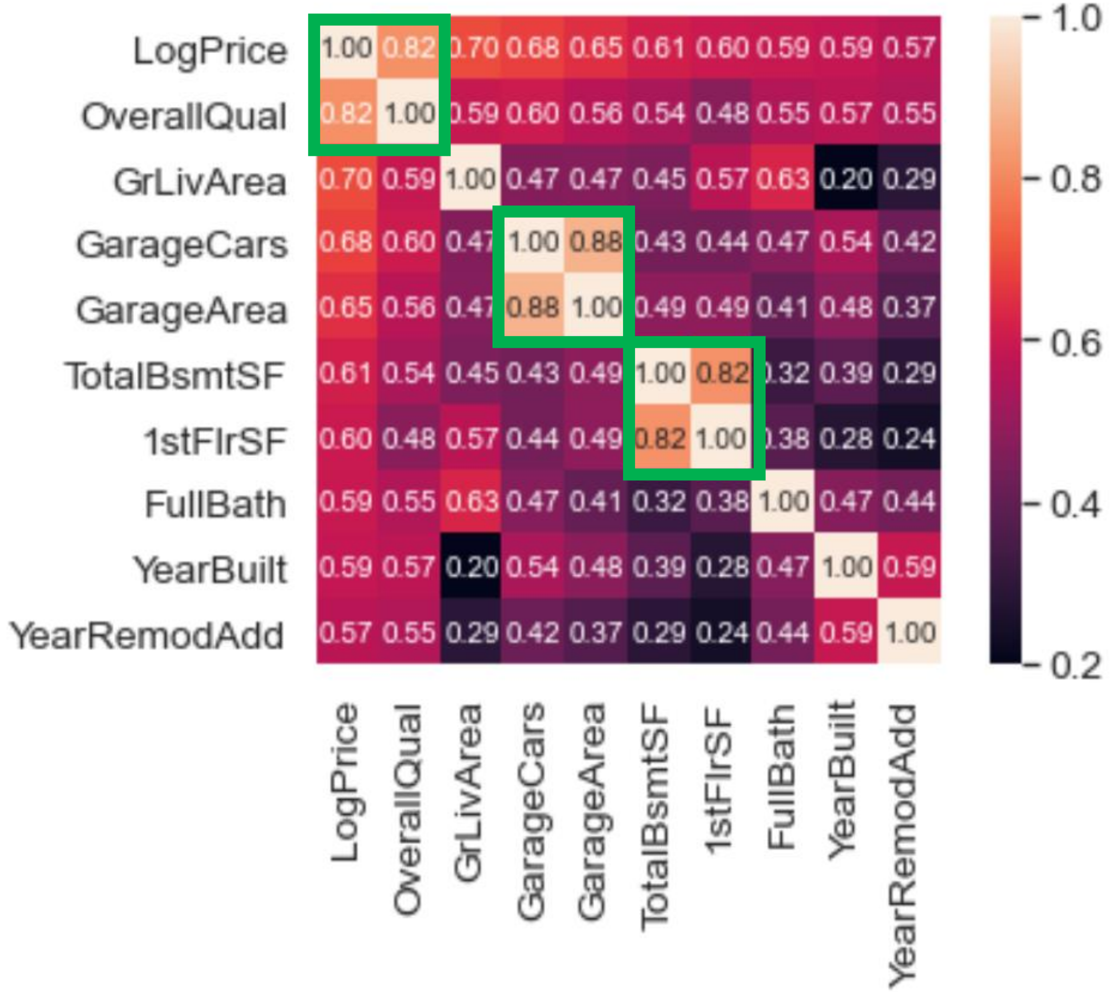
	Total	Percent
~~PoolQC~~	~~1451~~	~~0.996566~~
~~MiscFeature~~	~~1402~~	~~0.962912~~
~~Alley~~	~~1365~~	~~0.937500~~
~~Fence~~	~~1176~~	~~0.807692~~
~~FireplaceQu~~	~~690~~	~~0.473901~~
~~LotFrontage~~	~~259~~	~~0.177885~~
~~GarageYrBlt~~	~~81~~	~~0.055632~~
~~GarageFinish~~	~~81~~	~~0.055632~~
~~GarageQual~~	~~81~~	~~0.055632~~
~~GarageCond~~	~~81~~	~~0.055632~~
~~GarageType~~	~~81~~	~~0.055632~~
BsmtExposure	38	0.026099
BsmtFinType2	38	0.026099
BsmtCond	37	0.025412
BsmtQual	37	0.025412
BsmtFinType1	37	0.025412
MasVnrArea	8	0.005495
MasVnrType	8	0.005495
Electrical	1	0.000687

GARAŽA

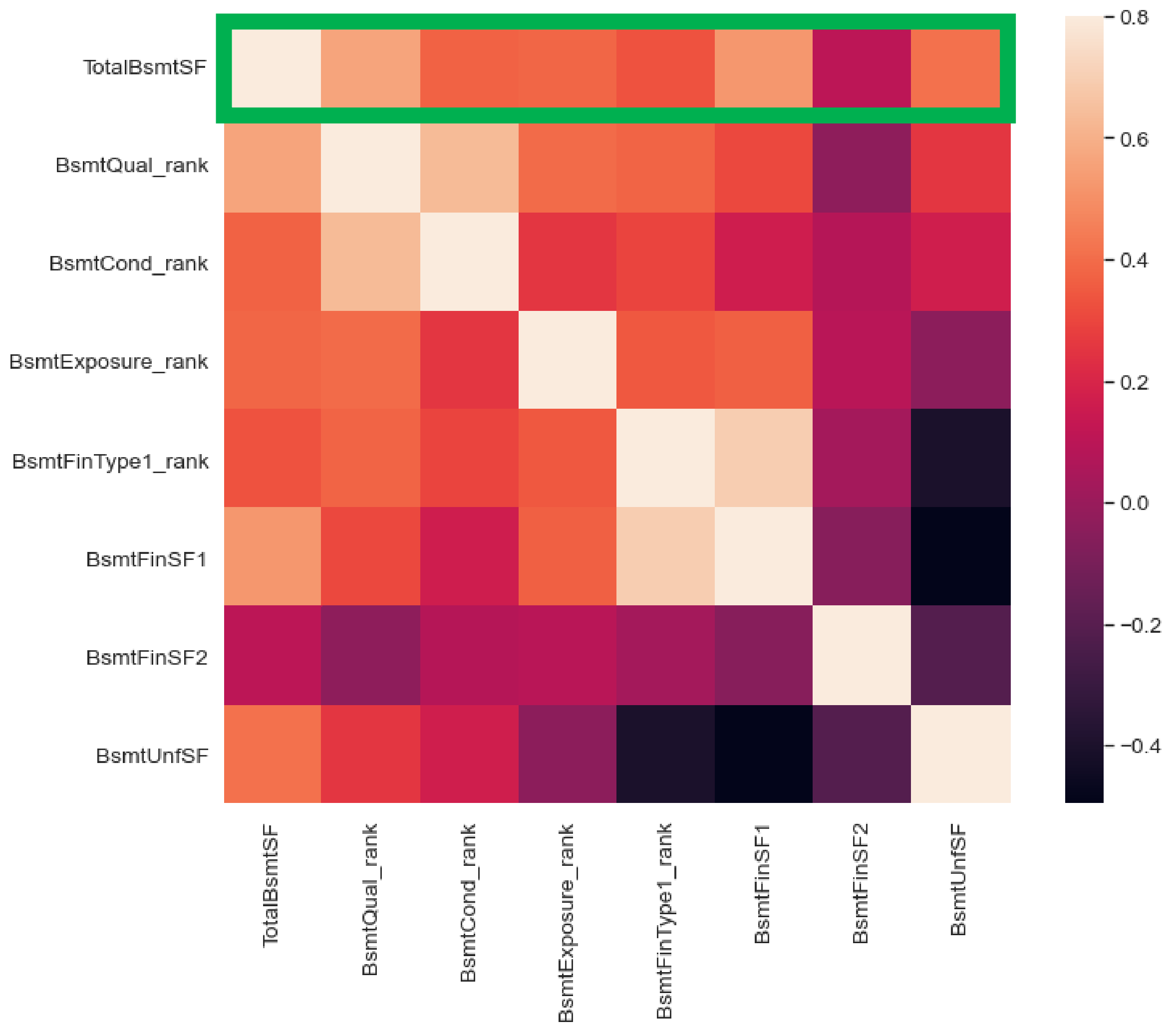
PODRUM

KORELACIJSKA MAPA OLAKŠAVA ODABIR VARIJABLI

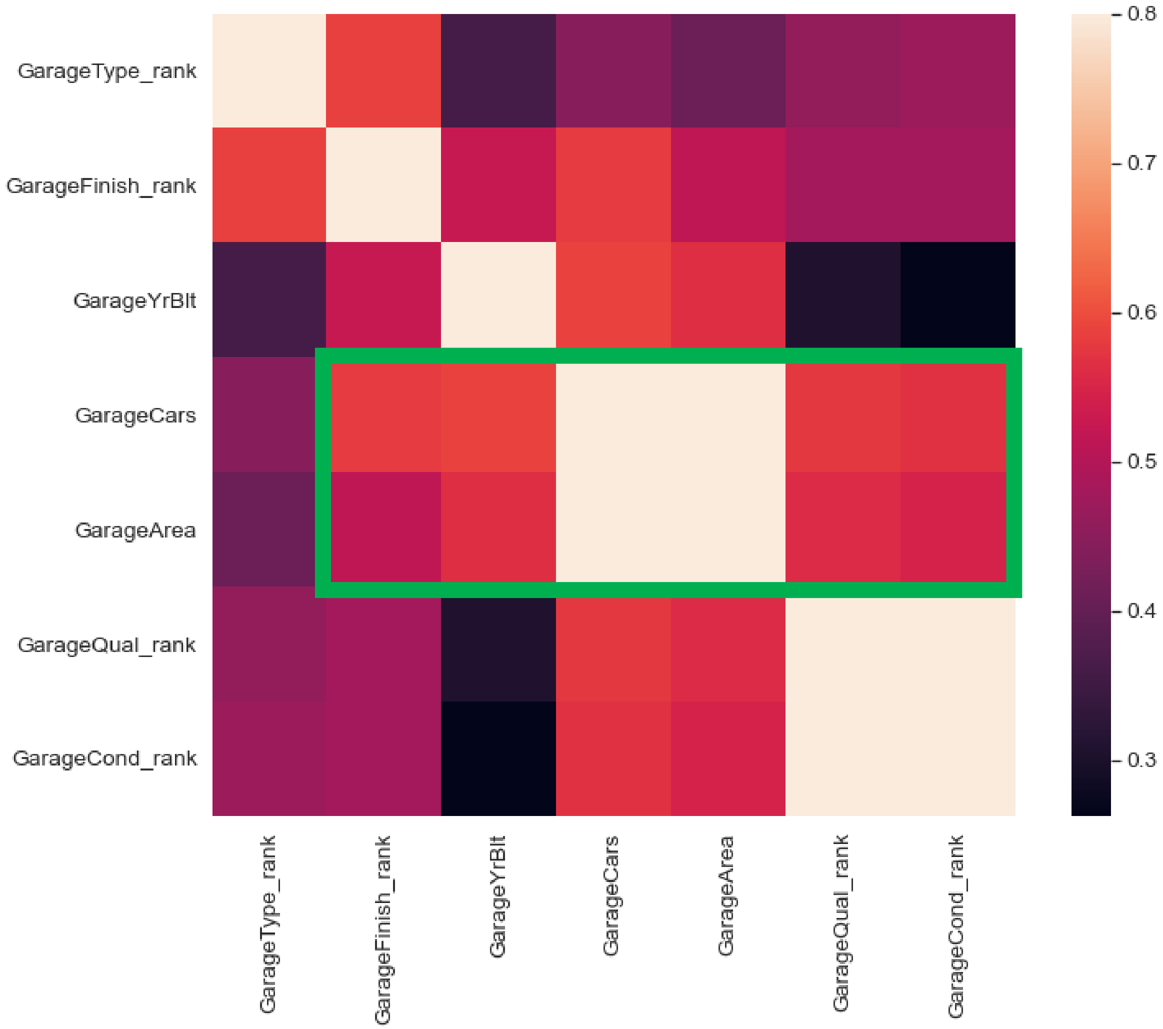




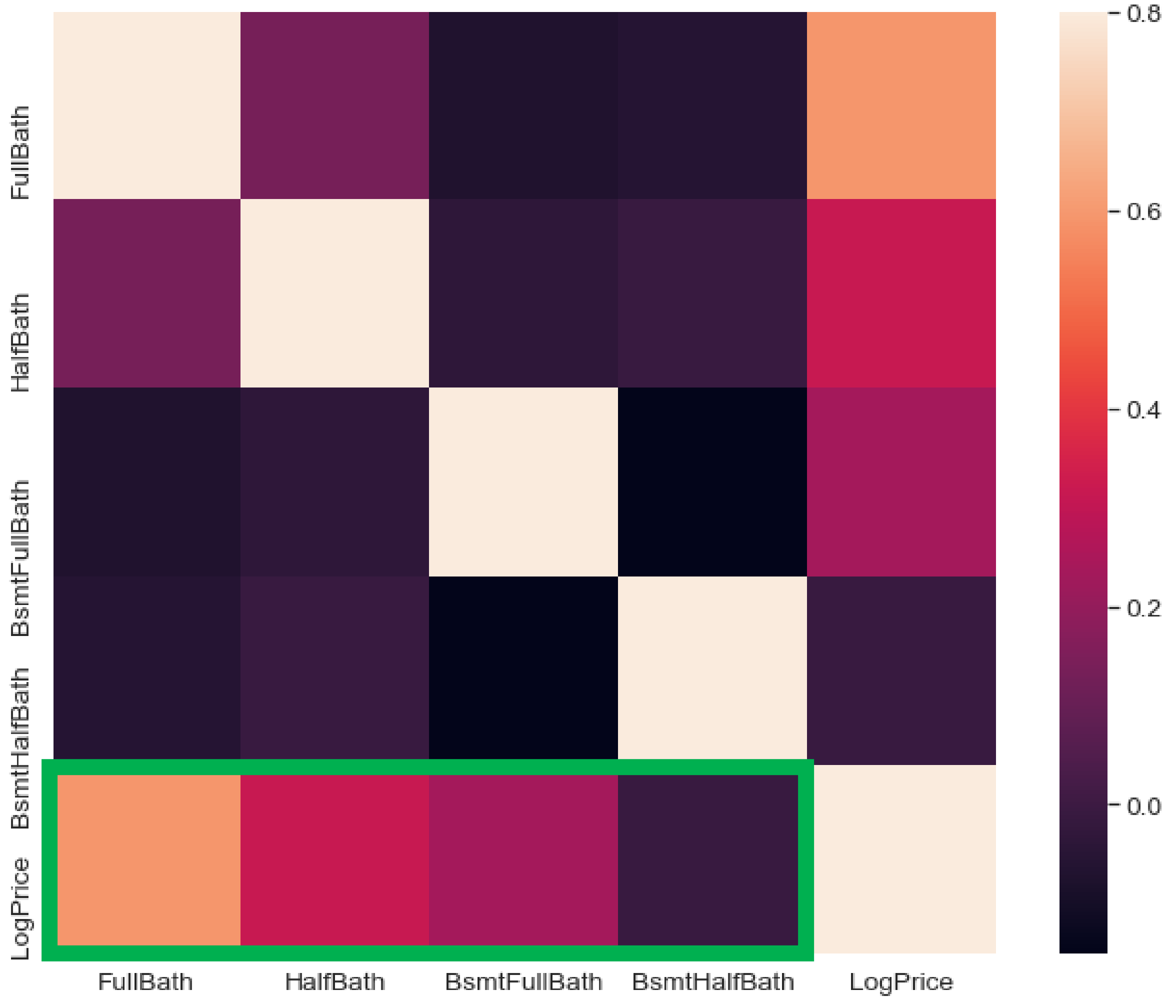
BASEMENT VARIJABLE



GARAGE VARIJABLE



BATH VARIJABLE



SADRŽAJ

- Uvod u problem
- Rekapitulacija opisne statistike
- Linearna regresija – uvod
- Regresija – modeli

POMOĆU VALIDACIJSKIH FUNKCIJA ODREĐUJEMO NAJBOLJI MODEL

$$R^2 = 1 - \frac{SS_{residuals}}{SS_{total}}$$

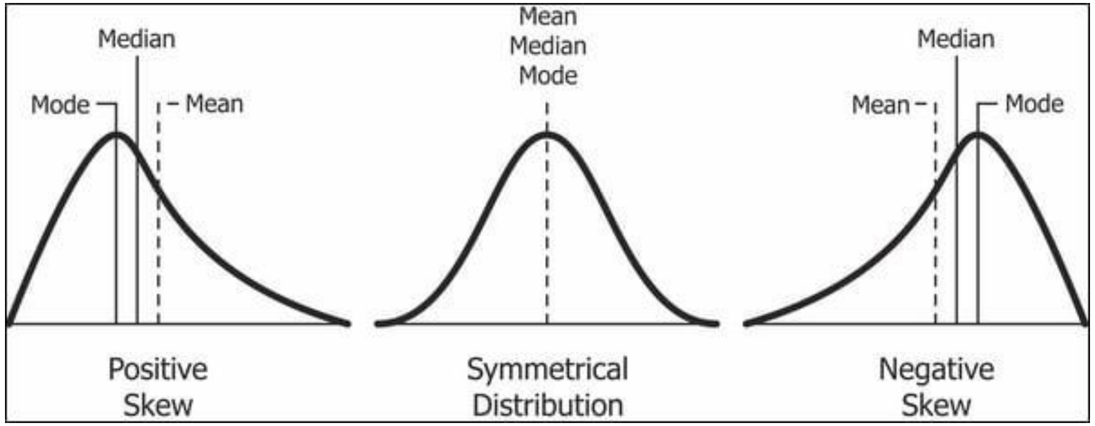
$$\text{Adjusted } R^2 = 1 - \frac{SS_{residuals} / (n-K)}{SS_{total} / (n-1)}$$

$$MSE = \frac{1}{n} \Sigma \underbrace{\left(y - \hat{y} \right)^2}_{\text{The square of the difference between actual and predicted}}$$

Koristimo SKEWNESS za „popravljanje” podataka

Što je SKEWNES?

- mjera asimetričnosti funkcije distribucije slučajne varijable realne vrijednosti u odnosu na njezinu srednju vrijednost
- pozitivna, nula, negativna i nedefinirana




```
skewed_feats = train[numeric_feats].apply(lambda x: skew(x.dropna()))
skewed_feats = skewed_feats[skewed_feats > 0.75]
skewed_feats = skewed_feats.index

train[skewed_feats] = np.log1p(train[skewed_feats])
```

1.KORAK: dummy varijable

79 nezavisnih varijabli

249 nezavisnih varijabli

Water	Temperature
A	Hot
B	Cold
C	Warm
D	Cold

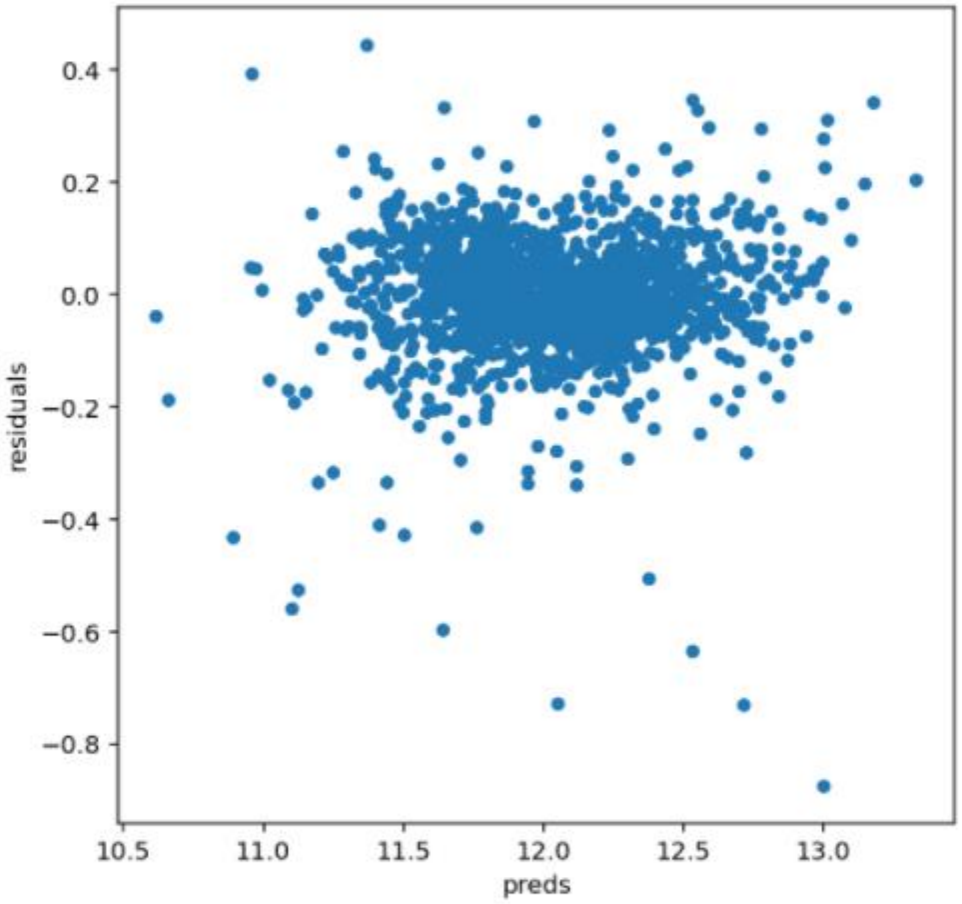
Dummy Variables

Water	Temperature	var_hot	var_warm	var_cold
A	Hot	1	0	0
B	Cold	0	0	1
C	Warm	0	1	0
D	Cold	1	0	0

2.KORAK: Linearna regresija s dummy varijablama

```
Train set evaluation:
_____________________________________
MAE: 0.06645438387074533
MSE: 0.008699787924455868
RMSE: 0.09327265367971402
R2 Square 0.9455260794693383
__________________________________
0.934333426369417 7
```

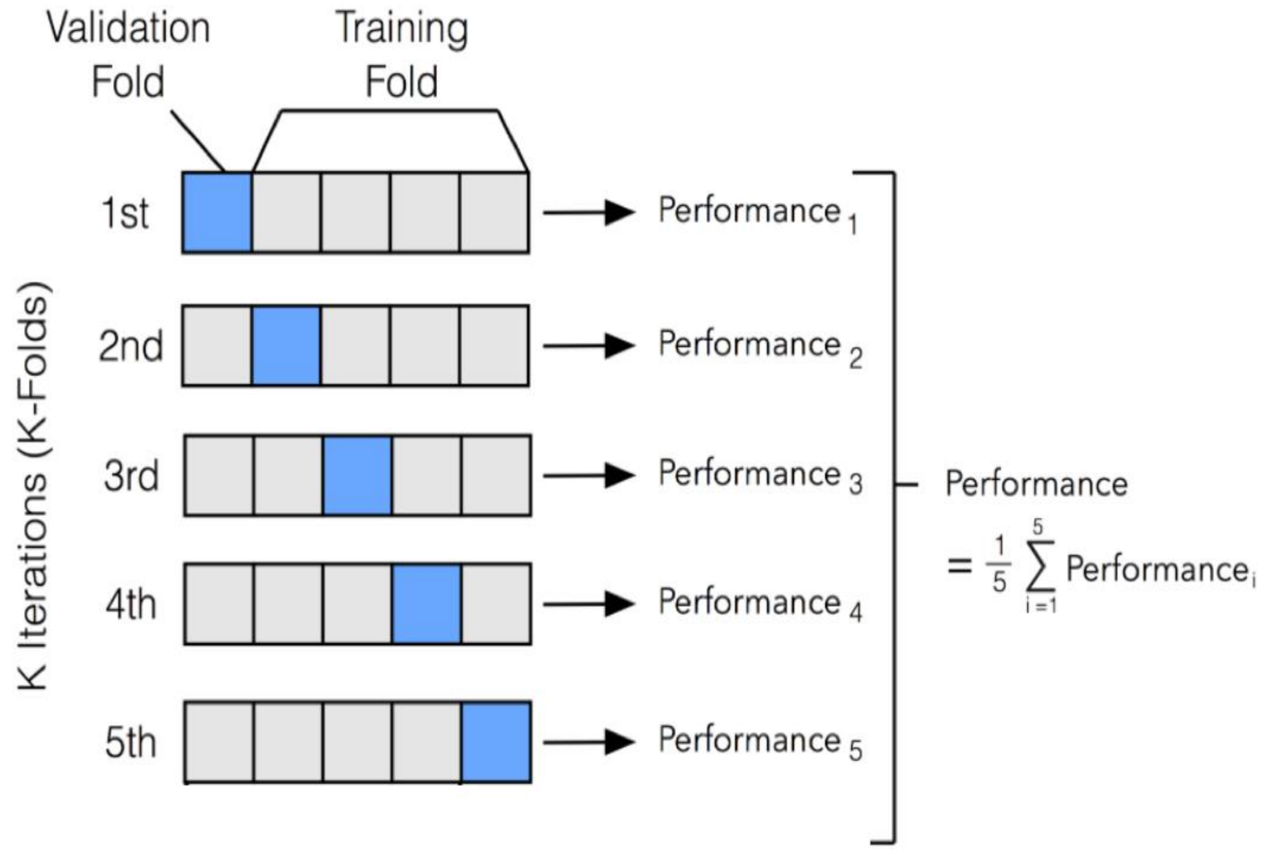
NIJE LOŠE? DA, ALI NE.



3.KORAK: k-fold cross validation protiv overfitting

1. Podijeli train set na k = 5 jednakih dijelova
2. Fiksiraj 1 dio podataka za validiranje modela
3. „Treniraj” na preostala 4 dijela podataka
4. Validiraj model dobiven„treniranjem” na dijelu koji si fiksirao
5. Izračunaj RMSE modela
6. Ponovi postupak 2. – 5. dok ne prođeš svaki dio podataka za testiranje
7. Uzmi srednju vrijednost RMSE za svaki k=1,..,5

```python
def rmse_cv(model, X, y):
    rmse = np.sqrt(-cross_val_score(model, X, y, scoring="neg_mean_squared_error", cv = 5))
    return(rmse.mean())
```



4.KORAK:
Ridge regresija protiv overfitting

metoda najmanjih kvadrata	regularizacija
$\underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\lVert y - X\beta \rVert_2^2}_{\text{Loss}}$	$\underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\lVert y - X\beta \rVert_2^2}_{\text{Loss}} + \lambda \underbrace{\lVert \beta \rVert_2^2}_{\text{Penalty}}$

SMANJUJE VARIJANCU

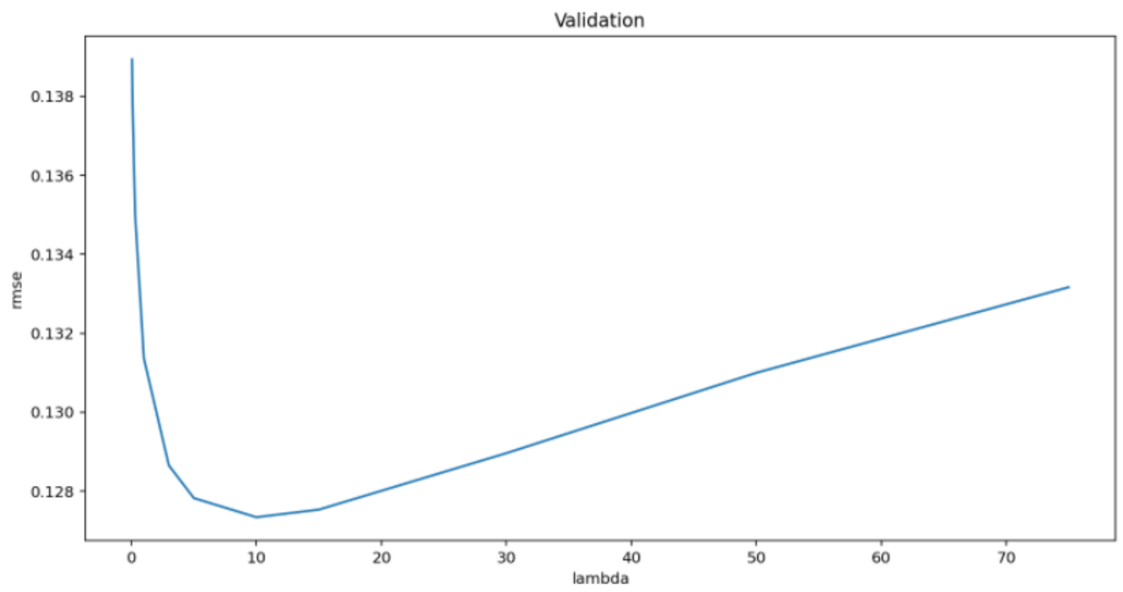
manje šanse za overfitting

Odabir parametra regularizacije za model s dummy varijablama

```
alphas = [0.05, 0.1, 0.3, 0.5, 0.7, 1, 2, 5, 10, 20, 30, 50]
cv_ridge = [rmse_cv(Ridge(alpha = alpha).fit(X, y), X, y)
            for alpha in alphas]
```

→ $\lambda = 10$ → $RMSE_{cross} = 0.1221$

Dakle, koristit ćemo Ridge regresiju za daljnje modele.



SADRŽAJ

- Uvod u problem
- Rekapitulacija opisne statistike
- Linearna regresija – uvod
- Regresija – modeli

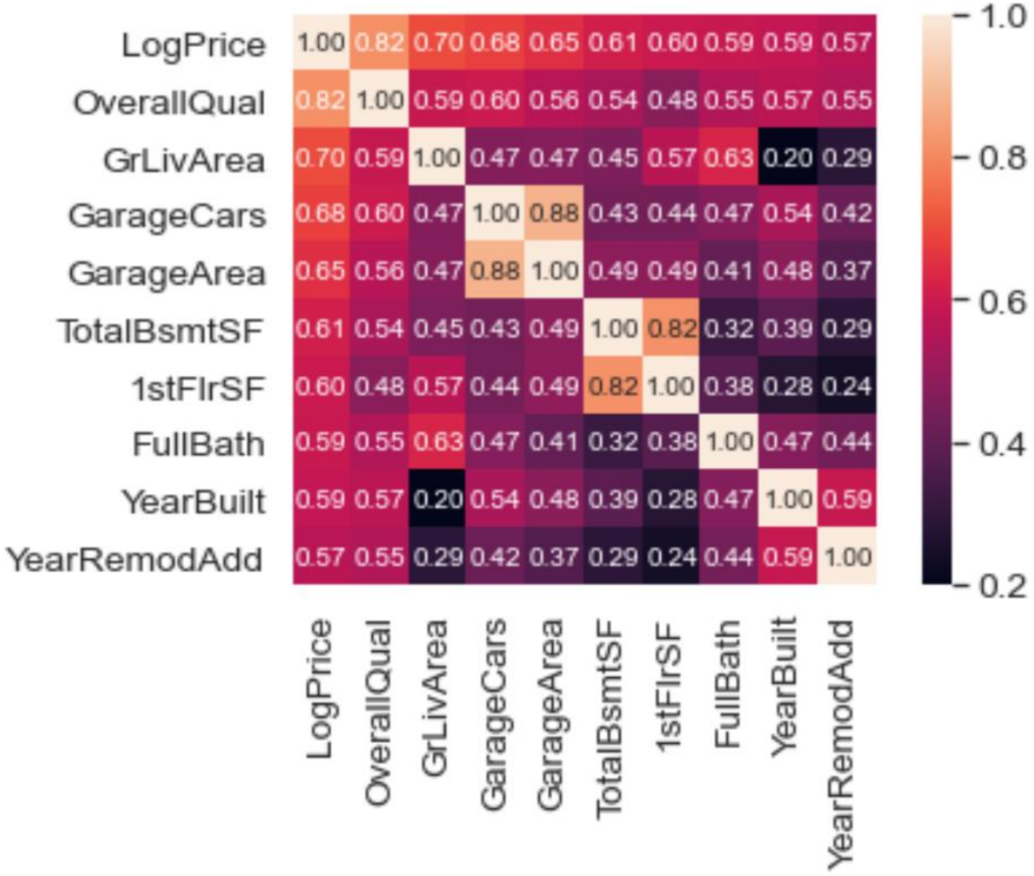
Ridge regresija – „mali” model

- nakon izbacivanja outliera, missing values i „prepravljanja” podataka sa skewness, provodimo Ridge regresiju
- uzimamo 9 najkoreliranijih numeričkih varijabli

→ $RMSE_{cross}$ = 0.168942

```
X = train[selected]
mali_model = Ridge(alpha = 0.01).fit(X, y)
rmse_cv(mali_model, X, y)
```

```
0.16894260231430222
```



Ridge regresija – „srednji” model

3626	Sanjin Juric Fot	0.15965	1	1s

Your First Entry ↑

Welcome to the leaderboard!

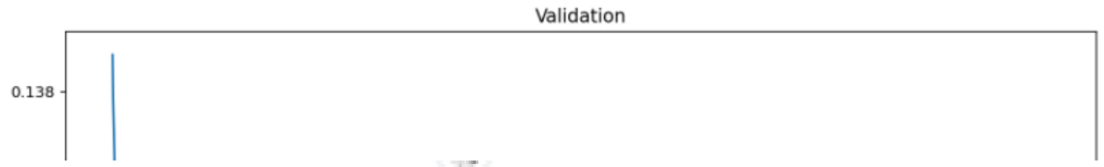
3627	Sirui Shao	0.15967	26	5d

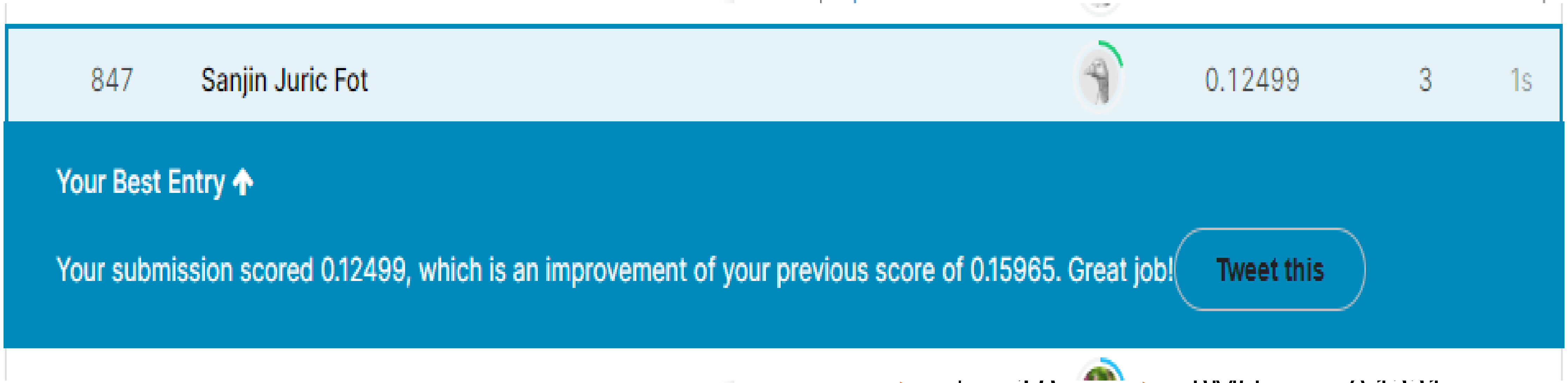
```
for zona in zone:
    X['LotArea*'+zona] = train['LotArea']*train[zona]
    X['GarageArea*'+zona] = train['GarageArea']*train[zona]
```

```
srednji_model = Ridge(alpha = 10).fit(X, y)
rmse_cv(srednji_model, X, y)
```

0.12625691440016526

Ridge regresija – „veliki” model



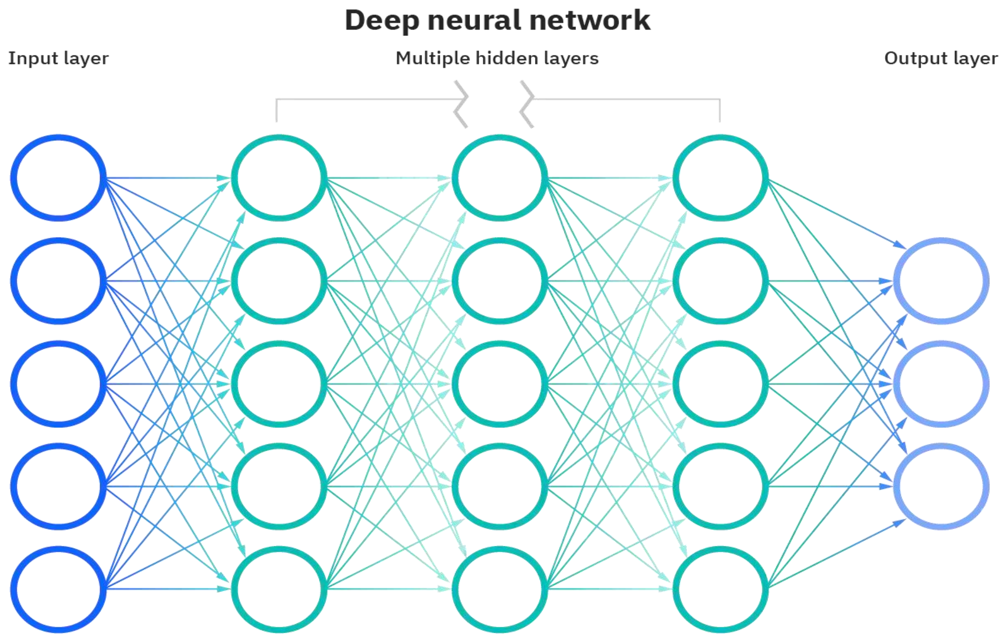


→ $\lambda = 10$ → $RMSE_{cross} = 0.1221$

III. Neuronske mreže

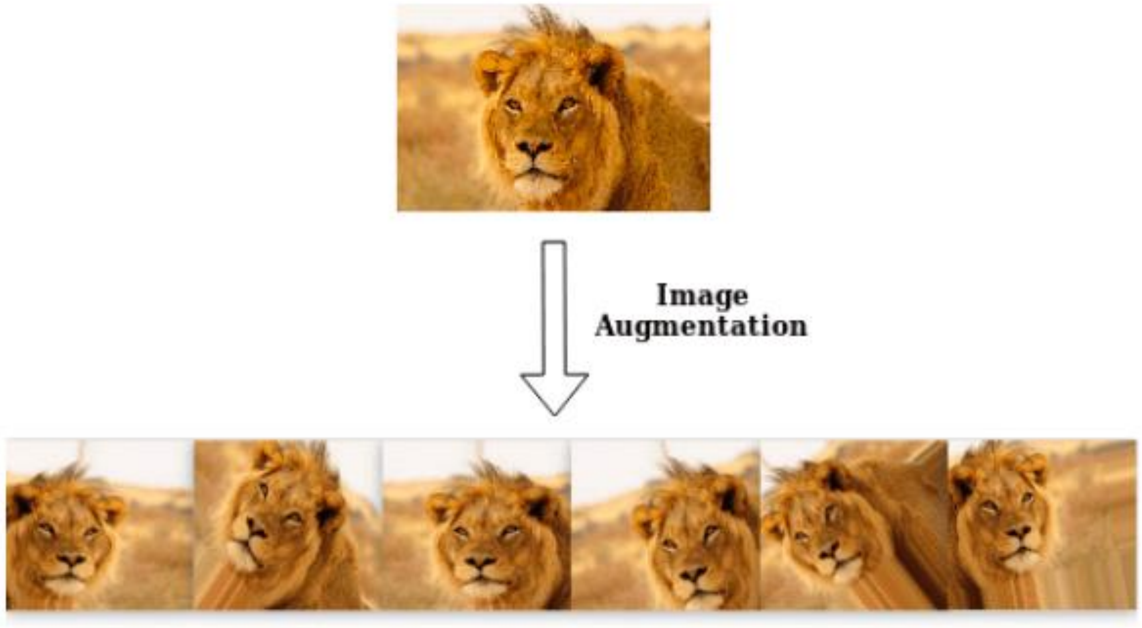
NEURONSKE MREŽE

- Koristimo samo dense slojeve
- Loši rezultati (0.47) → Premalo podataka?
- Kada treniramo na nešto manjem skupu još lošiji rezultati



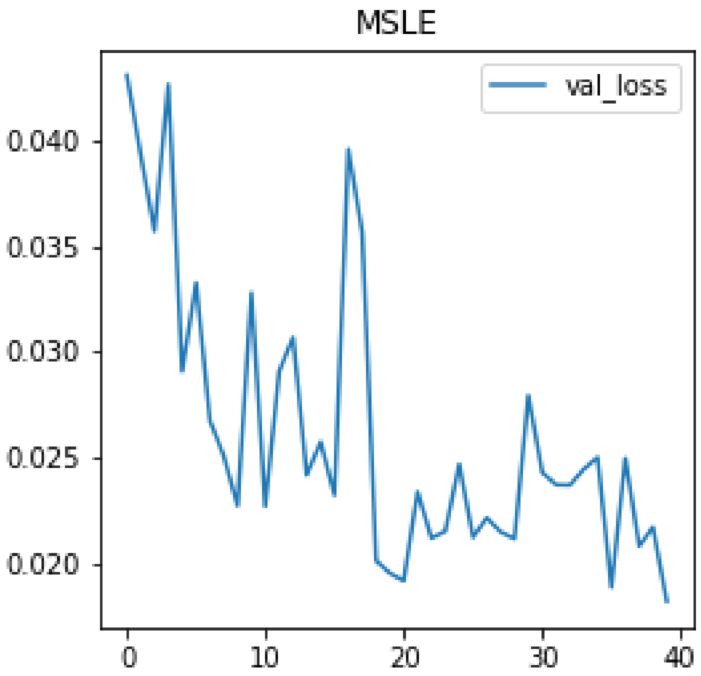
Generiranje podataka

- Podatke koje već imamo multipliciramo
- Dodajemo šumove i distorzije
- Validiramo isključivo na originalnim podatcima



Problem overfittinga

- Prilagodba broja epoha
- Dropout
- Early stopping
- Normalizacija?



- RMSE = 0.1703

Poboljšanje rezultata

- Dodavanje interakcija:
  - ako dodamo sve, opet je premalen dataset
  - biramo samo neke interakcije
  - normalizacija?
- Grid search hiperparametara

```
Model: "msle_model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_53 (InputLayer)        [(None, 299)]             0
_________________________________________________________________
dropout_66 (Dropout)         (None, 299)               0
_________________________________________________________________
dense_344 (Dense)            (None, 500)               150000
_________________________________________________________________
dense_345 (Dense)            (None, 500)               250500
_________________________________________________________________
dropout_67 (Dropout)         (None, 500)               0
_________________________________________________________________
dense_346 (Dense)            (None, 500)               250500
_________________________________________________________________
dense_347 (Dense)            (None, 300)               150300
_________________________________________________________________
dropout_68 (Dropout)         (None, 300)               0
_________________________________________________________________
dense_348 (Dense)            (None, 200)               60200
_________________________________________________________________
dense_349 (Dense)            (None, 1)                 201
=================================================================
Total params: 861,701
Trainable params: 861,701
Non-trainable params: 0
```

subm.csv 0.14017

2 hours ago by Bozidar Grgur Drmic

Interakcije, dropout, standardizacija


Hvala na pažnji!