

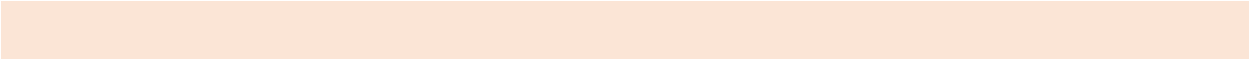


PROIECT REȚELE NEURONALE ARTIFICIALE

~ PARIS HOUSING PRICE ~

Branzea Ana-Maria

Grupa 40317A



Cuprins

1. Descrierea problemei
2. Baza de date
3. Variabila dependentă și variabilele independente
4. Prelucrarea inițială a bazei de date
5. Justificarea tipului de model. Parametrii statici
6. Experimente cu RNA
7. Codul sursă
8. Concluzii

1.Descrierea problemei

Problema predicției prețului locuințelor din Paris este una extrem de relevantă și utilă în contextul pieței imobiliare. Într-un oraș precum Paris, unde piața imobiliară poate fi complexă și fluctuantă, aproximarea prețului locuințelor pe baza unor caracteristici specifice este crucială atât pentru cumpărători, cât și pentru vânzători.

Această problemă derivă din necesitatea de a înțelege mai bine relațiile dintre caracteristicile locuințelor și prețul acestora. În general, o locuință poate fi evaluată în funcție de diverse caracteristici precum:

- Suprafața locuinței: este un factor cheie, deoarece dimensiunea unei proprietăți este adesea corelată cu prețul.
- Numărul de camere: multe camere înseamnă mai mult spațiu util.
- Zona în care este situată: locația este unul dintre cei mai importanți factori care determină prețul unei locuințe. În Paris, anumite cartiere sau apropierea față de anumite puncte de interes pot influența semnificativ prețul.
- Starea locuinței: starea generală a locuinței, renovările efectuate sau orice alte caracteristici care îmbunătățesc calitatea proprietății.

- Facilități și dotări: există anumite facilități sau dotări precum grădini private, terase, garaje care pot adăuga valoare și influența prețul.

Pornind de la aceste caracteristici, obiectivul este de a dezvolta un model predictiv care să utilizeze datele disponibile pentru a estima prețurile locuințelor. Acest model ar putea folosi tehnici de învățare automată pentru a identifica corelațiile și modelele din datele existente și apoi să prezică prețurile pentru locuințele noi sau necunoscute.

2. Baza de date

Baza de date utilizată pentru predicția prețului locuințelor din Paris provine de pe platforma Kaggle (<https://www.kaggle.com/datasets/mssmartypants/paris-housing-price-prediction>) și conține informații detaliate despre diverse caracteristici ale locuințelor, relevante pentru estimarea prețului acestora. Cu un total de 10001 de înregistrări, această bază de date conține următorii parametri:

- squareMeters- mărimea totală a locuinței în metri pătrați (cuprins între 89-99999 m²)
- numberOfRooms- numărul de camere din locuința
- hasYard- indica dacă există grădini în proprietate
- hasPool- indica prezența unei piscine în proprietate
- floors – numărul de etaje

- cityCode – codul postal
- cityPartRange- în ce parte a orasului se află
- numPrevOwners – numărul de proprietari anteriori
- made – anul construcției (între 1990-2021)
- isNewBuilt- indicator pentru locuințe noi
- hasStormProtector- indicator protecție împotriva furtunii
- basement – metrii pătrați pentru subsol (între 0-10000 m²)
- attic – metrii pătrați pentru pod(între 1-10000 m²)
- garage – suprafața garajului(între 100-1000 m²)
- hasStorageRoom- prezența camerei de depozitare
- hasGuestRoom – numărul camerelor de oaspeți
- price – prețul (între 10313.5-10006771.2 euro)

Acești parametrii oferă o perspectivă detaliată asupra caracteristicilor locuințelor și sunt esențiali pentru estimarea prețului acestora. Datele permit crearea unui model predictiv care să utilizeze relațiile și modelele identificate în datele existente pentru a estima prețul locuințelor nevândute sau necunoscute, fiind extrem de utile atât pentru cumpărători, cât și pentru vânzători în procesul de evaluare și negociere a prețului proprietăților.

3. Variabila dependentă și variabilele independente

În cadrul problemei, variabila dependentă este prețul propriu-zis al locuinței, acesta fiind elementul pe care încercăm să-l prezicem sau să-l estimăm pe baza altor caracteristici.

Variabilele independente sunt cele care sunt luate în considerare pentru a determina prețul, dar nu sunt influențate de acesta. Astfel, parametrii independenți sunt: suprafața locuinței în metri pătrați, numărul de camere, prezența grădinii și a piscinei, numărul de etaje, codul orașului, numărul proprietarilor anteriori, anul construcției, starea de nou a proprietății, măsurile de protecție împotriva furtunilor, dimensiunea subsolului, garajului și podului, precum și prezența camerelor de depozitare și a camerelor pentru oaspeți.

4. Prelucrarea inițială a bazei de date

În procesul inițial de prelucrare a bazei de date, am efectuat o analiză și am observat că în coloana care conținea numărul de camere (`numberOfRooms`) existau valori lipsă. Pentru a gestiona aceste spații goale, am creat o funcție care să înlocuiască valorile lipsă cu o medie a celorlalte valori din aceeași coloană. În plus, pentru a optimiza setul de date, am eliminat coloanele referitoare la informațiile despre numărul de

proprietari anteriori (`numPrevOwners`), deoarece acestea nu erau relevante pentru analiza noastră sau nu influențau rezultatele obținute. De asemenea, am aplicat o amestecare (`shuffle`) a datelor pentru a asigura o distribuție aleatoare.

5. Justificarea tipului de model. Parametrii statici

În procesul de dezvoltare a modelului pentru estimarea prețului locuințelor din Paris, am optat pentru utilizarea unui model de Regresie. Alegerea acestui model se datorează capacității sale de a gestiona relații complexe între multiplele caracteristici. El este ideal pentru predicția valorilor continue, precum prețurile locuințelor, pornind de la variabile independente, cum ar fi suprafața locuinței, numărul de camere, facilități adiționale și alți parametri relevanți. Prin adaptabilitatea sa, rețelele neuronale artificiale pot captura interdependențele complexe între aceste caracteristici, furnizând astfel estimări mai precise ale prețurilor.

Pentru evaluarea performanței acestui model, am ales parametrul statistic RMSE (Root Mean Squared Error). Alegerea acestui parametru se datorează faptului că RMSE este o măsură eficientă a diferenței dintre valorile prezise și valorile reale. Astfel, RMSE ne oferă o imagine clară a preciziei modelului nostru în estimarea prețurilor locuințelor. Cu cât valoarea RMSE este mai mică, cu atât modelul nostru este mai

precis în predicția prețurilor locuințelor. Utilizând RMSE ca parametru de evaluare, ne asigurăm că modelul nostru de regresie oferă predicții cât mai precise și relevante pentru piața imobiliară din Paris.

6.Experimente cu RNA

Nume Retea	Stratul Ascuns H1	Stratul Ascuns H2	Stratul Ascuns H3	Stratul Ascuns H4	Epoci	RMSE	Eroare/loss
Exp1	200	100	-	-	100	4430.4	14116231.0
Exp2	300	250	-	-	100	4440.3	13371878.0
Exp3	500	300	-	-	100	4233.0	17734860.0
Exp4	500	300	-	-	200	3795.3	13731290.0
Exp5	200	100	100	-	200	4236.1	19481040.0
Exp6	200	100	-	-	300	4494.0	14138266.0
Exp7	300	200	-	-	250	3901.3	11016656.0
Exp8	500	300	-	-	500	6264.1	44537672.0
Exp9	300	200	200	-	400	9268.9	12741140.0
Exp10	600	400	-	-	350	4356.5	19372744.0
Exp11	200	200	150	50	200	25657.4	232773808.0
Exp12	200	150	50	-	200	6138.9	27355674.0
Exp13	200	150	-	-	200	4264.9	22094732.0
Exp14	200	150	-	-	210	4517.4	11874603.0
Exp15	300	150	-	-	160	4657.6	28560608.0
Exp16	300	200	150	100	200	7181.5	16467859.0
Exp17	700	500	-	-	100	5986.2	13868438.0
Exp18	800	500	200	-	200	10046.8	19092386.0
Exp19	700	400	-	-	250	5466.8	93409008.0
Exp20	400	200	-	-	200	3722.9	15381207.0
Exp21	380	120	-	-	200	3917.1	20331416.0
Exp22	300	100	10	-	200	4232.2	26041358.0
Exp23	100	100	100	100	100	10380.6	38076368.0
Exp24	430	200			150	3996.9	9650994.0
Exp25	430	200	100	-	150	5918.9	29383940.0
Exp26	400	220	-	-	210	3429.5	13215068.0
Exp27	800	500	200	100	100	5607.7	15440734.0

Exp28	200	100	100	100	1000	8493.2	206672064.0
Exp29	1100	100	-	-	1000	3827.8	16784838.0
Exp30	3000	100	-	-	1000	4579.0	50427100.0

7.Codul sursă

```
# -*- coding: utf-8 -*-
"""
Created on Tue Nov 28 08:48:04 2023

@author: Anusk
"""

import pandas as pd
import os
import numpy as np
from sklearn import metrics
from sklearn.model_selection import KFold
from keras.models import Sequential
from keras.layers import Dense, Activation
from keras.callbacks import EarlyStopping
import matplotlib.pyplot as plt

path = "./"
filename_read = os.path.join(path, "ParisHousing.csv")
filename_write = os.path.join(path, "Housing.csv")
df = pd.read_csv(filename_read, na_values=['NA', '?'])

# Shuffle
np.random.seed(42)
df = df.reindex(np.random.permutation(df.index))
df.reset_index(inplace=True, drop=True)
#-----
# Preprocess
def missing_median(df, name):
    med = df[name].median()
    df[name] = df[name].fillna(med)
#-----

df.drop('numPrevOwners', axis=1, inplace=True)
missing_median(df, 'numberOfRooms')
```

```

dataset=df.values
x=dataset[:,0:14]
y=dataset[:,15]

# Cross-Validate
kf = KFold(2)

oos_y = []
oos_pred = []
fold = 0

for train, test in kf.split(x):
    fold+=1
    print("Fold #{}".format(fold))

    x_train = x[train]
    y_train = y[train]
    x_test = x[test]
    y_test = y[test]

    model = Sequential()
    model.add(Dense(700, input_dim=x.shape[1], activation='relu'))
    model.add(Dense(220, activation='relu'))
    #model.add(Dense(100, activation='relu'))
    #model.add(Dense(100, activation='relu'))
    #model.add(Dense(100, activation='relu'))
    model.add(Dense(1))
    model.compile(loss='mean_squared_error', optimizer='adam')

    monitor = EarlyStopping(monitor='val_loss', min_delta=1e-3, patience=5,
verbose=1, mode='auto')
    model.fit(x_train,y_train,validation_data=(x_test,y_test),callbacks=[monitor]
,verbose=1,epochs=200)

    pred = model.predict(x_test)

    oos_y.append(y_test)
    oos_pred.append(pred)

# Measure this fold's RMSE
score = np.sqrt(metrics.mean_squared_error(pred,y_test))
print("Fold score (RMSE): {}".format(score))

```

```

# Build the oos prediction list and calculate the error.
oos_y = np.concatenate(oos_y)
oos_pred = np.concatenate(oos_pred)
#-----
#grafic
plt.figure(figsize=(8, 6))

# Sortează valorile așteptate și prezise
sorted_indices_expected = np.argsort(oos_y)
sorted_indices_predicted = np.argsort(oos_pred)
sorted_expected = oos_y[sorted_indices_expected]
sorted_predicted = np.squeeze(oos_pred[sorted_indices_predicted])

# Plotare linie pentru datele așteptate și datele prezise
plt.plot(sorted_expected, label='Expected', color='blue')
plt.plot(sorted_predicted, label='Predicted', color='orange')

plt.ylabel('Output')
plt.title('Expected vs Predicted')
plt.legend()
plt.show()
#-----

score = np.sqrt(metrics.mean_squared_error(oos_pred,oos_y))
print("Final, out of sample score (RMSE): {}".format(score))
# Write the cross-validated prediction
oos_y = pd.DataFrame(oos_y)
oos_pred = pd.DataFrame(oos_pred)
oosDF = pd.concat( [df, oos_y, oos_pred],axis=1 )
oosDF.to_csv(filename_write,index=False)

```

8. Concluzii

Din analiza parametrilor statistici pentru diferitele arhitecturi ale rețelei noastre , putem trage următoarele concluzii pentru alegerea celei mai potrivite configurații:

- » se observă că o arhitectură mai complexă nu aduce întotdeauna îmbunătățiri semnificative în performanța modelului. De exemplu, experimentele 8, 11, 18, 23, 28 și 30, care au avut un număr mare de straturi și noduri ascunse, au prezentat o creștere semnificativă a erorii, ceea ce sugerează că o arhitectură prea complexă poate duce la overfitting.
- » Experimentele cu un număr mai mic de straturi și noduri, precum Exp1, Exp2, Exp3, prezintă rezultate bune în ceea ce privește eroarea RMSE. Totuși, există o limită în performanță odată ce se ajunge la un anumit număr de noduri.
- » Exp4, Exp7, Exp20, Exp21, **Exp26** și Exp29 prezintă performanțe decente în ceea ce privește RMSE. Aceste experimente sunt variantele cu o configurație echilibrată a straturilor.

Putem concluziona că, în general, configurațiile cu un număr moderat de straturi și noduri (nu prea simpliste, dar nici prea complexe) par să ofere rezultate mai bune pentru estimarea prețurilor. Alegerea unei rețele cu prea multe straturi sau noduri poate duce la o performanță mai slabă pe datele de testare din cauza overfitting-ului.