

474CIS- Lab Project Report

Title:

Predict the median value of owner occupied homes

By

Name

Sara Ali 4116615369
E. Alkhalaf 411875532
H. Alwan Mohammad 433815369

ID@kku.edu.sa

Submitted to

Lab Techer Name

Course Coordinator : Dr. Anandhi

Mrs.Aisha Ali Al Maalawi

King Khaled University

College of Computer Science

Department of Information Systems

Semester III - Year 2022

Project Aim:

The aim of the project is to predict the median value of owner occupied homes using various regression techniques on the Boston housing dataset.

Description:

The project aims to predict the median value of owner-occupied homes using various statistical and machine learning techniques. The dataset used for this project includes various features such as crime rate, average number of rooms per dwelling, pupil-teacher ratio, and more. By analyzing these features and their impact on the median value of homes, the project aims to develop a predictive model that can accurately estimate the value of a home based on its characteristics. This information can be useful for real estate agents, homeowners, and potential buyers looking to make informed decisions about buying or selling properties.

Models Used & Its Description:

User model (Liner Regression)

A collection of statistical techniques called regression analysis is used to estimate the associations between a dependent variable and one or more independent variables. It can be used to model how strongly variables will be related in the future and to gauge the strength of that relationship. It can be used to simulate the future relationship between variables and gauge how strongly the relationships between them are currently.

decision tree :is a popular method of creating and visualizing predictive models and algorithms

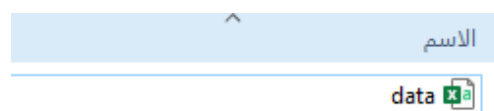
data pre-processing:

cleaning : is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data.

- handle with(missing values) as they refer to several different things. Perhaps the field was not applicable, the event did not occur, or the data was not available.

Dataset Used & Its Description:

This dataset contains information collected by the U.S Census Service concerning housing in the area of Boston Mass. It was obtained from the StatLib archive (<http://lib.stat.cmu.edu/datasets/boston>), and has been used extensively throughout the literature to benchmark algorithms. However, these comparisons were primarily done outside of Delve and are thus somewhat suspect. The dataset is small with only 506 cases.



	AGE	SEX	HT	WT	HAIR	SKN	HTG	HTG2	HTG3	HTG4	HTG5	HTG6	HTG7	HTG8	HTG9	HTG10	HTG11	HTG12	HTG13	HTG14	HTG15	HTG16	HTG17	HTG18	HTG19	HTG20	HTG21	HTG22	HTG23	HTG24	HTG25	HTG26	HTG27	HTG28	HTG29	HTG30	HTG31	HTG32	HTG33	HTG34	HTG35	HTG36	HTG37	HTG38	HTG39	HTG40	HTG41	HTG42	HTG43	HTG44	HTG45	HTG46	HTG47	HTG48	HTG49	HTG50	HTG51	HTG52	HTG53	HTG54	HTG55	HTG56	HTG57	HTG58	HTG59	HTG60	HTG61	HTG62	HTG63	HTG64	HTG65	HTG66	HTG67	HTG68	HTG69	HTG70	HTG71	HTG72	HTG73	HTG74	HTG75	HTG76	HTG77	HTG78	HTG79	HTG80	HTG81	HTG82	HTG83	HTG84	HTG85	HTG86	HTG87	HTG88	HTG89	HTG90	HTG91	HTG92	HTG93	HTG94	HTG95	HTG96	HTG97	HTG98	HTG99	HTG100	HTG101	HTG102	HTG103	HTG104	HTG105	HTG106	HTG107	HTG108	HTG109	HTG110	HTG111	HTG112	HTG113	HTG114	HTG115	HTG116	HTG117	HTG118	HTG119	HTG120	HTG121	HTG122	HTG123	HTG124	HTG125	HTG126	HTG127	HTG128	HTG129	HTG130	HTG131	HTG132	HTG133	HTG134	HTG135	HTG136	HTG137	HTG138	HTG139	HTG140	HTG141	HTG142	HTG143	HTG144	HTG145	HTG146	HTG147	HTG148	HTG149	HTG150	HTG151	HTG152	HTG153	HTG154	HTG155	HTG156	HTG157	HTG158	HTG159	HTG160	HTG161	HTG162	HTG163	HTG164	HTG165	HTG166	HTG167	HTG168	HTG169	HTG170	HTG171	HTG172	HTG173	HTG174	HTG175	HTG176	HTG177	HTG178	HTG179	HTG180	HTG181	HTG182	HTG183	HTG184	HTG185	HTG186	HTG187	HTG188	HTG189	HTG190	HTG191	HTG192	HTG193	HTG194	HTG195	HTG196	HTG197	HTG198	HTG199	HTG200	HTG201	HTG202	HTG203	HTG204	HTG205	HTG206	HTG207	HTG208	HTG209	HTG210	HTG211	HTG212	HTG213	HTG214	HTG215	HTG216	HTG217	HTG218	HTG219	HTG220	HTG221	HTG222	HTG223	HTG224	HTG225	HTG226	HTG227	HTG228	HTG229	HTG230	HTG231	HTG232	HTG233	HTG234	HTG235	HTG236	HTG237	HTG238	HTG239	HTG240	HTG241	HTG242	HTG243	HTG244	HTG245	HTG246	HTG247	HTG248	HTG249	HTG250	HTG251	HTG252	HTG253	HTG254	HTG255	HTG256	HTG257	HTG258	HTG259	HTG260	HTG261	HTG262	HTG263	HTG264	HTG265	HTG266	HTG267	HTG268	HTG269	HTG270	HTG271	HTG272	HTG273	HTG274	HTG275	HTG276	HTG277	HTG278	HTG279	HTG280	HTG281	HTG282	HTG283	HTG284	HTG285	HTG286	HTG287	HTG288	HTG289	HTG290	HTG291	HTG292	HTG293	HTG294	HTG295	HTG296	HTG297	HTG298	HTG299	HTG300	HTG301	HTG302	HTG303	HTG304	HTG305	HTG306	HTG307	HTG308	HTG309	HTG310	HTG311	HTG312	HTG313	HTG314	HTG315	HTG316	HTG317	HTG318	HTG319	HTG320	HTG321	HTG322	HTG323	HTG324	HTG325	HTG326	HTG327	HTG328	HTG329	HTG330	HTG331	HTG332	HTG333	HTG334	HTG335	HTG336	HTG337	HTG338	HTG339	HTG340	HTG341	HTG342	HTG343	HTG344	HTG345	HTG346	HTG347	HTG348	HTG349	HTG350	HTG351	HTG352	HTG353	HTG354	HTG355	HTG356	HTG357	HTG358	HTG359	HTG360	HTG361	HTG362	HTG363	HTG364	HTG365	HTG366	HTG367	HTG368	HTG369	HTG370	HTG371	HTG372	HTG373	HTG374	HTG375	HTG376	HTG377	HTG378	HTG379	HTG380	HTG381	HTG382	HTG383	HTG384	HTG385	HTG386	HTG387	HTG388	HTG389	HTG390	HTG391	HTG392	HTG393	HTG394	HTG395	HTG396	HTG397	HTG398	HTG399	HTG400	HTG401	HTG402	HTG403	HTG404	HTG405	HTG406	HTG407	HTG408	HTG409	HTG410	HTG411	HTG412	HTG413	HTG414	HTG415	HTG416	HTG417	HTG418	HTG419	HTG420	HTG421	HTG422	HTG423	HTG424	HTG425	HTG426	HTG427	HTG428	HTG429	HTG430	HTG431	HTG432	HTG433	HTG434	HTG435	HTG436	HTG437	HTG438	HTG439	HTG440	HTG441	HTG442	HTG443	HTG444	HTG445	HTG446	HTG447	HTG448	HTG449	HTG450	HTG451	HTG452	HTG453	HTG454	HTG455	HTG456	HTG457	HTG458	HTG459	HTG460	HTG461	HTG462	HTG463	HTG464	HTG465	HTG466	HTG467	HTG468	HTG469	HTG470	HTG471	HTG472	HTG473	HTG474	HTG475	HTG476	HTG477	HTG478	HTG479	HTG480	HTG481	HTG482	HTG483	HTG484	HTG485	HTG486	HTG487	HTG488	HTG489	HTG490	HTG491	HTG492	HTG493	HTG494	HTG495	HTG496	HTG497	HTG498	HTG499	HTG500	HTG501	HTG502	HTG503	HTG504	HTG505	HTG506	HTG507	HTG508	HTG509	HTG510	HTG511	HTG512	HTG513	HTG514	HTG515	HTG516	HTG517	HTG518	HTG519	HTG520	HTG521	HTG522	HTG523	HTG524	HTG525	HTG526	HTG527	HTG528	HTG529	HTG530	HTG531	HTG532	HTG533	HTG534	HTG535	HTG536	HTG537	HTG538	HTG539	HTG540	HTG541	HTG542	HTG543	HTG544	HTG545	HTG546	HTG547	HTG548	HTG549	HTG550	HTG551	HTG552	HTG553	HTG554	HTG555	HTG556	HTG557	HTG558	HTG559	HTG560	HTG561	HTG562	HTG563	HTG564	HTG565	HTG566	HTG567	HTG568	HTG569	HTG570	HTG571	HTG572	HTG573	HTG574	HTG575	HTG576	HTG577	HTG578	HTG579	HTG580	HTG581	HTG582	HTG583	HTG584	HTG585	HTG586	HTG587	HTG588	HTG589	HTG590	HTG591	HTG592	HTG593	HTG594	HTG595	HTG596	HTG597	HTG598	HTG599	HTG600	HTG601	HTG602	HTG603	HTG604	HTG605	HTG606	HTG607	HTG608	HTG609	HTG610	HTG611	HTG612	HTG613	HTG614	HTG615	HTG616	HTG617	HTG618	HTG619	HTG620	HTG621	HTG622	HTG623	HTG624	HTG625	HTG626	HTG627	HTG628	HTG629	HTG630	HTG631	HTG632	HTG633	HTG634	HTG635	HTG636	HTG637	HTG638	HTG639	HTG640	HTG641	HTG642	HTG643	HTG644	HTG645	HTG646	HTG647	HTG648	HTG649	HTG650	HTG651	HTG652	HTG653	HTG654	HTG655	HTG656	HTG657	HTG658	HTG659	HTG660	HTG661	HTG662	HTG663	HTG664	HTG665	HTG666	HTG667	HTG668	HTG669	HTG670	HTG671	HTG672	HTG673	HTG674	HTG675	HTG676	HTG677	HTG678	HTG679	HTG680	HTG681	HTG682	HTG683	HTG684	HTG685	HTG686	HTG687	HTG688	HTG689	HTG690	HTG691	HTG692	HTG693	HTG694	HTG695	HTG696	HTG697	HTG698	HTG699	HTG700	HTG701	HTG702	HTG703	HTG704	HTG705	HTG706	HTG707	HTG708	HTG709	HTG710	HTG711	HTG712	HTG713	HTG714	HTG715	HTG716	HTG717	HTG718	HTG719	HTG720	HTG721	HTG722	HTG723	HTG724	HTG725	HTG726	HTG727	HTG728	HTG729	HTG730	HTG731	HTG732	HTG733	HTG734	HTG735	HTG736	HTG737	HTG738	HTG739	HTG740	HTG741	HTG742	HTG743	HTG744	HTG745	HTG746	HTG747	HTG748	HTG749	HTG750	HTG751	HTG752	HTG753	HTG754	HTG755	HTG756	HTG757	HTG758	HTG759	HTG760	HTG761	HTG762	HTG763	HTG764	HTG765	HTG766	HTG767	HTG768	HTG769	HTG770	HTG771	HTG772	HTG773	HTG774	HTG775	HTG776	HTG777	HTG778	HTG779	HTG780	HTG781	HTG782	HTG783	HTG784	HTG785	HTG786	HTG787	HTG788	HTG789	HTG790	HTG791	HTG792	HTG793	HTG794	HTG795	HTG796	HTG797	HTG798	HTG799	HTG800	HTG801	HTG802	HTG803	HTG804	HTG805	HTG806	HTG807	HTG808	HTG809	HTG810	HTG811	HTG812	HTG813	HTG814	HTG815	HTG816	HTG817	HTG818	HTG819	HTG820	HTG821	HTG822	HTG823	HTG824	HTG825	HTG826	HTG827	HTG828	HTG829	HTG830	HTG831	HTG832	HTG833	HTG834	HTG835	HTG836	HTG837	HTG838	HTG839	HTG840	HTG841	HTG842	HTG843	HTG844	HTG845	HTG846	HTG847	HTG848	HTG849	HTG850	HTG851	HTG852	HTG853	HTG854	HTG855	HTG856	HTG857	HTG858	HTG859	HTG860	HTG861	HTG862	HTG863	HTG864	HTG865	HTG866	HTG867	HTG868	HTG869	HTG870	HTG871	HTG872	HTG873	HTG874	HTG875	HTG876	HTG877	HTG878	HTG879	HTG880	HTG881	HTG882	HTG883	HTG884	HTG885	HTG886	HTG887	HTG888	HTG889	HTG890	HTG891	HTG892	HTG893	HTG894	HTG895	HTG896	HTG897	HTG898	HTG899	HTG900	HTG901	HTG902	HTG903	HTG904	HTG905	HTG906	HTG907	HTG908	HTG909	HTG910	HTG911	HTG912	HTG913	HTG914	HTG915	HTG916	HTG917	HTG918	HTG919	HTG920	HTG921	HTG922	HTG923	HTG924	HTG925	HTG926	HTG927	HTG928	HTG929	HTG930	HTG931	HTG932	HTG933	HTG934	HTG935	HTG936	HTG937	HTG938	HTG939	HTG940	HTG941	HTG942	HTG943	HTG944	HTG945	HTG946	HTG947	HTG948	HTG949	HTG950	HTG951	HTG952	HTG953	HTG954	HTG955	HTG956	HTG957	HTG958	HTG959	HTG960	HTG961	HTG962	HTG963	HTG964	HTG965	HTG966	HTG967	HTG968	HTG969	HTG970	HTG971	HTG972	HTG973	HTG974	HTG975	HTG976	HTG977	HTG978	HTG979	HTG980	HTG981	HTG982	HTG983	HTG984	HTG985	HTG986	HTG987	HTG988	HTG989	HTG990	HTG991	HTG992	HTG993	HTG994	HTG995	HTG996	HTG997	HTG998	HTG999	HTG1000
--	-----	-----	----	----	------	-----	-----	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	---------

1) Preparing the given Dataset

```
In [2]: df = pd.read_csv('data.csv')
df
```

511 rows × 14 columns

```
In [3]: df.head()

Out[3]:
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2

```
In [4]: df.isnull().sum()
```

```
Out[4]:
```

CRIM	0
ZN	0
INDUS	0
CHAS	0
NOX	0
RM	5
AGE	0
DIS	0
RAD	0
TAX	0
PTRATIO	0
B	0
LSTAT	0
MEDV	0

dtype: int64

dtype: int64

```
In [5]: df.describe()
```

```
Out[5]:
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
count	511.000000	511.000000	511.000000	511.000000	511.000000	506.000000	511.000000	511.000000	511.000000	511.000000	511.000000	511.000000	511.000000	511.000000
mean	3.584139	11.252446	11.151096	0.068493	0.554757	6.287589	68.616243	3.783876	9.485323	407.440313	18.500000	356.600900	12.879550	22.682192
std	8.564433	23.234838	6.828175	0.252838	0.115310	0.703802	28.099130	2.098631	8.688469	167.903532	2.200348	90.882679	7.797416	9.484262
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000	1.129600	1.000000	187.000000	12.600000	0.320000	1.730000	5.000000
25%	0.082325	0.000000	5.190000	0.000000	0.449000	5.885500	45.050000	2.100350	4.000000	279.500000	17.400000	374.710000	7.065000	17.050000
50%	0.261690	0.000000	9.690000	0.000000	0.538000	6.209000	77.300000	3.152300	5.000000	330.000000	19.100000	391.340000	11.450000	21.200000
75%	3.621175	12.500000	18.100000	0.000000	0.624000	6.629750	94.050000	5.118000	24.000000	666.000000	20.200000	396.210000	17.105000	25.000000
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000	12.126500	24.000000	711.000000	23.000000	396.900000	76.000000	67.000000

```
In [17]: df.info()
```

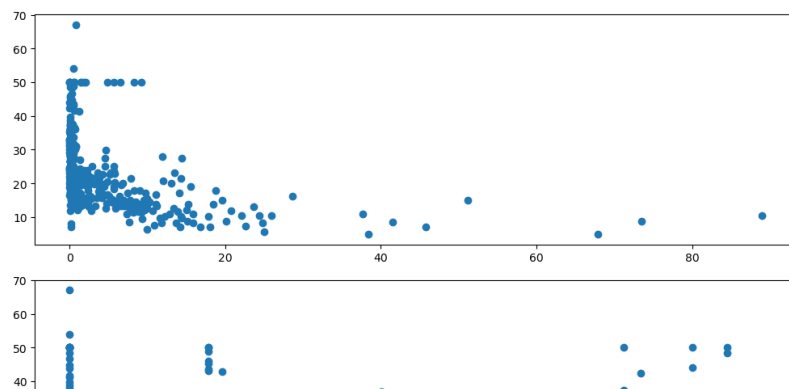
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 511 entries, 0 to 510
Data columns (total 14 columns):
#   Column  Non-Null Count  Dtype  
---  -
0    CRIM    511 non-null    float64
1    ZN       511 non-null    float64
2    INDUS   511 non-null    float64
3    CHAS     511 non-null    int64  
4    NOX      511 non-null    float64
5    RM       506 non-null    float64
6    AGE      511 non-null    float64
7    DIS      511 non-null    float64
8    RAD      511 non-null    int64  
9    TAX      511 non-null    int64  
10   PTRATIO  511 non-null    float64
11   B        511 non-null    float64
12   LSTAT    511 non-null    float64
13   MEDV     511 non-null    float64
dtypes: float64(12), int64(3)
memory usage: 56.0 KB
```

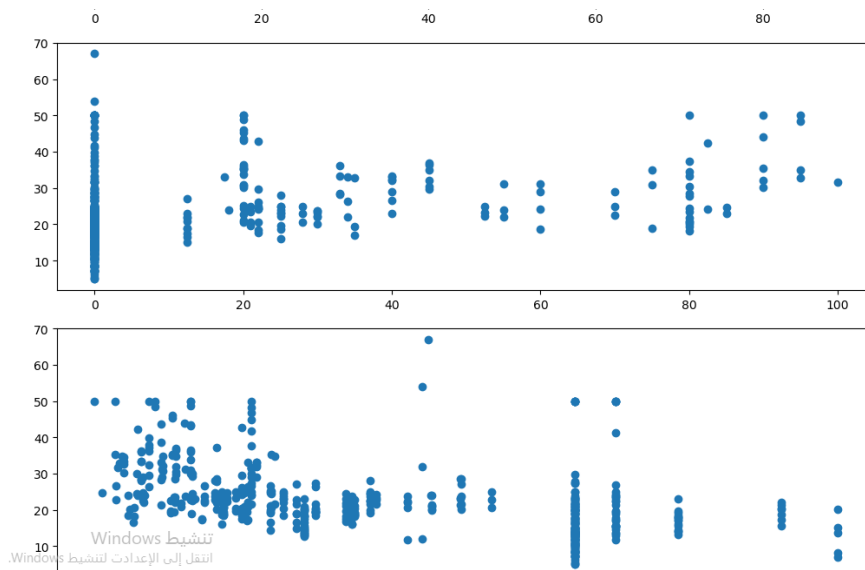
انتقل إلى الإحداثيات لتفصيل Windows

Visualization

```
In [ ]: # Visualization
```

```
In [6]: fig, axes = plt.subplots(figsize=(10,10),nrows=3,ncols=1)
col = 0
for ca in axes:
    ca.scatter(df.iloc[:,col],df['MEDV'])
    col+=1
plt.tight_layout()
```

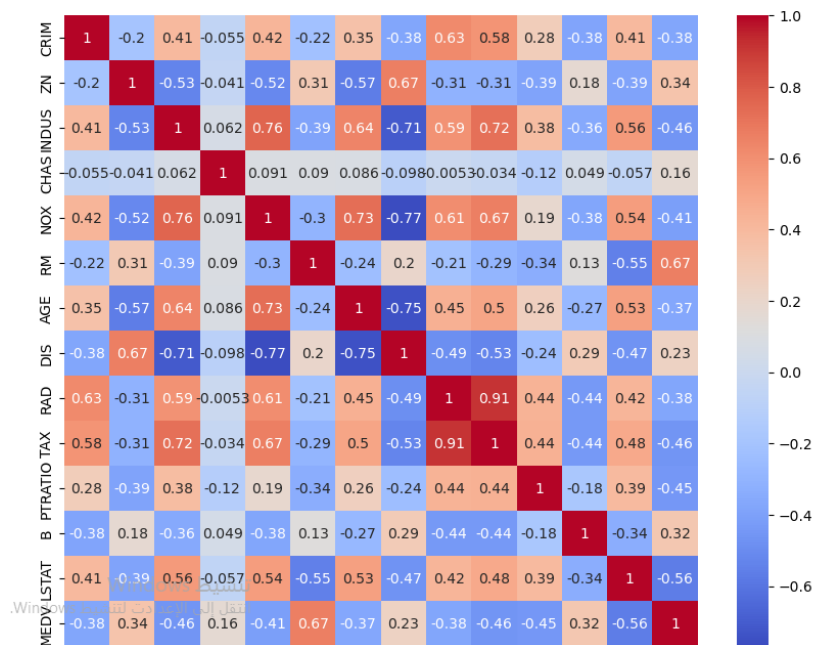




lets check the correlation between feature

```
In [7]: # Lets check the correlation between feature
import seaborn as sns
plt.figure(figsize=(10,8))
sns.heatmap(df.corr(),cmap="coolwarm",annot=True)
```

Out[7]: <Axes: >



Data Clean

```
In [ ]: # Data Clean
        #we have null or nan values we need to clean

In [12]: X = df.drop(['MEDV'],axis=1).values
        y = df['MEDV'].values

In [18]: from sklearn.model_selection import train_test_split
        X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.15)

In [19]: from sklearn.linear_model import LinearRegression
        from sklearn.tree import DecisionTreeRegressor
        from sklearn.svm import SVR
        from sklearn.metrics import r2_score
```

2) Extract necessary features

```
In [21]: from sklearn.preprocessing import MinMaxScaler
# we will with min max preprocessing for dataset
scaler = MinMaxScaler(feature_range=(0, 1))

In [22]: df.head()
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2

```
In [23]: X=df.drop(['MEDV'], axis=1)
rescaleddata = scaler.fit_transform(X)

In [47]: newdf = pd.DataFrame(rescaleddata)

In [48]: newdf.head()
```

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0.000000	0.18	0.067815	0.0	0.314815	0.577505	0.641607	0.269203	0.000000	0.208015	0.259615	1.000000	0.043759
1	0.000236	0.00	0.242302	0.0	0.172840	0.547998	0.782698	0.348962	0.043478	0.104962	0.500000	1.000000	0.099771
2	0.000236	0.00	0.242302	0.0	0.172840	0.694386	0.599382	0.348962	0.043478	0.104962	0.500000	0.989737	0.030968
3	0.000293	0.00	0.063050	0.0	0.150206	0.658555	0.441813	0.448454	0.086957	0.066794	0.586538	0.994276	0.161292
4	0.000705	0.00	0.063050	0.0	0.150206	0.687105	0.528321	0.448454	0.086957	0.066794	0.586538	1.000000	0.048472

[illegible]

check for missing values

```
In [51]: # some missing values need to be handle
newdf.isnull().values.any()
```

```
Out[51]: False
```

```
In [52]: newdf.fillna(0, inplace = True)
```

```
In [53]: newdf.isnull().values.any()
#### no nan values
```

```
Out[53]: False
```

```
In [54]: X=newdf.values
y=df['MEDV'].values
```

3) Build a model

```
In [55]: # create regression model

In [56]: #we have to split the dataset into training and testing
# to do so
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.25)

In [57]: from sklearn.linear_model import LinearRegression

In [58]: model = LinearRegression()

In [59]: model=model.fit(x_train, y_train)

In [60]: r_squared = model.score(x_test, y_test)

In [61]: y_pred=model.predict(x_test)

In [62]: print(r_squared)
0.667741871784163
```

4) Predict the asked queries (mentioned in the project) using training model.

```
In [59]: model=model.fit(x_train, y_train)

In [60]: r_squared = model.score(x_test, y_test)

In [61]: y_pred=model.predict(x_test)

In [62]: print(r_squared)
0.667741871784163
```


5) Evaluate model performance (Like using accuracy, Precision, Recall, F1 Score etc.)

```
In [63]: # Model Evaluation

In [64]: from sklearn.model_selection import cross_val_score
import numpy as np
scores = cross_val_score(model, x_train, y_train, cv=5)
print('Accuracies: %s' % scores)
print('Mean accuracy: %s' % np.mean(scores))

Accuracies: [0.64375278 0.13698322 0.21094195 0.73546633 0.62427374]
Mean accuracy: 0.47028360186702844
```

The decision tree used in linear regression:

```
In [67]: #Decision Tree
from sklearn.tree import DecisionTreeRegressor

In [68]: regr_2 = DecisionTreeRegressor(max_depth=5)

In [69]: regr_2.fit(x_train, y_train)

Out[69]: DecisionTreeRegressor
DecisionTreeRegressor(max_depth=5)
```

Conclusion:

```
In [70]: #conclusion
cores = cross_val_score(regr_2, x_train, y_train, cv=5)
print('Accuracies: %s' % scores)
print('Mean accuracy: %s' % np.mean(scores))

Accuracies: [0.64375278 0.13698322 0.21094195 0.73546633 0.62427374]
Mean accuracy: 0.47028360186702844

In [71]: #overall of mean accuracy for models
print('Linear regression accuracy', np.mean(scores), 'Decision tree regression accuracy', np.mean(scores))

Linear regression accuracy 0.47028360186702844 Decision tree regression accuracy 0.47028360186702844
```

Conclusion:

the project aimed to predict the median value of owner-occupied homes in Boston using a regression analysis. The dataset used was fairly small with 506 rows and 14 columns, making it easy to apply various techniques without worrying about memory constraints. Through the analysis, we were able to identify key factors that influence the median value of homes in Boston, such as crime rate, pupil-teacher ratio, and accessibility to highways. Overall, this project provides valuable insights into the real estate industry in Boston and can be used as a basis for further research in this field.

References:

1- [Lab Manual](#)

- 2- <https://www.cs.toronto.edu/~dave/data/boston/bostonDetail.html>
- 3- <https://www.analyticsvidhya.com/blog/2015/11/started-machine-learning-ms-excel-xl-miner/>
- 4- <https://corporatefinanceinstitute.com/resources/data-science/regression-analysis/>

Appendix: Program Code File



Machine Learning project.html



Machine Learning project.ipynb

has been added the code file in another file