به نام خدا

حسین محمدخانی مستندات پروژه ی پیش بینی قیمت ملک

جمع آوری دیتاست :

ابتدا لینک مربوط به محله ی موردنظر در سایت شیپور توسط chromedriver باز می‌شود. تا زمانی که بخش "آگهی های اطراف" مشاهده نشده است ، اسکرول اتوماتیک انجام می‌شود و لینک تک به تک آگهی های موجود در صفحه ، داخل آرایه ی collected_links قرار داده می‌شود.

سپس پیمایش آرایه ی collected_links آغاز می‌شود. هر یک از لینک های جمع آوری شده آگهی مربوط به یک خانه است. حلقه یک به یک وارد لینک ها می‌شود ، فیچر ها را جمع آوری می‌کند و به آرایه ی مربوط به هر فیچر (meterages_list , rooms_number list ,...) اضافه می‌کند. اگر آگهی مربوط به شهر تهران نباشد یا مربوط به محله ی مورد نظر نباشد (آگهی های فوری مربوط به شهر ها یا محله های دیگر) ، به دیتاست اضافه نخواهد شد. همچنین آگهی هایی که فیچر های موردنظر را ندارند به دیتاست افزوده نمی‌شوند.

در پایان با استفاده از کتابخانه pandas ، آرایه ی هر فیچر وارد یک فایل csv خواهد شد

ماشین لرنینگ :

ابتدا فایل csv دیتاست که در فاز اول پروژه ساخته شد ، طبق ورودی های کاربر ، خوانده می‌شود و اطلاعات مربوط به 6 فیچر (متراژ ، تعداد اتاق ، پارکینگ ، انباری ، آسانسور و سال ساخت) و اطلاعات مربوط به Label (قیمت هر خانه) به ترتیب در متغیر های x و y ذخیره می‌شوند.

سپس 80٪ داده ها برای Train کردن مدل و 20٪ مابقی برای Test کردن مدل استفاده می‌شود. (برای ترین و تست بهتر داده ها ، به ترتیب از دو تابع fit_transform و transform برای استاندارد کردن داده ها استفاده شده است). تابع train_test_split ، هر بار مقادیر train و test را به صورت تصادفی انتخاب می‌کند که برای تولید نتیجه مشکل ساز است. بنابراین از random_state که مقدار آن دلخواه است استفاده می‌کنیم تا هر بار که کد اجرا می‌شود نتایج مشابه بگیریم.

تابع evaluate_model برای بررسی عملکرد مدل روی دیتاست استفاده شده است. به طوری که واریانس و RMSE هر الگوریتم را نمایش می‌دهد و نمودار مقایسه ی قیمت پیش بینی شده و قیمت واقعی را رسم می‌کند.

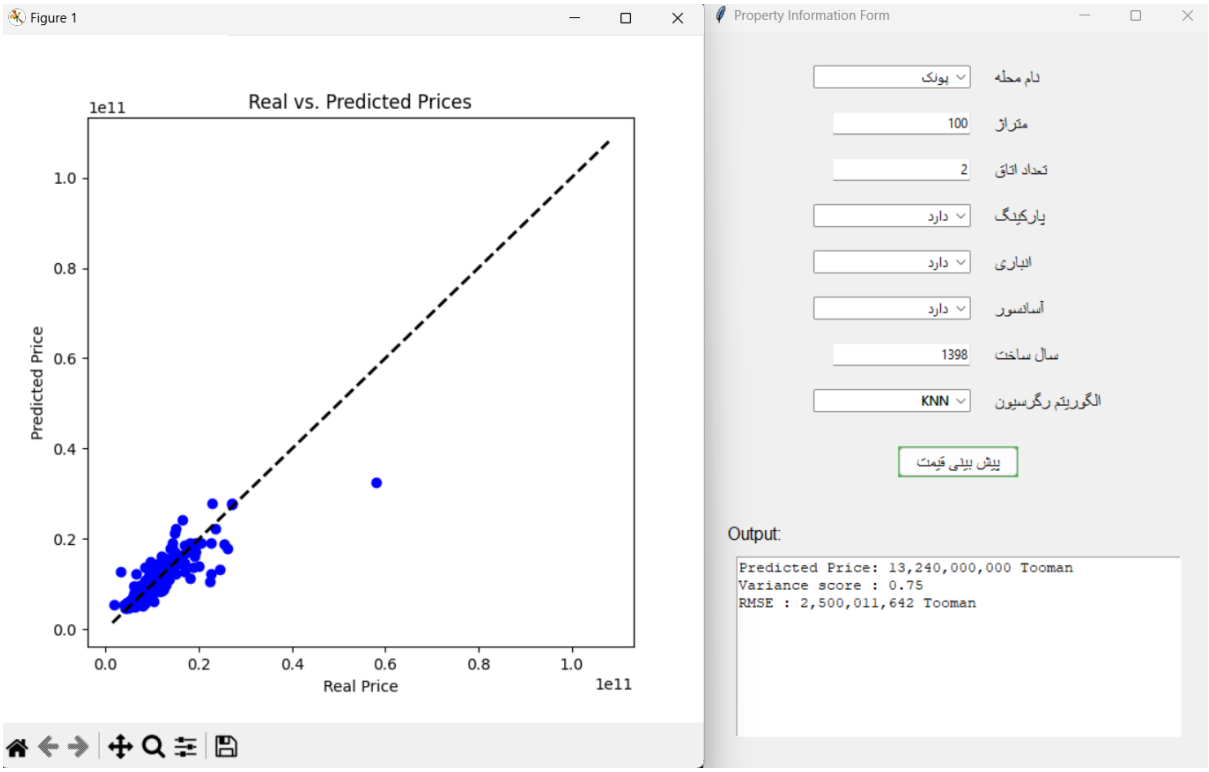
تابع knn_algorithm : از الگوریتم knn برای پیش‌بینی قیمت بر اساس ورودی های کاربر استفاده می‌کند. 5 نقطه ی نزدیک به مقدار ورودی کاربر را روی نمودار پیدا می‌کند و بر اساس آن ها قیمت خانه ی مورد انتظار کاربر را پیش‌بینی می‌کند.

تابع linear_regression_algorithm : از رگرسیون خطی چندگانه استفاده می‌کند برای بدست آوردن بهترین coefficient و intercept در فرمول میان نقاط موجود روی نمودار

تابع gaussain_NB : از فرمول gaussian naïve bayes استفاده می‌کند تا بر اساس فیچر ها ، قیمت را بر اساس ورودی مدنظر کاربر پیش‌بینی کند.

تابع decision_tree_algorithm : از الگوریتم درخت تصمیم برای پیش بینی قیمت استفاده می‌کند. به طوری که ابتدا دیتافریم مربوط به فیچر ها را در دیتاست رصد می‌کند سپس یک مدل در ساختار یک درخت train می‌کند تا قیمت نهایی را طبق ورودی پیش بینی کند

خروجی نمونه از نرم افزار :



پس از پیش بینی قیمت ، مدل ماشین لرنینگ به دو صورت نموداری و عددی Evaluate می‌شوند. نقطه های روی نمودار حین پروسه ی train روی 80٪ داده ها، مختصات مربوط به قیمت واقعی و قیمت پیش بینی شده را نشان می‌دهند. خط روی نمودار نیز فاصله ی بین کمترین و بیشترین قیمت در دیتاست را نشان می دهد. هر چقدر نقاط به خط رسم شده نزدیک تر باشند یعنی الگوریتم بهتر Train شده و قیمت پیش بینی شده به قیمت واقعی نزدیک تر است. اعداد مربوط به variance score و RMSE به صورت عددی بیان می‌کنند که الگوریتم تا چه حد خوب عمل کرده است. (variance score نزدیک به 1 و RMSE کمتر نشان دهنده ی دقت بالاتر مدل هستند.)