IE 7275: Data Mining in Engineering

Housing Price Prediction

Milestone: Project Report

Group11

Gaddam Sreeramulu Rohit Kumar

Mobile Student : (857)-265-0719

gaddamsreeramulu.r@northeastern.edu

Signature of Student : Rohit Kumar

Submission Date: 04-28-2023

Title of the Project: House Pricing Prediction

Problem Setting

The concept of house pricing is the process of determining the value of a house based on various factors such as location, size, condition, amenities, and market demand. House pricing is a critical component of the real estate industry and is used to set the asking price for a house when it is put up for sale.

Challenges:

1. Higher interest rates have stopped buyers in their tracks.
2. House inventory has gotten low.
3. Getting the right price for the house is tough.

Application:

There are several applications of house pricing models in the real world, including:

Real estate valuation: House pricing models can be used to estimate the value of a property for sales or rental purposes.

Homeownership: Homeowners can use house pricing models to understand the market value of their property, to make decisions about when to sell or make improvements, and to plan for their financial future.

Urban planning and development: Urban planners and developers can use house pricing models to make informed decisions about land use, zoning, and investment opportunities.

Problem Definition

The problem of predicting the sale price of a house based on the various features like size of the house, number of bedrooms, location, etc., can help real estate agents and homeowners in pricing their houses correctly, and can also assist potential buyers in determining a fair price for a house they are interested in. The target variable (price) can be influenced by these factors and the goal is to build a model that accurately predicts the price of a house based on these factors.

Data Source

In this project for house pricing prediction, we will be using data set from 'Kaggle', which is a Google LLC subsidiary, where users can search data sets and build models in a web-based data science environment.

Here's a snippet of our dataset after performing the data cleaning and preprocessing, where we removed null and duplicate values and dropped unnecessary columns.

Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Address
79545.45857431680	5.682861321615590	7.009188142792240	4.09	23086.800502686500	1059033.5578701200	208 Michael Ferry Apt. 674 Laurabury, NE 37010-5101
79248.64245482570	6.0028998082752400	6.730821019094920	3.09	40173.07217364480	1505890.91484695	188 Johnson Views Suite 079 Lake Kathleen, CA 48958
61287.067178656800	5.865889840310000	8.512727430375100	5.13	36882.15939970460	1058987.9878760800	9127 Elizabeth Stravenue Danieltown, WI 06482-3489
63345.24004622800	7.1882360945186400	5.586728664827650	3.26	34310.24283090710	1260616.8066294500	USS Barnett FPO AP 44820
59982.197225708000	5.040554523106280	7.839387785120490	4.23	26354.109472103100	630943.4893385400	USNS Raymond FPO AE 09386
80175.7541594853	4.9884077575337100	6.104512439428880	4.04	26748.428424689700	1068138.0743935300	06039 Jennifer Islands Apt. 443 Tracyport, KS 16077
64698.46342788770	6.025335906887150	8.147759585023430	3.41	60828.24908540720	1502055.8173744100	4759 Daniel Shoals Suite 442 Nguyenburgh, CO 20247
78394.33927753090	6.9897797477182800	6.620477995185030	2.42	36516.358972493800	1573936.5644777200	972 Joyce Viaduct Lake William, TN 17778-6483
59927.66081334960	5.36212556960358	6.3931209805509000	2.3	29387.39600281590	798869.5328331630	USS Gilbert FPO AA 20957
81885.92718409570	4.423671789897880	8.167688003472350	6.1	40149.96574921340	1545154.8126419600	Unit 9446 Box 0958 DPO AE 97025
80527.47208292290	8.09351268063935	5.042746799645980	4.1	47224.35984022190	1707045.722158060	6368 John Motorway Suite 700 Janetbury, NM 26854
50593.69549704280	4.496512793097040	7.467627404008020	4.49	34343.991885578800	663732.3968963270	911 Castillo Park Apt. 717 Davisborough, PW 78603
39033.809236982400	7.671755372854430	7.250029317273500	3.1	39220.36146737250	1042814.0978200900	209 Natasha Stream Suite 961 Huffmanland, NE 52457
73163.6634410467	6.919534825456560	5.9931879009455700	2.27	32326.123139488100	1291331.5184858200	829 Welch Track Apt. 992 North John, AR 26532-5136
69391.3801843616	5.344776176735730	8.406417714534250	4.37	35521.294033173200	1402818.2101658500	PSC 5330, Box 4420 APO AP 08302
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Data Description

The dataset contains features like sales price, neighborhood, number of bedrooms, number kitchens, locality, sale type, sale condition and so on.

The target variable (price) can be influenced by these factors and the goal is to build a model that accurately predicts the price of a house based on these factors.

We will be using the USA_Housing dataset, which contains information about houses. The objective of this task is to predict the price of a house, which is a continuous variable, making this a regression problem. The dataset includes several columns such as the average income of residents in the city where the house is located, the average age and number of rooms and bedrooms in houses in the same city, the population of the city, the selling price of the house, and the house's address.

Below you can see the heatmaps of our dataset.


Avg. Area Income
Avg. Area House Age
Avg. Area Number of Rooms
Avg. Area Number of Bedrooms
Area Population
Price
0.0
0.2
0.4
0.6
0.8
1.0

Model Exploration and Model Selection

Here, we will try to explore various models and to find the model which has proven to be the most efficient one.

We will explore around with several models before we make the selection.

There are two types of models:

- Predictive Models
- Descriptive Models

Predictive models functions have the order as Classification, Regression, Time Series Analysis, Prediction.

Descriptive models process in the order as Clustering, Summarization, Associative Rules, Sequence.

There are several models such as logistic regression, SVM, KNN, and so on...

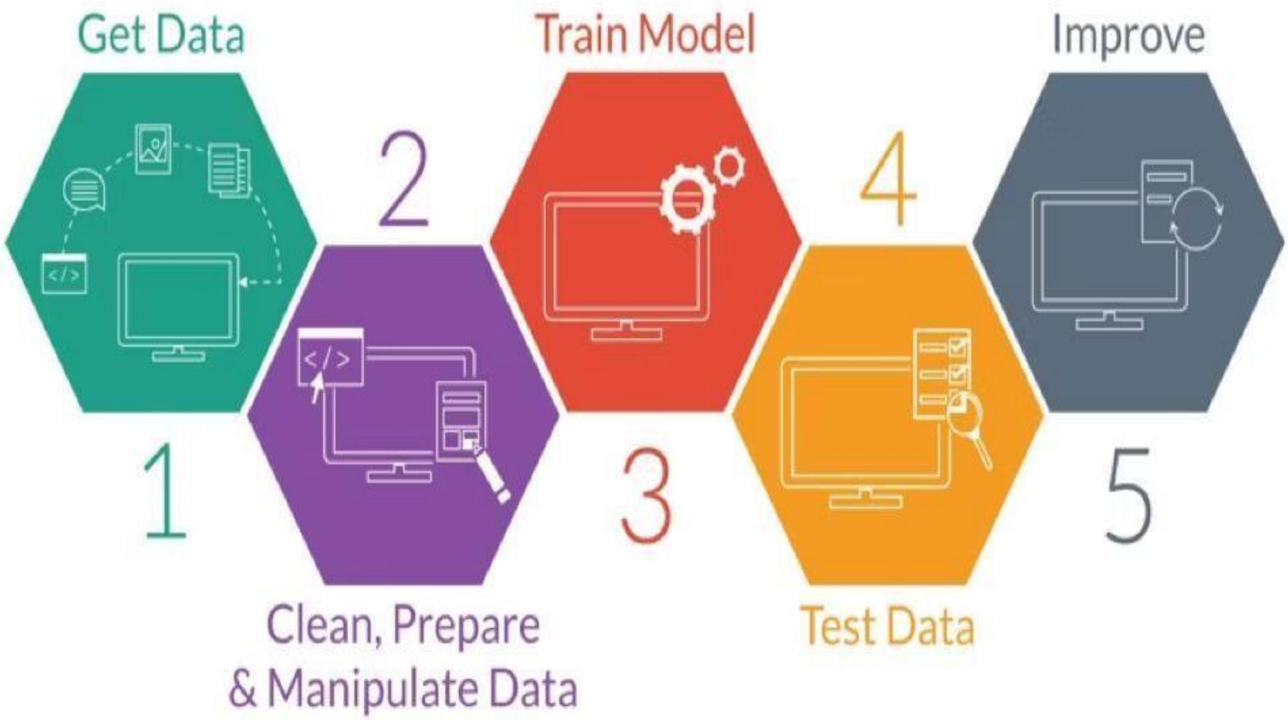
We will make the decision of the model selection by using some of the points as below:

- a model that satisfies the demands and limitations of project stakeholders.
- a model that, given the time and resources at hand, is suitably skilled.
- a model as that is skillful as compared to naive models.
- a model that performs well compared to other models that have been examined.
- a model that is proficiently relative to the state-of-the-art.

In a perfect world, the data would be divided into training, validation, and test sets. Candidate models would then be fitted on the training set, evaluated, and chosen on the validation set, and the performance of the chosen model would be reported on the test set.

Process Flowchart

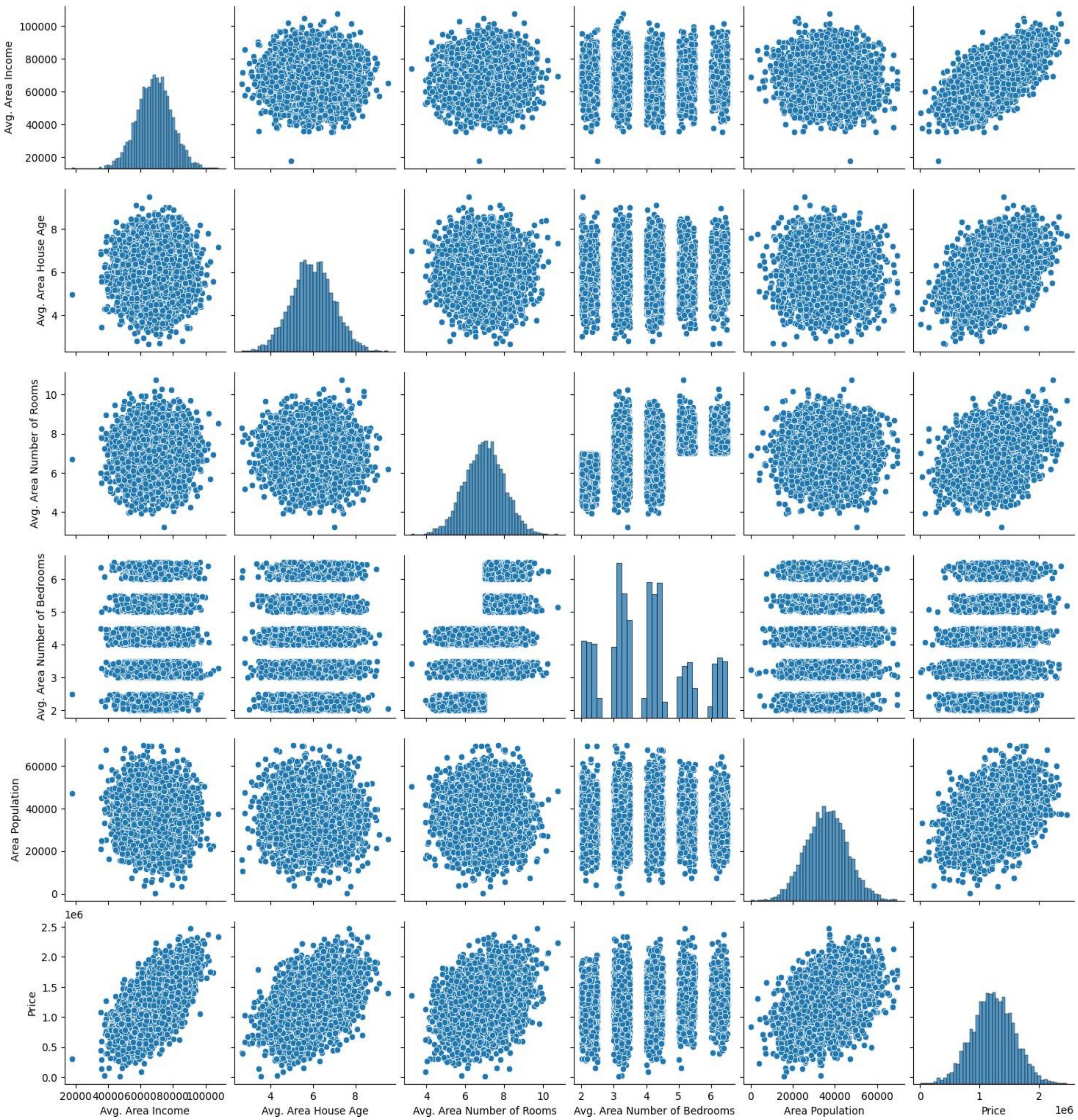
Our project process involves several steps that are essential for extracting useful insights from a large dataset. Firstly, data needs to be collected and curated from various sources. Once the data is collected, it needs to be cleaned and manipulated to remove any errors, missing values, or inconsistencies. After that, machine learning models are trained using the pre-processed data. The model is then tested on a separate set of data to evaluate its performance. Based on the results of the testing, the model can be refined and improved to enhance its accuracy and efficiency. The iterative process of improving the model can be repeated until the desired level of performance is achieved. Overall, this process is an essential step towards unlocking the value of big data and gaining insights that can drive business decisions.



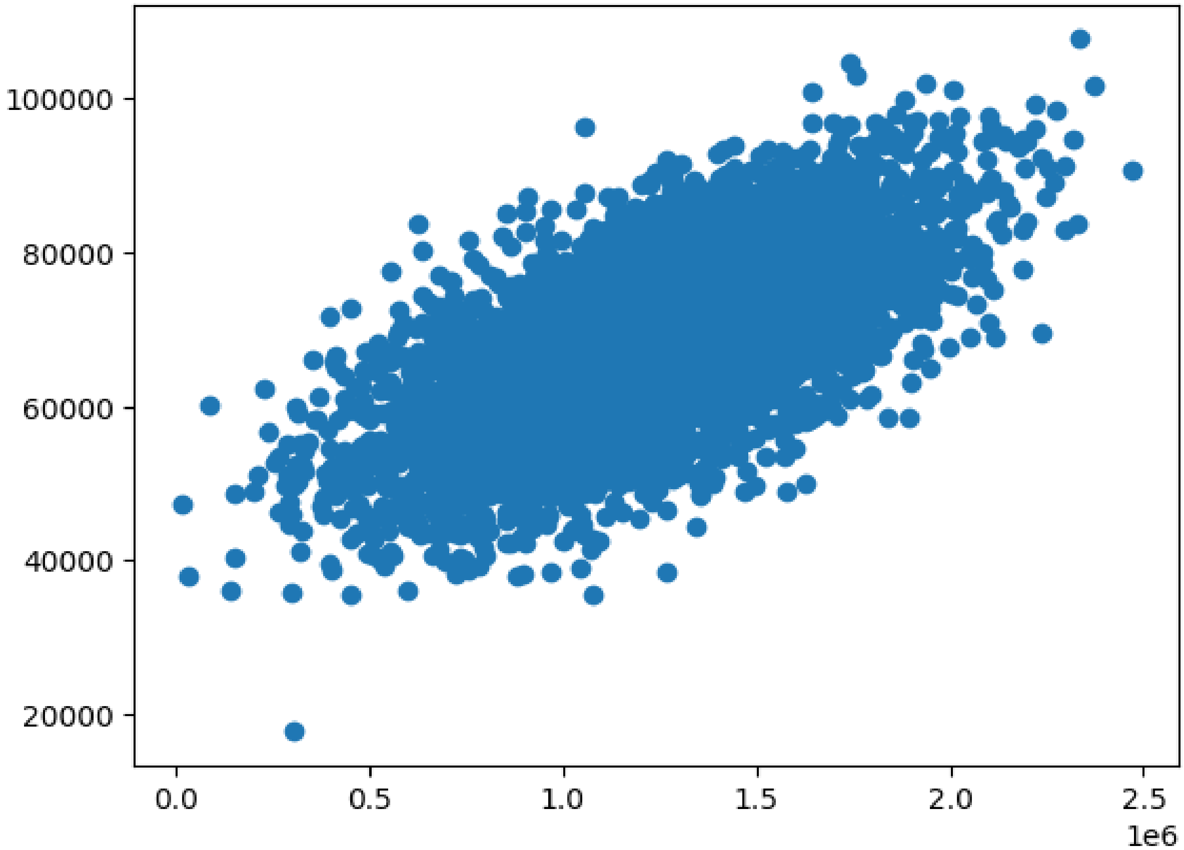
Project Process Flowchart

Data Exploration and Visualization

From our dataset, we have plotted a Pair plot each column in our dataset as seen below such as Avg. Area income, Area population, Avg. House Age, and so on.

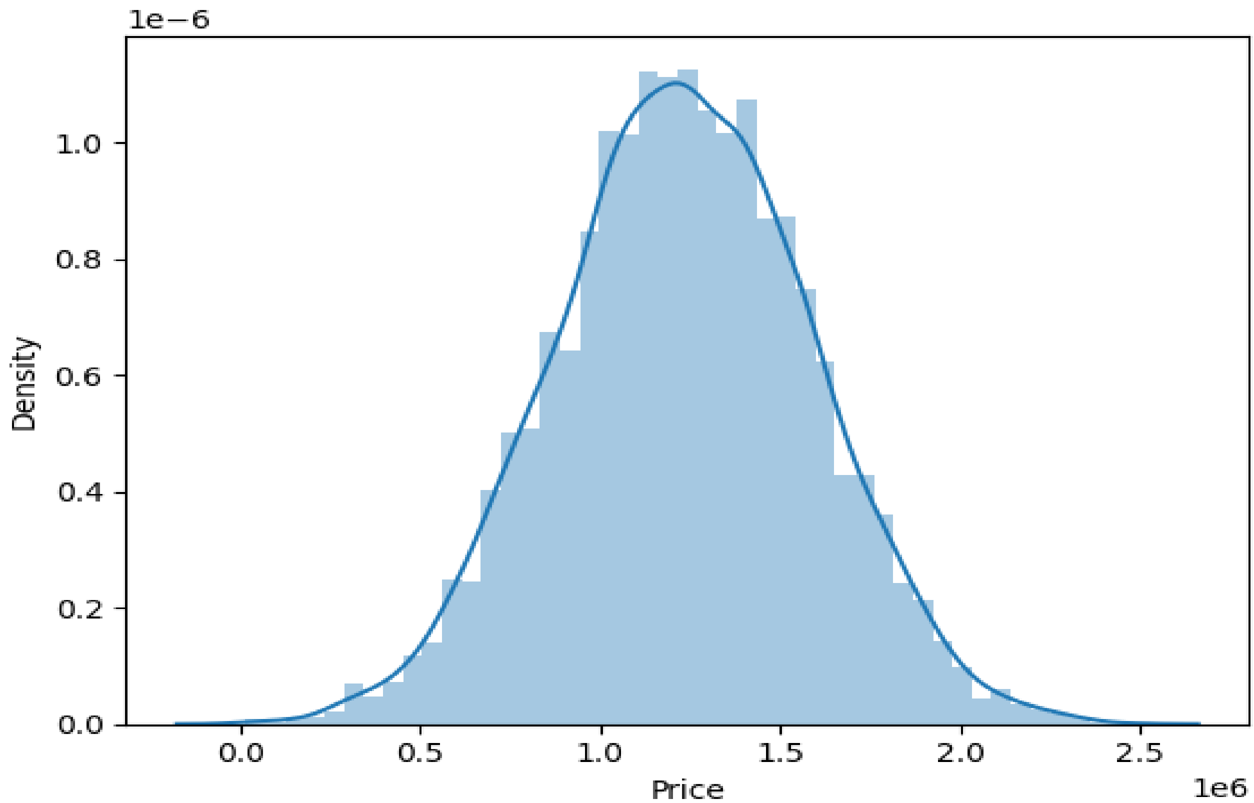


Here we plotted a Scatterplot for average area income.



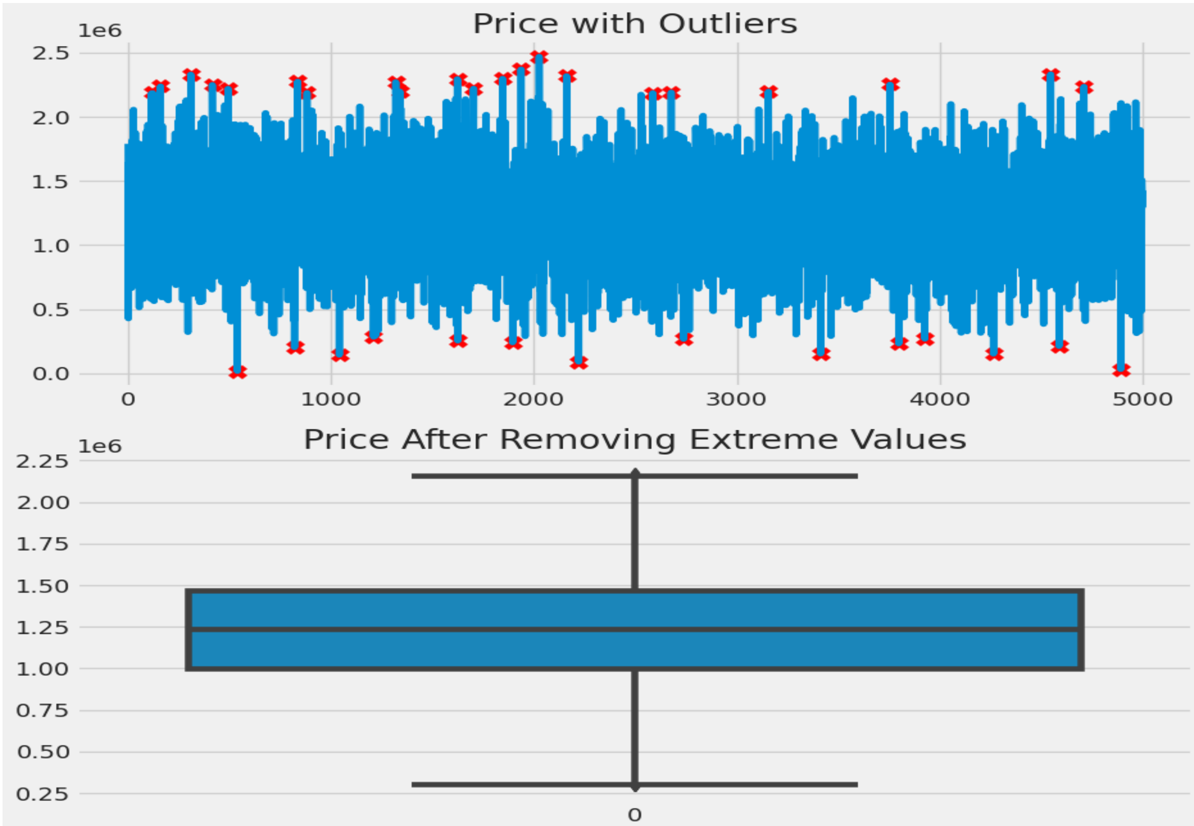
Average Area Income

Distant plot for Price and Density in our dataset.



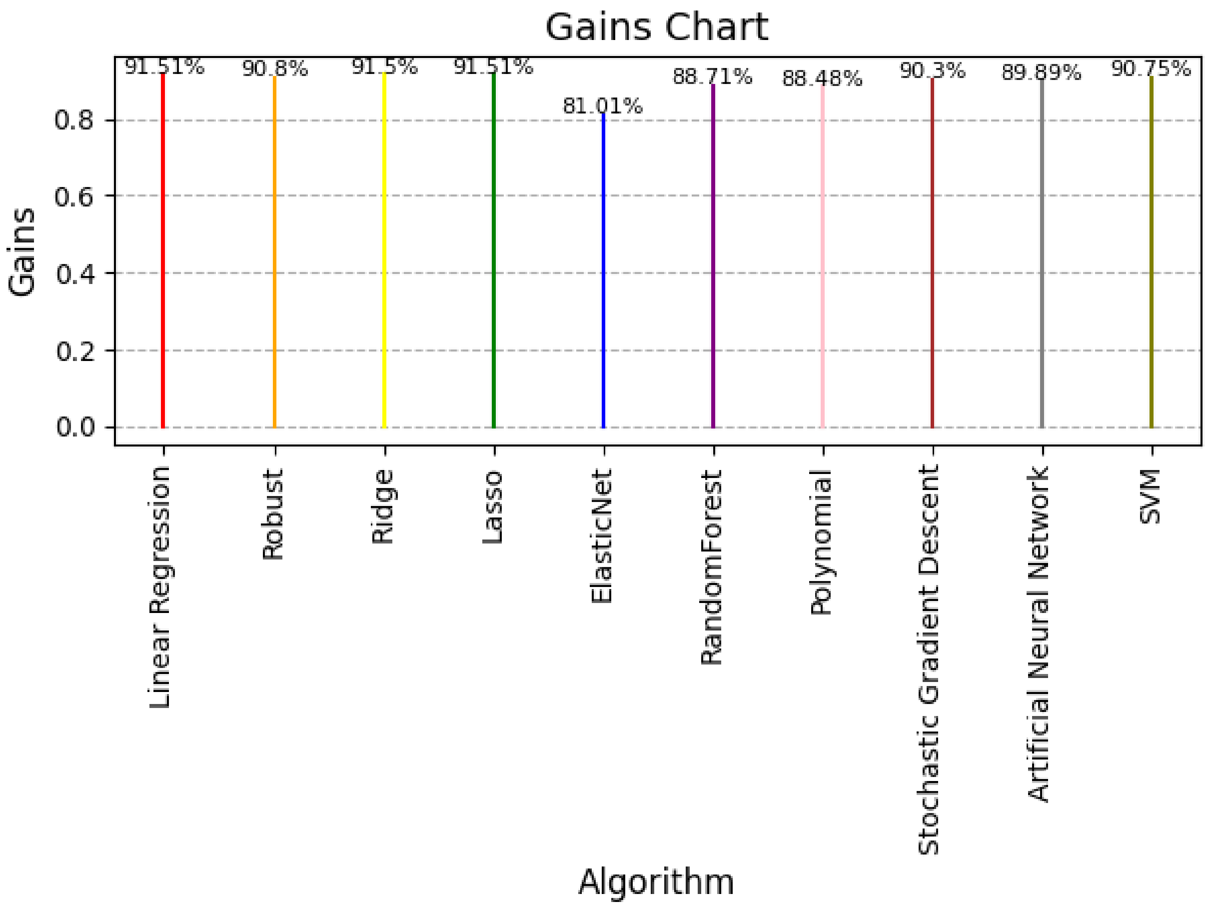
Outlier Analysis

Below we have performed Outlier Analysis in our dataset based on different columns.


Price with Outliers
1e6
2.5
2.0
1.5
1.0
0.5
0.0
0
1000
2000
3000
4000
5000
Price After Removing Extreme Values
1e6
2.25
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0

Gains Chart

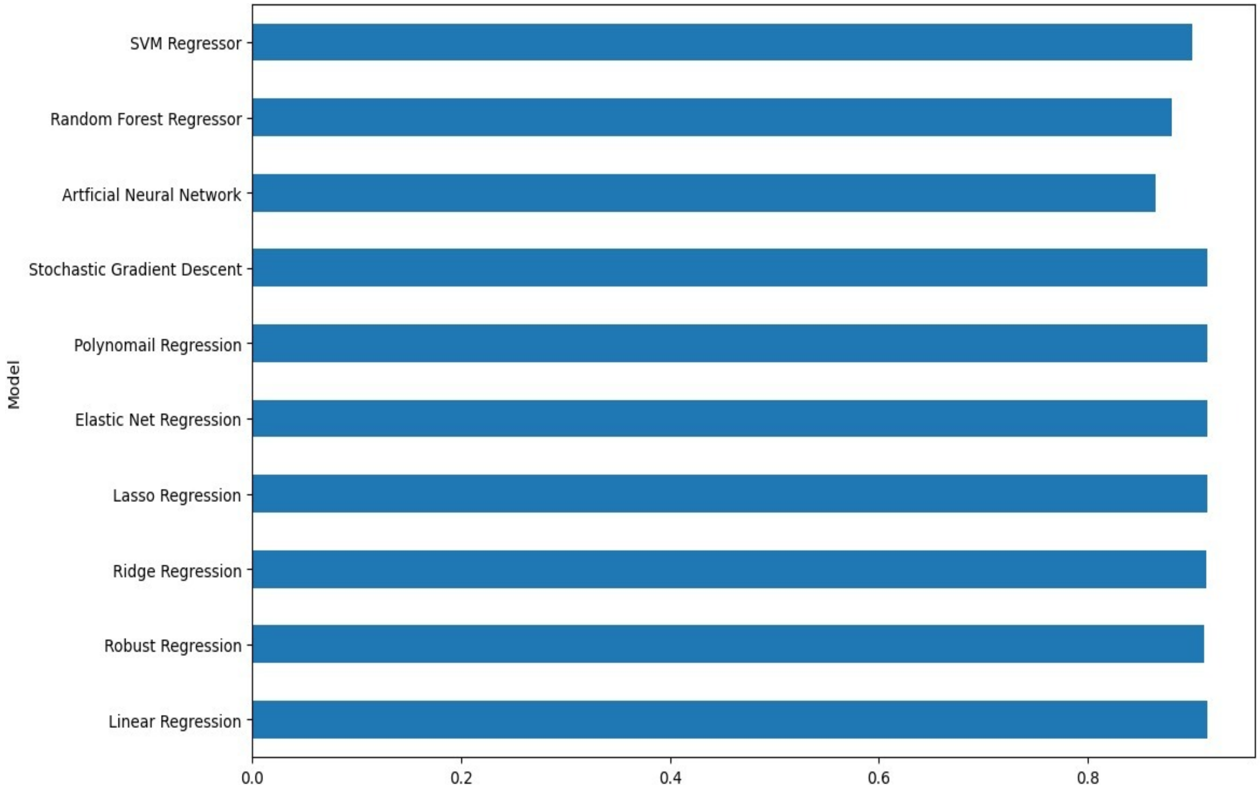
Below we have a gains chart with dotted lines to see the accuracy of all the models that we performed our analysis on.



Results

Model	MAE	MSE	RMSE	R2 Square	Cross Validation
Linear Regression	81135.566093	1.006842e+10	100341.529545	0.914682	0.917379
Robust Regression	84841.957533	1.109778e+10	105345.989279	0.905959	0.911262
Ridge Regression	81428.648355	1.015327e+10	100763.435337	0.913963	0.917379
Lasso Regression	81135.698517	1.006845e+10	100341.683215	0.914682	0.917379
Elastic Net Regression	81184.431473	1.007805e+10	100389.492321	0.914600	0.879545
Polynomail Regression	81174.518441	1.008198e+10	100409.083243	0.914567	0.000000
Stochastic Gradient Descent	81135.565670	1.006842e+10	100341.528834	0.914682	0.000000
Artficial Neural Network	599600.081205	4.920804e+11	701484.430402	-3.169808	0.000000
Random Forest	94365.592485	1.419648e+10	119148.981672	0.879701	0.000000
SVM Regressor	87205.730510	1.172093e+10	108263.256765	0.900679	0.000000

Models Comparision



Project Outcomes & Recommendations

Project Outcomes

In our project, numerous machine learning algorithms were evaluated for their effectiveness in carrying out a binary classification task. The dataset used for training was normalized, and it was determined that the Linear Regression achieved the best results.

Recommendations

1. To enhance the AI-based system's analytical abilities, natural language processing (NLP) capabilities can be incorporated. At present, the system mainly examines structured data like housing data, but NLP could help analyze unstructured data like customer reviews, leading to the identification of better houses at recommended prices.
2. By utilizing blockchain technology, a secure and transparent system can be established to store and exchange housing data, making real-time feedback implementation feasible.
3. The system can be expanded to other housing search applications by scaling it up.

Appendix: Python Code for use case study

```
Ipip ioetall -g dvgVot
```

- Import Libraries

```
'impom matplotlib.pyplot ae pIt
```

```
miatpJotlib iodine

f  eve.eet_styIe('whitegrid')
f gIi.etyie:use('fivetAirtyeight"}
```

• Check out the Data

```
USAboueing - pd.read_csv('/content/u°A_Bousing.ceo'
'USAboueing.bead()
```

	ny . area In rose	Avg . Rrea Bouse Sge	Dwg. sea or oz Bourns	Avg. area NuaBer o* eeacz cms	Azea Population	
0	79545.458574	5.682861	7.0061 BB	4.09	23086.800503	1.059034e+06
1	79248.642455	6.002900	0.730BZT	3.OB	40a 7 T 072174	T . ñ0b8B1 e +O0
2	012B7.0071 70	5.B00B00	B.5127Z7	?':13	308BZ. 1 b9400	T.0@B9 +DB
3	63345.240046	7.1B8356	5.586729	3.26	34310.242831	1.260617e+06
4	?'0BBZ. T87ZZ0	5.040555	7.B393B8	4.23	26354.109472	6.309435e+05

```
USAboueing.info()
```

```
‹cTaee 'paodae.core.tram.DataFrame'›
Baogelndew: a0ns entries, 0 to 4999
Dara eoluene (:otaI - coluoms›:
 #   rolumm                        Non-Null Count  Dtype
                                                   -----
 0   Avg. Lea•income               a0as oon-ouJl   float64
 i   Avg. sea Bouse Age            s0aO not-ouzl   float64
 ?   Avg. •Vea bumber• of Boome    Saad noo-oull   float64
 z   Avg. sea Pmsber of Beéroors   asaa now-ouIl   float64
     Area Po iation                aa0s oon-ou2l   Iboa-6
 a   Price                         a0aa now-ouJl   float64
 6   Addreee                       aa0s oon-ou2l   ob§eM
dtypes: float64(6), object(1)
memory usage: 273.6+ KB
```

```
USWoueing.describe }
```

	Avg. Area Income	Avg. Area Bouse Age	Avg. Area Bumber of Rooms	iv@. area Nuaber o* Bedrooms	Area Population	
count	5000.0JX+0£t0	5000.000000	5000.000000	TOO ODOOOO	5000.000000	5.000000e+03
meen	f1B5BZIO8BB#	5.977222	B.88T?B2	3.OBE Bg0	30T 63.51B05B	I .2Z3073a+OB
std	100b7.9B1214	0.991456	1.005833	1.Z341S7	9BB5.850114	3.fi31178e+0b
min	1T7B6.0g t1 BO	2.644304	3W01B4	2 DOOOOO	17Z B1OOBB	1 .fiB38B8 e +04
25%	B1480.5023B8	5.3ZZ283	6.299250	3.1é0000	39403.8Z8702	8.B7?'771e+05

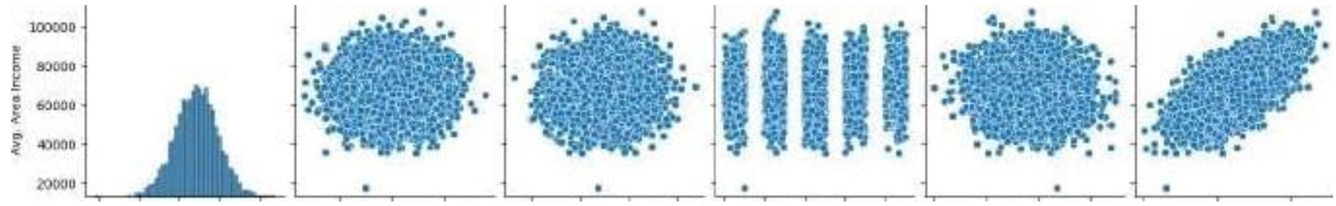
```
USAboueing.coIrmoe
```

```
Index{{'Avg. Area Income', 'Avg. Area Rouse Age', 'Avg. Area Muober of Roome',
      'avg. sea mumber of Sedronme', 'Area Pogulatioo', 'Price', 'Address'],
      dtype-'object')
```

Exploratory Data Analysis (EOA)

```
sns.pairplot(vSAhousing)
```

eea5orn: egri0.PqifsriD:at 0p7IaCJéf##YIDi




```
ipi’g ioetall éeplot
```

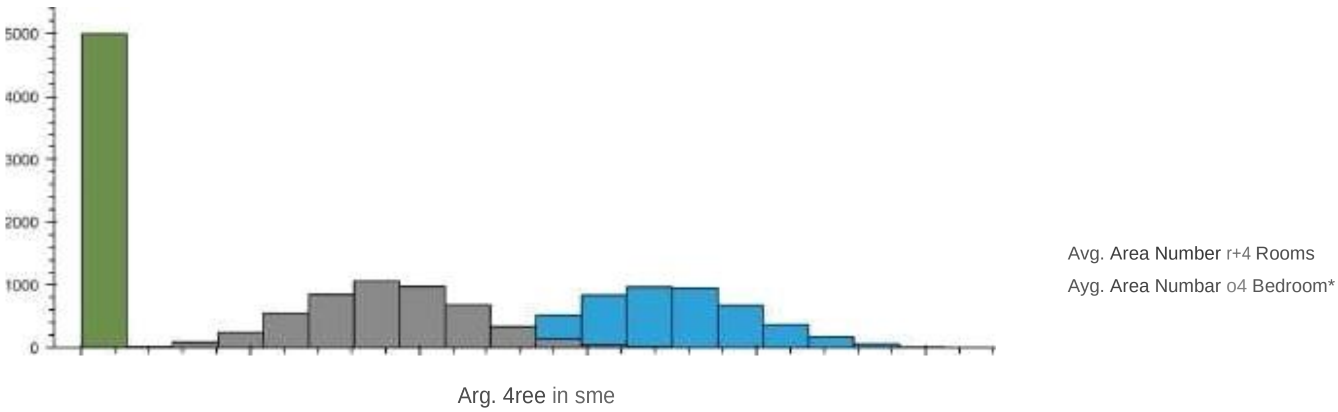
```
Looking in iodeieg: bttpe•//gypi•mg/eimplg, Atips•{/us-python pig dev/colab-wbeeJs{pubJic/eimpLs
Requirement already aatiefiea:•bvplét in /Car/local{lrb/ honJ.v/died-pacxagee }a.#:s)
Bequireméot already aatiefied:•pac ging io /uer/locaV/lib/pytñoos.9/dist-pac gee jIrom.hspiot) (z3:i)
maquiremeot already zatiefied: ouxpy>-i.ñ ’io /uer{lqcdl/1ib/python3.9/dist-pactagex (fzom ñwplotj (l.:z.4)
Bsquiremeot alreddy zñtiefied. colorcet -z io /uer/local/J /pytbénz. /diât-paciagea (IN hvplor) (3.a.i)
stir «ml ameaa »aci^ri»a: toxic •,i.ii.a’in/ r’/iooai/is0/Pmu»na.s/ai^t-Pan 4 e ‹ r,oma root fi.is.a)
Bsquiremeot alreddy zñtiefiedi eJ ›-a:iI.D io /usr•/IocaJ/iib{g/âond.4{diei-pâctages (from bvplot) #4.I•.A#
aequiremeot already satisfied:•boxeh ›-i.A.D•’io /uer/loca1/lib/g#bona.&/diet-pactagee (from hvplof) (?.•.##
ant a eaa,ea‹ i^zied: a, in /o’r/local;2u›/ taoni.e/ai t-Mac ge, lrror a rot ‹ i.s.zi
Bequiremeot alr’eady aatiefied:•parâm›ii.s:D•io /ñer{Ioca1/lib{g/bood.9/dizt-paciagee•(from’hvplot) (?.IA.D#
maq iremeot already zatiefiedz.gilt-7.i.D io /uar/Ioca1/lib{p/bond.s{diri-paqiager (from bokeb>-i.a.é- bvplot
miquirement already aatiefied••A io9 exfew s9-?.ID.A in’ iusr/iwa17lib/g bond.4/éiet-gac aqñe (from boreé ›-i’
Bequiremeot alrgady eatiefied: tornado ›-s.i in /uer/local/lib{p/boo3. {dirt=pactager:(from boreb ›-1.a.O- ›bvplot)
Requirement already aatiefied:•PyyI ›•J.ié io /uer{léch1/lib/p/bomd.9/dizt-pactagea (from boreb -1.0.#->beplot]
s irm‹m’» am»ad, a‹’i,ri»a:.?kojaz•-z..’’io/»^r/inoai/iib/sy‹mo3.:7di^-ear aga^(ir‹m trem-i.c’.a-• im’)
Requirement already aatiefiea:•gyin ›-a.•.4 in /uer/Voñal/lib{gyt6oo3:9/dist-pactagea #Irom coloicét>-z*›6vpldzj
Bequireméot already aatiefied:•’gywis-oomom ›-o:7.4 ’in’ /uei/local/lib/ bonsi9/éiei-packages from bol ieme ›-i:u
maq iremeot already zatiefiedz.g hpn-datertiN-4.8:i io /uer{local/lib7pythons.9/di’et-pactagea (fréepandas- bop:
Bsquiremeot alreddy zñtiefied. p/ -4o20.i io /uer{Iéca1/lib{p/hdo .S?dizt-pattagez (fret paiflâs- hvplot) (?022’
B#quiremeot alrgady eatiefied: retuptoois ›-4z in /usr/IécaI/lib/g/bon3.9/died-gacraqea (from•paneU-a„iI.D- dvpi ›
Bsquiremeot alreddy zñtiefiedi car 0wn in /uer/Vocal/lib/ations.g/diet-packâgee {from panel ›-D.ii.0ññvpIo’t) a’
Bequiremeot already aatiefied••bVeañh in /usr/i al{Iib/pythooJ.v/diet-packages ;from’paoel ›-0:ii.0- hvglot) (s.a
B#quiiement alieady eatiefied: t -d„4#.0 io’ /usr{2oca1/iib{p/hood.¥/dizt-pactager (from pdnel ›-0.ii.s- ›bepiot
Bequiremeot alr’eady aatiefied:•:iequeete in /uer/local/lib/pythons.9/died pac ’gee {from’’panel ›-D.11.0 ›bvpioA).\T
mo‹ am eaar»a‹ i ^€iea::m xueb%ia•-:..o i^ /v /Aai/nb/g bona. /di,i-pack'‹m, (rr‹ ›mnnjat•-2.-•meb:
s irm‹mo‹ am»aay »a‹i,zied..»i -i.s in/u,rJl’ooai/in/ations.g/ai^c-Pao 4», ‹ *wn hon-‹eu*il•-z.’.i-apr
B#quiremeot alrgady eatiefied: beocodinge:in /uer/locaT/lib?p/bood. {dirt=pactager:(from bleacb- ›panelr e.li.0•
Requirement already aatiefied:•ieportlib-ñétadata ›-4.4 iâ /usr{loc#l/lib{@/éooa• /diat-pactager (from marxddvo- ›¿
aequiremdot Already saCiefied:•vzVlibJ‹ I.2?„ ›-i.c1.i io /uer/local/lii/pytbon3.9/dist-packages (from regoertzñ.pa
Bñqu nf aliéady »atieflea: •.châreet-nor’oa1iser-•i.n.0 io /o’ar/local/lib/g bo’na.s/diet-pacrager (mhm requeeze•
miquirement already aatiefied••idoa 4} ›-z.a •io /usr{Iocal/lib/g/honu.9/diét-pactager from regreeie-panel -0.ii’
maquiremeot already zatiefiedz cemifi ›-2ei7. „i? in '/rer/locaV/lib/pytAop3:é/diet-p kagee ]from. requeztx- paoel:
s m‹mo‹ am»aay »a‹i,ziea..i •-c’.s in /u,r/iooai/z ’7e,tbno,.s70iac-gas», f**nm me»rciit-wtaaa»-’.’,--
```

```
pig ingtall boLoei’ggs
```

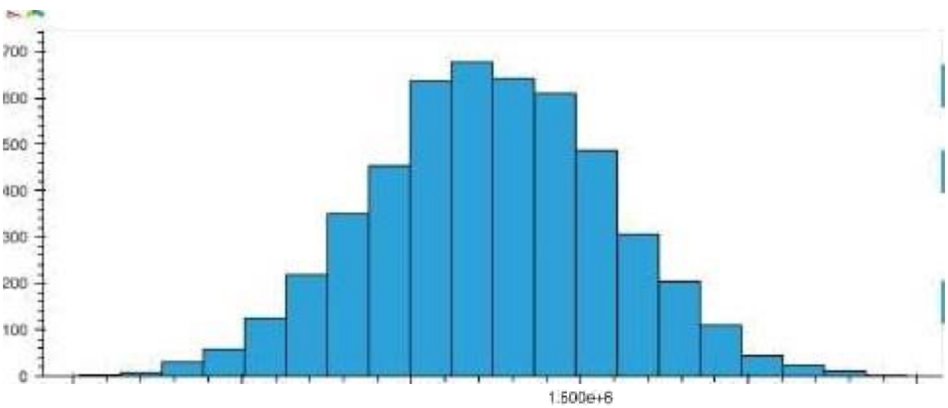
```
Looking iz.indexesz btt i , bitpz•//ua-python.pkgrdew/Colab-wbeeIs/pub1ic?simgle/
ñemeo‹ aUeadr aa‹ *^fiea: boloei ié /uer local/iib/pytbooa.9 digt-paoxagee {1.is„»)
Bsquiremeot alreddy zñtiefiedi jsckâging i’o /uer?iocaI/lib/pytboos.9/disi- agea {from hoiovieee) pi3.i}
o ire«mo ahead aaci^fied: oomcet in./u,r7iooai/in/g,t one.s./dist-outage,.,*roml iv ieea) (z.o. }
x irm‹mo‹ aueaay aa‹i,ziea. psi -com,m•-o.z.a in’/»,r/iocai/iim/ t>o»>.v/ai^‹-elkage^ ‹ *re roiwi a› (z.z
Bequiremeot already aatiefied:•.paoe1 ›-&.i3„i io /usr/local/lib/pyibon3:»/died-pactagee (from boloviewe} {s:i4:. )
B#quiiement alieady gatiefied: é -i.0 io /uer/lo’caI/1 /pytbooz.9/dist-#c2agea (Irtm• holoviewe) li.z?.4)
Bequiremeot alr’eady aatiefied:•.param ?:e, ›-i.9:3 in ’/uer/1ocaV/Iib/pythons.v/diet-gackagee |froe.hoJovisem) (i:i3
maquiremeot already zatiefied:.pandae ›-4.:é.:0 in boar/local/lib/ bon3>s{diet-gackapeg {from holovi ) {i.S.•3
Bsquiremeot alreddy zñtiefiedi p/bon-dâteñti1 ›-2.8.i io /ier/iécal/iib/pytbpn3%s/di’et-phciâges (fmm pandat ›-0.
aequiremeot already satisfied:•p/•s ›-?&iO.i ’io /uer/loca1/.iib/g#bon3.&/diet-pactagee (from pandas ›-0:?s.s- ›boiov.
Requirement already aatiefied:•oar ñwn io’/uer/local/lib/pytñgza.9/diet pac%gee {froe panels-D.i3.i- ›boloviepe)
aequiremdot Already saCiefied:•bIeach in {uar/Jocai/lib/pjthof3.s/died-pacxagee \froe gsie’l ›iO.is.i- ›ho’loviewe) '
nf añéad, »a‹ i^ lea: ’aéiupCoois ›- z in./osr/I ai/iib/ bon3:9/digt-paenaqee (from’paie1»a:ii.i- ›doio
miquirement aVr’eady aatiefied••’tqda ›-a:•#„D.•io /usr{Iocal/lib7g/hon•3.9{diét-pactager” from panel ›-0.i3.i- ›hñlovi
maquiremeot already zatiefied: debt?.s.é, ›-Y.a.0’’ig /uer7locaI/Vib/pptbons.9/diet-pacxageg {from panel ›-D.i3:i-’
Bsquiremeot alreddy zñtiefiedi typiog-exteneidos in /zrr/local{lib/pythons.s/died-pacragee jfroe gaoel ›-O.1a.i- ›h
B#quiremeot alrgady eatiefied:.requegte'in• /uer/local/lib/pytbogs.9/diet-pacx°4°° f*'° panei9-D.13.i-;Foioviewe)
Requirement already aatiefiea:•gyin ›-0.•.4 io /uer/Voñal/lib{gyt6oo3:9/dist-pactagea #Irom pangl ›•0.i3:i->âoIoei%
Bequireméot already aatiefied:•’P I ›-).iD•io /ñer{Ioca1/lib{g/boo3.9/dizt-paciagee•(from boxeb<?.sr0, ›-i.4:.0- ›¿
B#quiiement alieady eatiefiea: t ado ›-3.i io /usr{IocaJ/lib{g/hood.9/dizt-packages (from bdrebiz.S.0, ›-i.a.0w
miquirement already aatiefied••jiVlom ›-7.i.D io /isr/Ihca1/lib/pytbona.s/daai-pac/ager (frñm boxeb‹?.a.s; ›-:..0-
maqiiremeot already zat}efied:.J}oja? ›-’:.s’io /ugr{lqcal/1ib/python3.9/dist-packages (fzom boreb J.a.o; ›-2.4.4- ›pt
o irm‹mo‹ ameady aa‹i visa. air-i.c in /ur/l’ocai i / tnonz.g/area-sac ge ‹ *wn hon-‹euiil•-z.’a.i-ver
aequiremeot already satisfied:•webeocodinga ’io /uer/loca1/.lib/g#bon3.&/diet-pactagee (from ble b- ›panel ›-D.13.J•
Requirement already aatiefiea:•ieportlib-ñâtadata ›-4.4 iâ /usr{local/lib{p/boo3• /digt-pactager (from mar vo- ›¿
Bequireméot already aatiefied:•rr11ib3‹ i.i?; ›-i.?i.i io /uer/local/Iib/gytbooz.9/dist-gackagee (from•requeetañpa
```

```
requirement already satisfied: idna‹u,›-i.s in ?uer/locaJ/lib g/bond.9 diet-pacaagee (from regreeie-›ganeI•-0.id

requirement already satisfied: cbareet-noreaIizer--°.n.n in /usr/local/lib/p)vdona. /die:-gacrages (from requeete•
Bsquirement already eataefied: ceñifi›-?0i?. .17 in /rer/local/lib/gytZons.9/diet-pacxagee bros requests- paoel:
requirement already satisfied: sippo-a.S iz /uer/loc*I/Jib/pytboos.9ldist-pacxagee‹from iigortlib-metadata=-#.•-:
Bsquirement already eataefied: :mrkug&aIe›-z.0 in /uer/Iocal/lib/p hone.»/died-pacragee (from Jizja?›-?. -oleh*
```

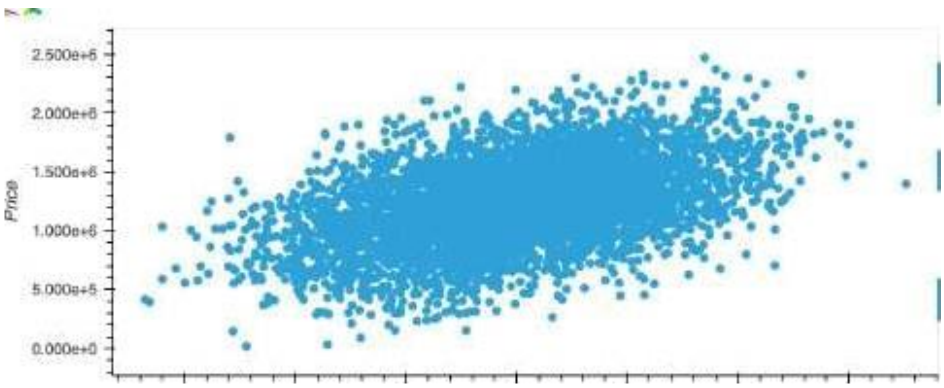
```
import holovieee ae hv
ñv.emeneion('bekeh'}
UsVoueing.hvoIoz.hist{by-’Prire', srbplots-Falee, ridtb-i000)
```



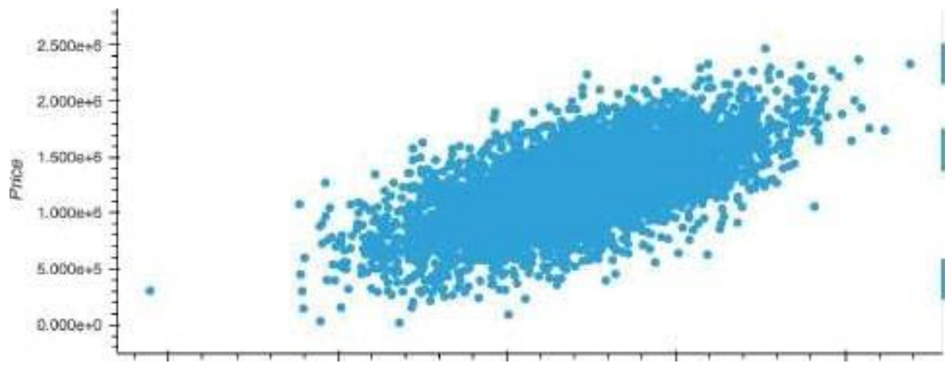

```
import holoviewe ae bv
ñv.esxension('boxeh'}
UsA oueing.bvpIm.hist{”Price”1
```




```
impnrr holommn  B lv
bv.eaxeneion('   eh'}
USAboueing.bvpIm.ecatterIx-'Avg. Area aouse &qe’, y-'Price’l
```



```
impnm holomewn  n hv
bv.erreneioo('  eñ'}
USAAoueing.bvpIm.scatter(x-'Avg. Area inrome', y-'Price')
```

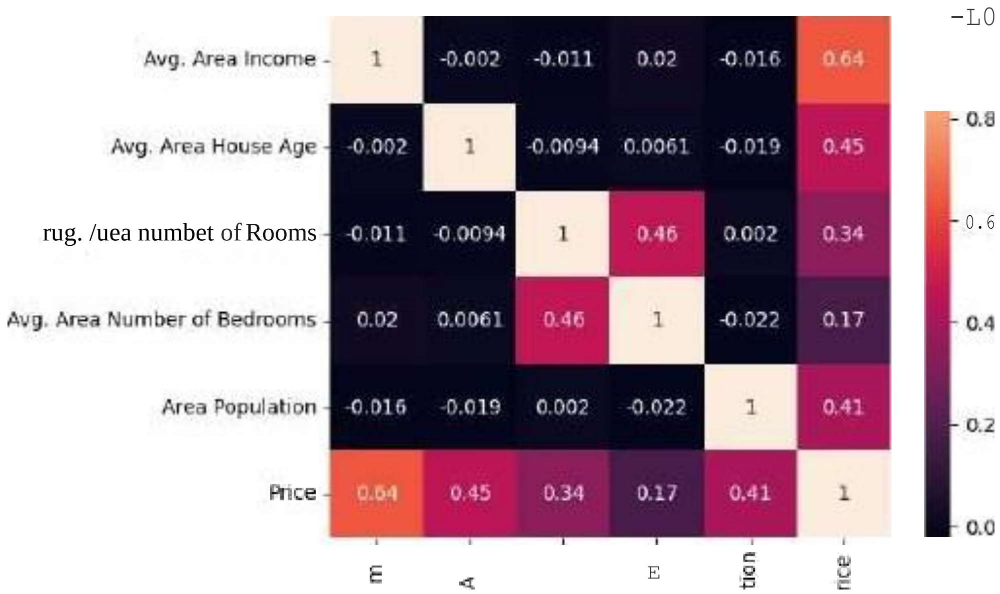



```
impom holovieee ae hv
bv.emeneion('bokeh'{
USAboueing.coirmoe
```

```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Dumber of Rorme'
       'Avg. Area Mumber of Bedrooms", 'Area vogulation', 'Price', 'Addreer'],
     dtype-'object';
```

```
sts.beatiap(nS.housing.corr(), anoot-True)
```

<âpyt from— input-JS $—4366Be6a5 o6 £>: i : Tut uroWarni rig: The fie£as £t va £ue of nunz r:icyonly in Datagram . norr in dspreca-
sue. tleatarap (IJSiif on sing. corr] [, an not-True)



- # Logistic Regression Model

X and y arrays

```
z - DsAhousiog{{'Avg. sea lncoom', 'Avg. Area mougs Age', 'Avq. Area number oI Boome',
               'Avg. Area Pueder o'f Bedrooms', 'Area Population' }
y - D°Ahousioq{'Price' '
```

- # Train Test Split

flow, let's sport the data into a training set and a testing set. We will train our mode4 on the trBiN‹ng set and then use the test set to.eyalua\e

```
from eklearn.modeJ_eelectioo imgort irain_teet_ep1ic

z_train x_tert, y_Craio, y_‹eet - train_test_sp?it x, y, teei_size-n.s randoz_eCate-•?#


bros eilearn iaport metrics
froe exlearn.code1_eelectioo imgort croee_vaI_rcnre

del croee_vaIsedel):
    pred - croee_val_score(model, J, y, cr-i0)
    reiurn•gred.sean{}

def•pmnt_evalraie true, predicted):
    mae - metrics.mean_abeolute_error(true, predicted)
    mee - met:ice.mean_rquared_error(true, gred   e0
    imse - np.rqr: metrics.mean_sgrareñ_error(true gre cued))
    ri_equare - :metrics.r•_ecoreftrue, predicted)
    print 'I: ', mee
    print 'msx:', mee
    print('RMSE:', rmse)
    pre nt ' R* 'squar e ' , • z square [
    print   '__________________________________')

def•maluate1true, predictedlr
    mae - met:ice.meao_abrolute_er•rgr(true, predicted}
    mee - metriJe v»_rquar#d_er'ror(true, gredi:ned
    imse - np.rqr: metrics.mean_sgrareñ_error(true gre cued))
    ri_equare -:metrics.r•_ecoreftrue, predicted)
    return mae, mee, rmse, ri_eguare
```

- # Preparing Data For Linear Regression

  - Linger As8umpt .

  - Oeussien Oistzdzutiora.

```
from eklearn.preprocessioq import Staodardscaler
froe exlearn.gipeJine izpom Pigelize

gipelioe - Pipeline{{
    ('etf_ecalar', stand dscaler{))


¥_traio - pipeline.fit_traoeform x_train]
x_teei - pipeline.transform(a_test)
```

• Linear Regression

```
froo 'eklear'n:Iinear_médeI import zizearRegreeeioo

lin_req- InaearRegression()•.
lin_reg.fitIx_train;y_trainj
```

```
• Linear8egreeeio'n.
ziaearRegresrion(
```

• /' Model Evaluation

Let's evaluate the model by checking our it's cDefficients and how we can interpret them.

```
grioz(Iin_reg.inzercegt_)
```

```
1228219.1492415662
```

```
co'éff_ñf - od.oaiarrame(Iin_reg.do'ef_, x.rolumne,rñlumna-{'Coeificieñt"}
coef_df
```

	cee*ficieo+
Avg. Area Income	Z3Zd7B7248A3
Avg. Area House Age	163B41.04&583
Avg. Area Numbar ol Pooms	121110.5s547B
Avg. A'rea Nurirber* of Bedroome	'Z89Z.8INt1B
Area Population	1f1ZZ234?37I

Doub Ie-click (or enter) to edit

• Predictions from our Model

Let's gra'b predictions off our test set and see ho'w well it did!

```
pred - lin_reg.precict‹x_:ea 1
```

```
gd.Dan rame( '=rue Valree'. •y_test, Predicted Ualues': pred)).nvpIo:.erattez(x-'True.valuee', v-'Predicted vaaree')
```

Residual Histogram

```
.od.Dan rame;\'rrror ?alree': y_teei - pred›)).hvglot.rde(
```

• Regression Evaluation Metrics

Here Bre three common evaluation metrics for regression problems:

- gean Ab&oTwte Error (MAE} is the meen of the a bsol ute value of the errors.

- Mean Squared Error (MBE) is the mean of the squa+ed errors.

$$\frac{1}{n}\sum^{n}(y_i - \hat{y}_i)^2$$

Root Mean Squared Error (RMSE) is the square root of the mean of the squared errors:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

CompBTl ng these metrics:

- gAT is the easiest 1o understand. because it’s the average error.
- MSC is more popular than MAE because MSE ’punishes’ larger errors, which tends to be useful In thé reel wor4d.
- ItMSG Is even more popuTBr than MSE. bees use RMSE is \ntergretabIe in the 'y’ units.

API of these are Yes Nelsons, because we want to minimize lhem.

```
teet  ed -’Iio_reg.predict(a_test)
’traio#red -Iio_reg.predim’(x_traio)

prior(’Teet set.evaluation:\n _______________________________________')
grio:_evaluate(y_ieet, tee:bred}
prim(’brain aei evaluation:Vn _______________________________________')
prioi_evalzate(y_irain, train#red)

resulte_df - pd.DataFreme(data-{{”rizeâr Regrereion', *evaJuazg{y_tesi, test  ed) , croes_val zizearBegressioo())Il.
                          co’lusm»•{'lel', 'I’, 'is'; ’VsE', 'Bz• °guare', ’Cross validation”])
```

```
Ceei eet evaluation:
_________________________________
MAE: 81135.56609336878
MSE: 10068422551.40088
RMSE: 100341.52954485436
R2 Square 0.9146818498754016
______________________________
Cram.eet.evaluation:
_________________________________
 : 8 1 48 0 . 4 9 9 ? 2 J 748 92
MSE: 10287043161.197224
RMSE: 101425.06180031257
Bz 5qu ar e D . 9 i 9 2 9 e 6 s 7 9 D 7 552 G
______________________________
```

- # Robust Regression

- ## Random Sample Consensus - RANSAC

```
froe exlearn.Iinear•model import  w°AcSegrerror•

model- VaSAcaegressor(bâee_eetimatoi-iinearmsgreeaion(), max_trials-ia0)
model.fitlx_train, y_train)

’teet#r’ed - mode•.predict(a_ieet}
traio#red - eodel.predict(a_train)

prior(’Teet set eoaluatioo:\n_______________________________________')
prior evaluat°(r_'est, teei#red}
print('========================================')
```

```
prior('Trâin rei evaluation:\n ________________________________')
prioi_evaloate(y_irain, train#red)

rerr1ie_df_z - pd.DafaFrame(data-{{"mnbuei egreeeion", *evaluate‹y_teet, test reds , croee_val{RABdACRegreseor()1]},
                            colusme-{'xodel', 'xA&', 'mss', 'VSs', 'B? sguare', "Cross validation'])
reeulie_df - reeuJte_df.appeod{resrIte_df_z, ignore_index-True]
```

```
/uer/local/lib/pytñoos.9/diet-oacxagea7srlearn/linear_eodel?_razsac.gy:#4?: ruiurewarning: 'base_eeiimator' wae r'
  warnirgs.who
Ceei eet evaluation:
_________________________________
MSE: 10829564871.937178
RMSE: 104065.195295724
R2 Square 0.9082320555368967
__________________________________

Crain eet evaluation:
_________________________________
: 83910.60259G69 117
MSE:
RMSE: 104552.13670255442
R2 Square 0.9142456773615641
__________________________________
‹ipythoo-input-i i-zoraaccbdzac•:i : Futurewarning: Cbs'frame.append method ie deprecated and will be removed fri
  rea«lts_df - reaulte_df.apgendlreeults_df_?, ignore_index-True)
```

- Ridge Regression

```
from eklearn.Iioear_mddeI import Ridge

model - Ridge alpda-ieo, solver-'cdoleety!, too-0.a00i, random_state- c)
model.fit x_traio, y_traiz)
pred - moéel.predict{«_teei)

teet  ed - modea.predict(a_ieet›
'traio#red - moéeJ.predict(a_zrain)

prior('Teet set.evaluation:\n ________________________________')
.grio:_evaluate(y_ieet, tee:bred}
prior('━━--———————•---━━━━━━━---━━━━━━━━━-')
prior('Trâin rei evaluation:\n ________________________________')
prioi_evaloate(y_irain, train#red)

rerr1ie_df_z - pd.DafaFrame(data-{{"Bidge Regreeeaon", *eva1uatef¥_"°°'; rest i*dl , cr e_oal{Ridqe()){{,
                            colusme-{'xodel', 'xA&', 'mss', 'VSs', 'B? sguare', "Cross validation'])
reeulie_df - reeuJte_df.appeod{resrIte_df_z, ignore_index-True]
```

```
Cee- eet evaluation:
_________________________________
: 8142B.64B3553*+°I36
ME : J0L5'326990d.B9°'6D9
CASE : 1 OO d 6 3 . 4 3 5 3 86 8 9 4 9 4
82 0 quare D . 913 9 G 286 VAA 6 4 G 0 9
__________________________________

main eet evaJuatioo:
_________________________________
IU+E : 8 1 9.7 E . 3 9 d 5 8 5 85fi' D T

V::E: iolsvs.461 s:i9:s?
a? souare n.9izaa44F?::#?444
__________________________________
‹iphon-iopui-i 4-dDaAivsef:7 ›'zt: xturewarning:.ibd•frame.append method ie degrecated and will be removed frx
  results_df = results_df.append(results_df_2, ignore_index=True)
```

- LASSO Regression

```
from eklearn.Ii r vosel import zaseo

model - iasso(alpba•O.i,
            prenomputeMrue;
#            .warm_star •True,
            positive-True,
            selection-'raodom',
            raodom state-:'l
model.fit x_traio, y_train)

test  ed - iédeJ.predict(x_teet}
train#red - modeJ.predict(a_train)

prior('Teet.eet evaluation:\n ________________________________')
grint_evaluet*lF'°®'  teai#red)
prior('━━--━━━━━━━━•-━━━━━━━-  ━━━━━━━━-')
.grio:('Eraiz eet eyaluatioo:\o ________________________________')
prioz_evaluat°(r_irain, train#red)

reruJie_df_z - pd.DafaFz   \data-{{'Laeeo Aegreeeion', •evaluate y_teet, restored) , croae_val{zareo())}},
                            cols.-I'xodel', 'xA&', 'mrs', 'VSs', 'B° sguare', "Cross vaJidatioo' )
rerr1ie_df - reeu1te_df.agpené resrIte_df_?, 'ignore_inéex-Tiue]
```

```
Ceei eet evaluation:
___________________________________
NAE : 8 1 L °I 5 . 6 98 5 J 7 2 6* E
t4BE : J d 06 B't 5 3 3 9 d . 3fi4 B E 3
T*E : J 00 3 I L . 6 B 3 2 L'4 7 * 6 48
B2 Bort AZe D . 9 J 4 68 LV' B B 5 B J J 1 6
___________________________________
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
Crain eet evaJuatioo:
___________________________________
 : 8 J 4 BD . 6 30 0 2 J 855 D 6
MSE: J d 28 T D 1 3 J 9 6 . 6°I4 * 9 B
WSE : J O L I * B . 0 6 J 9 ?5O D B't
B2 * o0Az:e D . 9 J 9 2 986 T T 6 * 9 5 5 d 5
___________________________________
<ipython-input-i4s-abtb:aadidi?>:?i: FutureJarning: Cbs frame.apgend method ie deprecated.and will be removed fra
  reeultg_dI - resulte_df.apgend reeults_df_?, ignore_index-True)
```

- Elastic Net

```
froe exlearn.Iinear• model import slaexixet

model - El%sticaei{alpba-D.? li_ratio-0:9, rele‹nion-'raodom' raodom_eiate-::j
model.fitlx_train, y_train)

'teet#r'ed - mode•.predict(a_ieet}
traio#red - eodel.predict(a_train)

prior('Teet set eoaluatioo:\n ________________________________')
prior evaluat°(r_'est, teei#red}
print('----------------------------------------')
prior('Train rei eyaluation: n ________________________________')
.grio:_evaluate(y_irain, traio#reé)

reerJte_df_z - pd.DataFr   (data-# rrJaezic Set Begressioo', *evaluate(y_iest, ieez#red) , croes_vaI(rlaeticrei ))]},
                            colmsm•-I'"°é•V  '  '  '*¥', 'VSs', 'B? square', 'Crose validation' )
resulte_df - reeults_df.appeod resuIte_df_i, i§oore_index-True)
```

```
Ceei eet evaluation:
___________________________________
MAE: 81184.43147330944
MSE: JdOFB05d 6B.470 LD6
RMSE: 100 3B9. 1 92 22 LOD99T
R2 Square 0.9146002670381437
___________________________________
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
'rrain set evaJuat ion :
```

```
    : 8 15? 7 . B 8 B 3 J 53 17 BI

    RMSE: 101485.34351373828
    R2 Square 0.9192027001474953
    __________________________________
    ‹iphon-input-1«-zae?fee?c#i4›'z?: Futurewx•ning• Cue frame.apgend metnod ie degrecated and will be removed fri
      results_df - resulte_df.apgendIreeults_df_?, iqnore_index-True)
```

Polynomial Regression

```
from eklearn.preprnre sion import PoTyoomiaiFea-uree

.goIy_reg  - PoIynomialFeatrree{degree-:)

z_train_?_d - poJy_reg.ILt_iransform(x_train}
Z feei_?_" - €"-F_reg.traoe*"* fE_teri)

liz_reg - zizearBegressioo()
iir_r 4.ric(x_‹rain_:_a,r_‹rai«t

test  ed - lio_reg.predict(z_test_:_d)
train#red - Iio_reg.predict(x_traio_?_d)

prior('Teet.eet evaluation:\n _______________________________')
grint_evaluet*lF'°®'  teai#red)
prior('━━--━━━━━━━•-━━━━━━---━━━━━━━-')
prior('Tralo eel evaluation:\n _______________________________')
prioz_evaluat°(r_irain, train#red)

reruJie_df_z - pd.DafaFr   \data-{{'PoJynomaFL Begreesioo', *evaluate(y_test, teetered), 0]],
                            cols.-I'xodel', 'xA&', 'mrs', 'VSs', 'B° sguare', 'cross va?idatioo' )
resrlie_df - reeul'te_df.agpené r%srIte_df_z, 'ignore_index-Tiue)
```

```
    see: eet evaluation:
    __________________________________
    IU+E : 8 1 L 7 I . B J 81 4 1 L96 9 S

    RMSE: 100409.08324260656
    R2 Square 0.914566932419506
    __________________________________

    Crain eet evaluation:
    __________________________________

    MSE:
         101323.67517519198
    R2 Square 0.9194599187853729
    __________________________________
    ‹ipython-input-i +-FDIbcieo0 iu-:?i: Future arning: CFe frame.apgend method ie dégrecated and will be removed fri
      reeultg_dI - resulte_df.apgend reeults_df_?, ignore_index-True)
```

- ## Stochastic Gradient Descent

```
froe exlearn.Iinear• model import sGoBegreesor

sgd_reg - SGDmegreeeor{i_iter_no_cbadge-zao, penalty-mono, etaa-4.DDDI, max_IfemI00##F#
sgd_reg.fit‹J_train  y_train)

'teet#r'ed - egd_ieg.predict a_test)
traio#red - egd_reg.gredi‹n x_train)

prior('Teet set eoaluatioo:\n _______________________________')
prioi_evaloate(y_zéet, tert#red
prior('━━•-━━━━━━━━---━━━━━━━━-•:-━━━━━━━━-')
prior('Train rei eyaluation: n _______________________________')
.grio:_evaluate(y_irain, traio#reé)
```

```
reruJie_df_z - pd.DataFraoe(data-{{’stochastic Gradient Descent', •evaluaze¿y_tee:, test red ,.e ;
                                   cols. -I’xodel', 'xA&', 'mrs', 'VSs', ’B° sguare', 'cross va?idatioo' )
rerr1ie_df - reeu1te_df.agpené resrIte_df_?, ignore_inéex-Tiue]
```

```
Ceei eet evaluation:
_______________________________
NAE : 8 1 L °I 5 . 5 6 I? 45 706 E
RMSE: 100341.52846153275
R2 Square 0.9146818517176497
_______________________________
Crain eet evaJuatioo:
_______________________________
   : 8J4BD.49B7J82496
     10287043161.210154
RMSE: 101425.06180037631
R2 Square 0.9192986579074511
_______________________________
‹ipython-input-i #-4• I?99bi?i#›:i : FutureJarning: Cbs frame.apgend method ie deprecated.and will be removed fra
  reeultg_dI - resulte_df.apgend reeults_df_?, ignore_index-True)
```

- •/" Artficial Neural Network

```
froe ieneorflow.xeras.modele imgort °equentiaL
from ieneorIIov:draw.layere ieport :oput, Deore,.Bctivatioo, Dropout
from teneorllov.xeras.optimizers iopñr: Adam

z_traio - np.array x_train)
z_teet - nP arraF(x_test)
y_traio - np.array y_train)
y_teei - np.array(y_test)

model - segueotiaJ )

model.add Deoee(a_train.ebape[l , activation-'zelu' )
model.add Oe e(a°, éctivaiion-'rem’))
f model.addf  oPoutlO.i ))

model.add{Deoee(6 , activation-'rem’))
# moael:add o o’p:mt o.z

model.add Deoee(izs, activation-'refu’)+
* ••›a°,.add(D%p‹mt o.s

model.add Deoee(aic ’activaiion•'reIu’)l
model.add{Dropoui(a.i}}
model.add(Dense(1))

model.compile(o#imizer-Vam{a.00#DI), loes-'xse’)

r - model.fit(i_irain, y_train
            vaJidation_daia-lx_test,y_test},
            bazcñ_sise-i,
            epos h'a - LO )
```

```
3500}2500 [------------] - 13e 3me/step - loss: l63ssz7zss1s:.oaoo - val loss: i6sai<s7zDBsz.aa'
3500/3500 [------------] - 12s 3ms/step - loss: 1635543154688.0000 - val loss: 1657098731520.00
Epoch 3/10
3500/3500 [------------j —i i s ams / seep — los s : i sa *a sz s n i z i a . o ono — vai_loss : i a s*oa " ** n ssa . o or
3500/3500 [------------] - 13s 4ms/step - loss: 1623434723328.0000 - val loss: 1637010898944.00
3500/3500 [------------] - 13s 4ms/step - loss: 1599173427200.0000 - val loss: 1601585020928.00
3500/3500 [------------] - 12s 3ms/step - loss: 1547818500096.0000 - val loss: 1531431092224.00
```

```
3500/3500 [ ---        ---━━━━━━━━--; - :ce 3sm/etep - lose: l sisa:zissz8.aaoo - val loes: i o6vs sioszz:ao'
Egon ft 8 / L 0
350 0/ 35OO [━━━━---  ━━━━━━━━---━━━━━━━━-- - isa 4ms/atep - loss: ::siaaoo:iiez.oouo - val loss: no i i*zB:i :oo'

350 0/ 3500 [━━━━---  ━━━━━━━━---  ━━━━━━━━--] - 13s 4ms/step - loss: 1053762191360.0000 - val loss: 931060580352.000

3500/?500 [━━━━━━-━━━━━━━━━━━━━━━━━━━━- - ozs cmm/step - lose: z<zaiaz ssoe.aoao - val_loes: ses scazeiDi.ooaa
```

```
pd.DaiaFrame({'True Valree': y_test, 'Predicted values': gred}).bvgIoi.scatter(x-'True valuee', y-'Predicted vaiuea'l
```

```
.pd.Dair°rame;z.history}
```

II	I.03b077e+12	1. 1T0e+IZ
1	1 .035543e+1 Z	1.657099e+12
2	1.632863e+12	1 0fSOfl7e+72
0	T .023435e+12	I. 7Ot I e•12
4	I :50B173e+12	1.601585e+12
'i	t. 5478 T 0e+1 Z	1.531431e+12
B	I .éd1.BB3e+12	1.408755e+1 Z
7	1 14 B0e+12	1.3001'4 t e+IZ
8	1.053762e+12	9.310606e+11
8	7.4B 7B3e+11	5.957625e+11

```
gd.Dat 'rame;i.bietñry}.b Jot.line(y-{'loss', 'val_lose')
```

```
test  ed - model.predict(s_teat}
traio#red - :endeJ.predict(a_train)

prior('Teet aet evaluation:\n ______________________________')
prioi_evaloate(y_ieet, teai#red}

grint('Train rei evaluation: n ______________________________')
prioi_evalzate(y_irain, train#red)

resulte_df_z - pd.DafaFr   #data•{{'artfici'aI 8enal metvorx', '*evalraie y_teat, teetered), a{ ,
                            .col•m•m ¿'nodél', 'xAc', . 'nss', 'msss', ''ai sguare', 'croas validation'])
reruJie_df - reeu1te_df.appeod resuIte_df_:, rgoore_index-True
```

```
47/47 [------------------------------] - 0s 2ms/step
mo/iia ¿━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━] - 0s 2ms/step
reef eet evaluations
___________________________________
M:"24S96.46009GO566

      771856.5495098421
R2 Square -4.048393327894736
________________________________
Cram eet evaluation:
___________________________________

MSE: 574624782796.526
      758040.0931326298
R2 Square -3.5079028487194366
________________________________
‹ipython-inguz-id+-bzcbase? ii»:ii: Future arning: CFe frame.apgend method ie dégrecated and will be removed fri
  results_df - resulte_df.agpeod reeults_df_?, ignore_index-True)
```

- Random Forest

```
from eklearx.eoeemble impo/r eandoWoreetRegreeeor

rf_reg - BandomForestRegreeeorln_ertinators-inu0)
rf_rsg.fitf* _traio, y_t:ain)

tert#red - rf_reg.predict(a_test
traio#red - rf_reg.gredici(a_traio)

prior('meet set evaluation:\n ______________________________')
prioi_evaluate(y_iest, teetered}

.grio:('Eraiz eet eyaluatioo:\o ______________________________')
prioz_evaluat°(r_irain, train#red)

reeulie_df_° - pd.DataFrame(data-{['Baodom Foreei "  *evaIuaie#y_tegt, teetered), :a{ ,
                        colomnr-{'model','xA&','nss','wSs','Bz sguare',!iross validatipo' )
rerr1ie_df - reeu1te_df.agpené resrIte_df_?, 'ignore_inéex-Tiue]
```

```
    Ceei eet evaluation:
    _________________________________
       :9 8 22 L . 5 0 0 2 1 J 748 T 6
    fi*E: J 0 LV I B T 5 2 9 d . 97 0 T 9.7
    DBE : 1 L4 D'B . ?8 5 2 68O*6 D T B
    Bi square D.8?9ñ44A?4#9? 4
    _________________________________
    Cram.eet.evaluation:
    _________________________________
    MAE: 35239.40946017513
    MSE: 1979420849.5510504
    RMSE: 44490.68272740991
    R2 Square 0.9844715418588803
    _________________________________
    <iphon-znput-1 s-a2si?d??a7 0>'z?: Futurewx•ning• Cue frame.apgend metnod ie degrecated and will be removed fri
      reeults_df - resulte_df.aggend(reeulte_df_i, ignore_inde True)
```

- Support Vector Machine

```
bros eilearn.rem import svB

sm reg - GB(kernel-'rbi', K-?O44B0V„ epsilon-0.00?
sve_reg.fit{J_train, y_train)

teet  ed - rrn_reg.predict(a_test)
'traio#red - evm_reg.predim'(x_traio)

prior('Teet eet evaluation:\n ______________________________')
prioi_evaloate(y_ieet, tegi#red

prior('Trâin rei evaluation:\n ______________________________')
prioi_evaloate(y_irain, train#red)

rerr1ie_df_z - pd.DafaFrame(data-{{'°vW Aegreeeor' *evalraie y_tert, ieet#red), a){,
                        colusme-{'xodel', 'xA&', 'mss', 'VSs', 'B? sguare', 'Cross validation'])
reeulie_df - reeuJte_df.appeod{resrIte_df_z, ignore_index-True]
```

```
    Cee- eet evaluation:
    _________________________________
    MAE: 87205.73051021632
    MSE: 11720932765.275513
    RMSE: 108263.25676458987
    R2 Square 0.9006787511983232
    _________________________________
    main eel evaluation:
    _________________________________
```

```
MSE: 9363827731.411339
RMSE: 96766.87310960988
R2 Square 0.9265412370487783
```

```
<ipython-input-170-5d3d7f23418c>:17: FutureWarning: The frame.append method is deprecated and will be removed fro
  results_df = results_df.append(results_df_2, ignore_index=True)
```

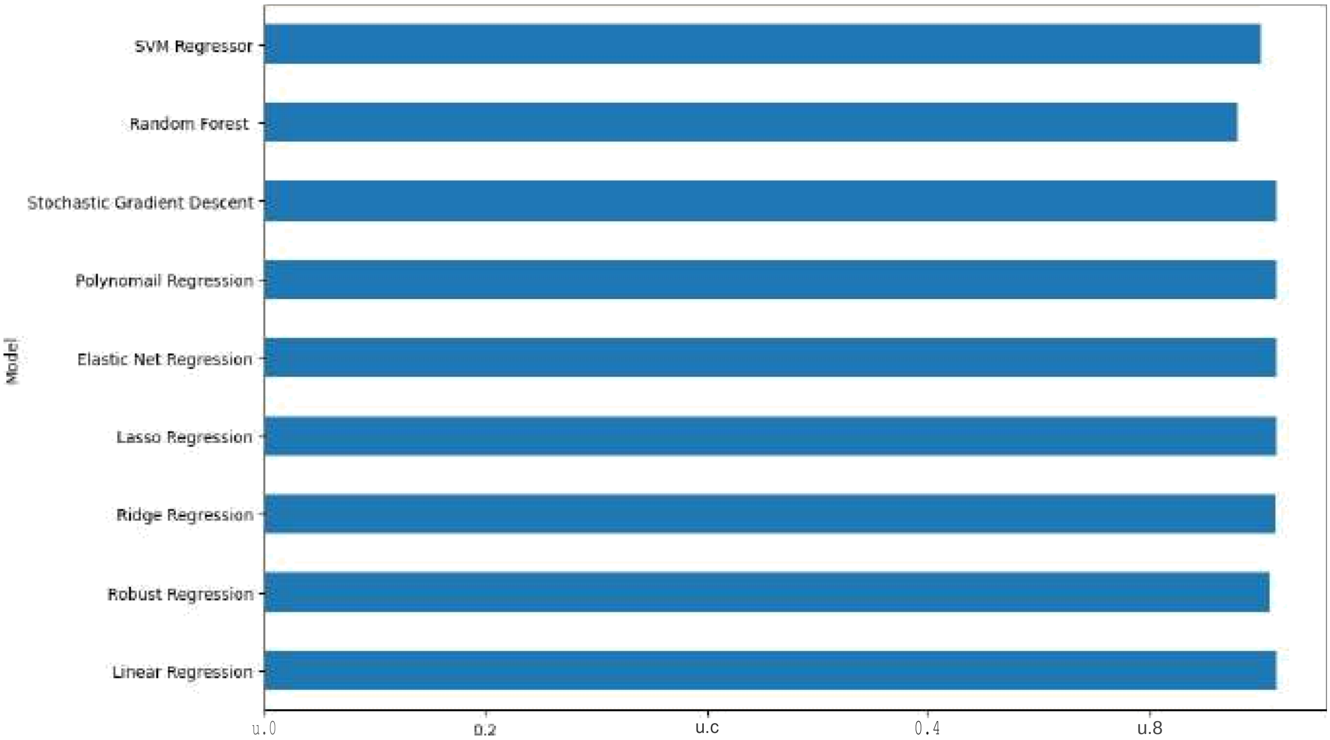
```
results_df
```

	Model	MAE	MSE	RMSE	M square	Cross ¥aiiaatioo
0	Linear Ragraaaion	811 M .S880B3	1.00084Ze+1 0	100341.529545	9.9140BZ	08bB0f1
1	Ro ouat Ragreaaion	834B2.9B1 773	1.083068e+ 1 0	104065.1B52B6	0.B08232	0.81"379
2	Rdge Ragreaaion	81438. 83ñs' Laaso	1.01 53Z7e+ 1 0	100703.43533a	0.9138g3	0.B173?8
3	Ragreaaion B 1!	M .088ñ17	1.000845e+ IB	100041.8B3Z15	0.8140B2	0B?BZ#5
4	Elastic Net Regreaa!on	B 118-4 4314?3	1.007805e+ IB	109S8B .4B?3Z1	0.914000	0.000000
5	Pa lynornnil Regreaaion	B It 74.01844t	1.00818Be+1 0	10540B.0B3243	0.914567	0.000000
II	Stochastic Orad lent Deene nt	B 11 M .084246	1.00084Ze+1 0	100341.5Z84g2	0.914082	0u00000
7	Ftsndom Forest	8kc1.000212	1.41748Be+1 0	118058.ZB5288	C8?88B#	0.0OOODO
8	SVF'J FJegroae r	872'O5.730010	1.1T2093e+ 1 0	10B203.25678 5	0.900679	0.917379

- Models Comparison

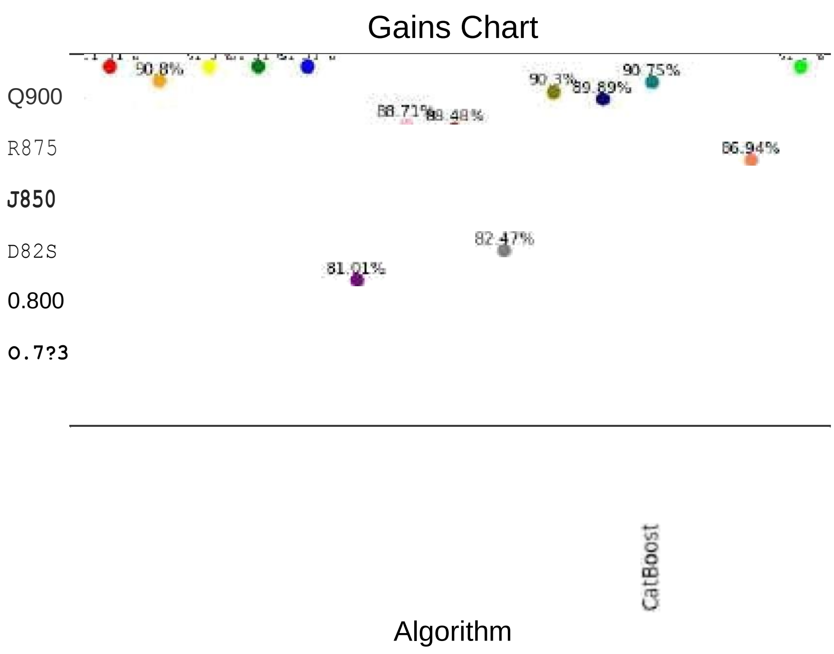
```
resri-e_df.ser_:ndex‹'Model , inplace-=rue
rest e_ñf 'Sz square'].rlc:1 izd-'nard', Tiger:e-{1?, 8))
```

```
<Axes: ylabel-'Model':
```




```
impor: numoy as op
ieporz mazplo=iib.pyolot ae glz

#defioe velres and cor espondizg algorithm names
```

```
:va triea - [ a . s ¥ s a a a yazs4 s s 7 a  , a . 9o: s sP a a Ja as e z s z., °.a . é t4 s e 7 s y s s' s sais T T , a . s Jso e oa s 4 a z a'JIs * ,
            0 . 9 T 5 B B B °I °i3 LV 24 9 6 9 , 0 . 8 LD D 9 4 4.7 73 7 L 64-' , 0 . B87g 9 6.7 4 4 9 7 J 698 B , 0 . 8 8 d 8 3' T 4 6 3 B 2 9 d 6-3'6,
            0' I B 24 6 6 6 8*0 878 7*' B 9 5 , 0 . 90 7 9 9 0 6 4 4*3 7 L 6 6 B? , 0 . 898 9 3 6*2 B 5 5 d 460 6 9 , B . 9 0 T45 94 0 7 06 6 L2 * ,
            0.7508569551273301, 0.8693855376039462, 0.9150085163619179]
'algoriihm_namer•- ['zogirtic Begreazion', 'Robret?, !TbeiI men', '&idge',• 'zaseo', 'sVaztic8m:'; 'sandocForert',
                      'x6Boosi', 'Adabooei', 'GradieocBoost', 'iigbteSN', 'catBoori', 'Deciaioo free', 'KKN', 'SvM']

f coooert véIree'io percentage ané rouoé'to tvd decimal glacee
vairee   t - {round;yal.* i0a, z) for val in valuee{

#'creaie a list of colors féi each algorithm
colore' - ['red', 'orange', 'yellom', 'green', 'bJue', '{nrrple'  'gink' 'brovo'; 'gray™, 'oJive',
            !navy', 'teal', 'magenta', 'coral', 'lime'

é create the Aft châit•
fig, ax - pit.rubplots{)
ior i in range{len valuer)):
    plt.plo't({dlgoritñm_ramee[ij , [va1uee{ij], color•coIoie\i} marker-'o'}
   'pJ‹ .texrlalgorithm_zamee{i{, valree[i}+0:00s :etr¿vaIree#ct{i])+'%' ña-'ceñier', footeize-#}
plf.xlabel\'AlgoiitM' Ionieiae-I•#
pit.yJabel¿'  ion', footeise-i?j
pit.:iile 'saioe thart', Ioñteize-i4j
pit.sxicae{rotaiioc-s0 ›
glt.iight_layoui ()
```



Colab paid products Cancel contracts here

✓ 2s completed at 9:26 PM