



Machine Learning Project

Apartment price
prediction



Rozhkov Denis 213
Galkin Ivan 211
Babaev Daniil 211
Dolgopolova Alina 211

Our Solution

QR code with the repository of our
solution



Aim of our Project

To **predict** the cost of apartments in Moscow and the region using the given dataset

To do this we:

- Analyze given dataset
- Build a machine learning model
- Search for correlations and insights from the analysis of the generated diagrams
- Apply it to a test sample



Our Dataset

	Type	Number of rooms	Area	Floor parameter	Number of floors	Building	Repair	Time to metro	Region	District	Name of residential complex	Price
0	Квартира	2	70.03	0.117647	17	Новостройка	Без ремонта	48	Москва	НАО (Новомосковский)	ЖК «Белая звезда»	7073030
1	Квартира	3	76.47	0.647059	17	Новостройка	Без ремонта	48	Москва	НАО (Новомосковский)	ЖК «Белая звезда»	7799940
2	Квартира	2	60.00	0.250000	20	Новостройка	Без ремонта	12	Москва	СЗАО	ЖК «Хорошевский»	11700000
3	Квартира	2	65.20	0.400000	5	Новостройка	Без ремонта	500	Красногорск	Красногорск городской округ	ЖК «Рублевское предместье»	6055000
4	Квартира	1	38.18	0.333333	3	Новостройка	Без ремонта	132	Истра	Истра городской округ	ЖК «Павловский квартал»	2710780
...
29039	Квартира	2	62.50	1.000000	3	Новостройка	Без ремонта	500	Истра	Истра городской округ	ЖК «Павловский квартал»	4312500
29040	Квартира	3	80.40	1.000000	3	Новостройка	Без ремонта	92	Истра	Истра городской округ	ЖК «Павловский квартал»	5102860
29041	Квартира	3	84.00	1.000000	3	Новостройка	Без ремонта	500	Истра	Истра городской округ	ЖК «Павловский квартал»	5376000
29042	Квартира	3	104.90	0.500000	20	Новостройка	Без ремонта	500	Балашихинский	Балашиха	ЖК «Эдельвейс-Комфорт»	7000000
29043	Квартира	2	72.71	0.333333	3	Новостройка	Без ремонта	92	Истра	Истра городской округ	ЖК «Павловский квартал»	4144470

29044 rows × 12 columns

K-nearest neighbour Algorithm

KNN

```
1 X = prep_df.drop('Price', axis=1)
2 y = prep_df['Price'].values
3
4 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
5
6 scaler = MinMaxScaler()
7 X_train_scaled = scaler.fit_transform(X_train)
8 X_test_scaled = scaler.transform(X_test)
9
10 knn = KNeighborsRegressor(n_neighbors=3)
11 knn.fit(X_train_scaled, y_train)
12
13 y_pred = knn.predict(X_test_scaled)
14
15 rmse = np.sqrt(mean_squared_error(y_test, y_pred))
16 print("RMSE:", rmse)
17 r2_score = r2_score(y_test, y_pred)
18 print("r2_score:", r2_score)
```

RMSE: 2548343.098887824

r2_score: 0.8080678921442257

Linear Regression

Linear Regression

```
1 linear_reg = LinearRegression()  
2 linear_reg.fit(X_train_scaled, y_train)  
3  
4 y_pred = linear_reg.predict(X_test_scaled)  
5  
6 rmse = np.sqrt(mean_squared_error(y_test, y_pred))  
7 print("RMSE:", rmse)
```

RMSE: 3889526.900627334

Random Forest

Random Forest

```
1 # X = new_df.drop("Price", axis=1).values
2 # y = new_df["Price"].values
3 # X = new_df.drop("Price for m2", axis=1).values
4 # y = new_df["Price for m2"].values
5
6 model = RandomForestRegressor(n_estimators=100)
7 model.fit(X_train, y_train)
8
9 y_pred = model.predict(X_test)
10 # new_df['Predicted Price for m2'] = ([0] * (len(new_df) - len(y_pred))) + list(y_pred)
11
12 rmse = np.sqrt(mean_squared_error(y_test, y_pred))
13 print("RMSE:", rmse)
14 rmsle = np.sqrt(mean_squared_log_error(y_test, y_pred))
15 print("RMSLE:", rmsle)
```

RMSE: 1407481.5118087337

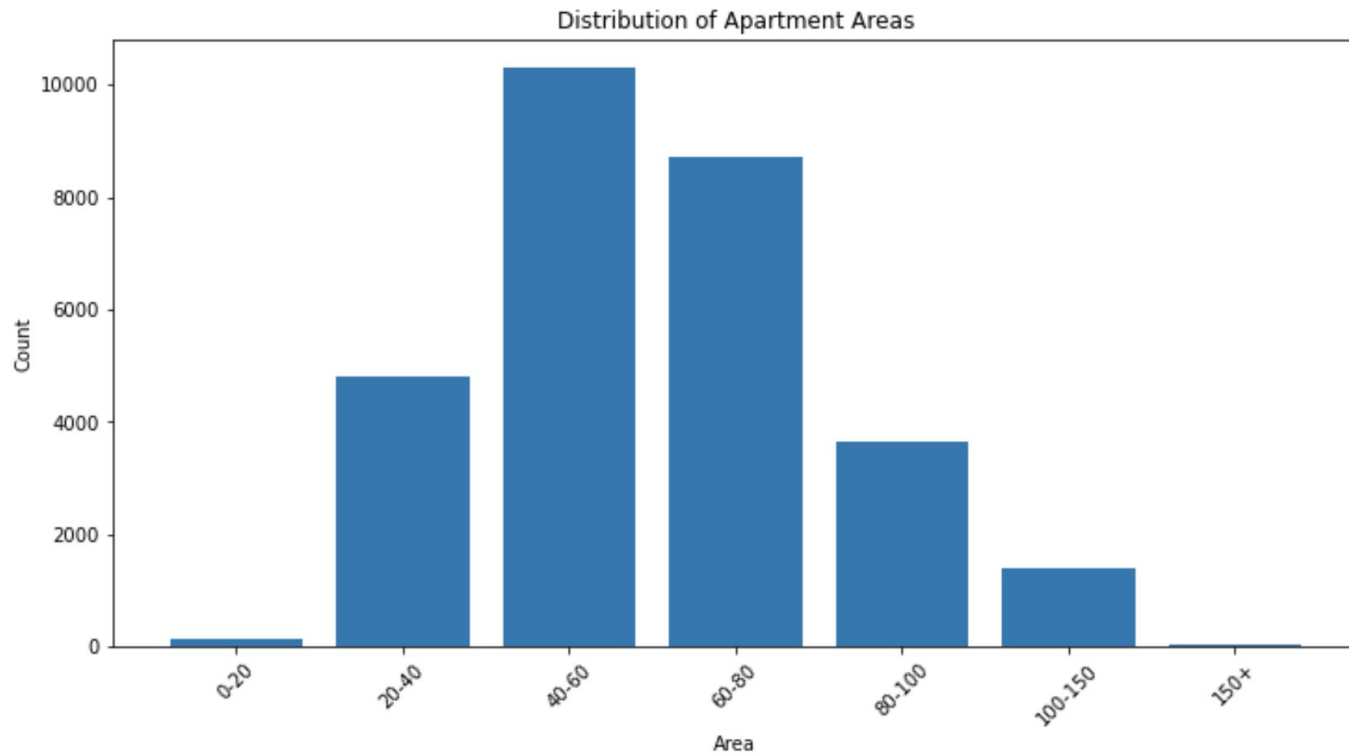
RMSLE: 0.10052839931429551

Data Analysis

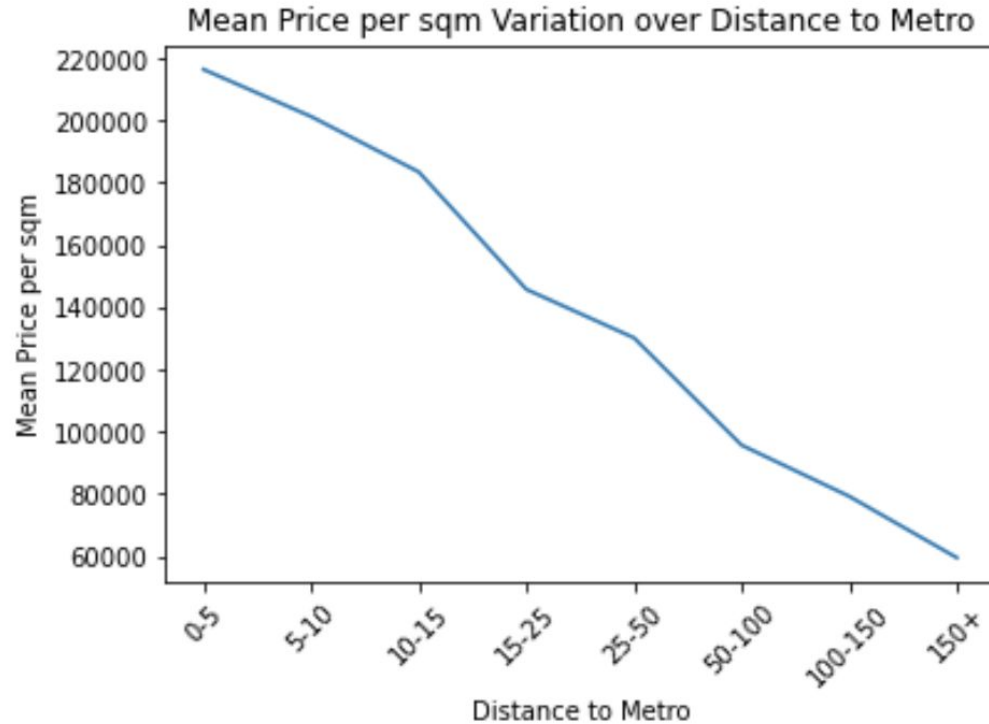
Distribution of Apartment Types



Data Analysis

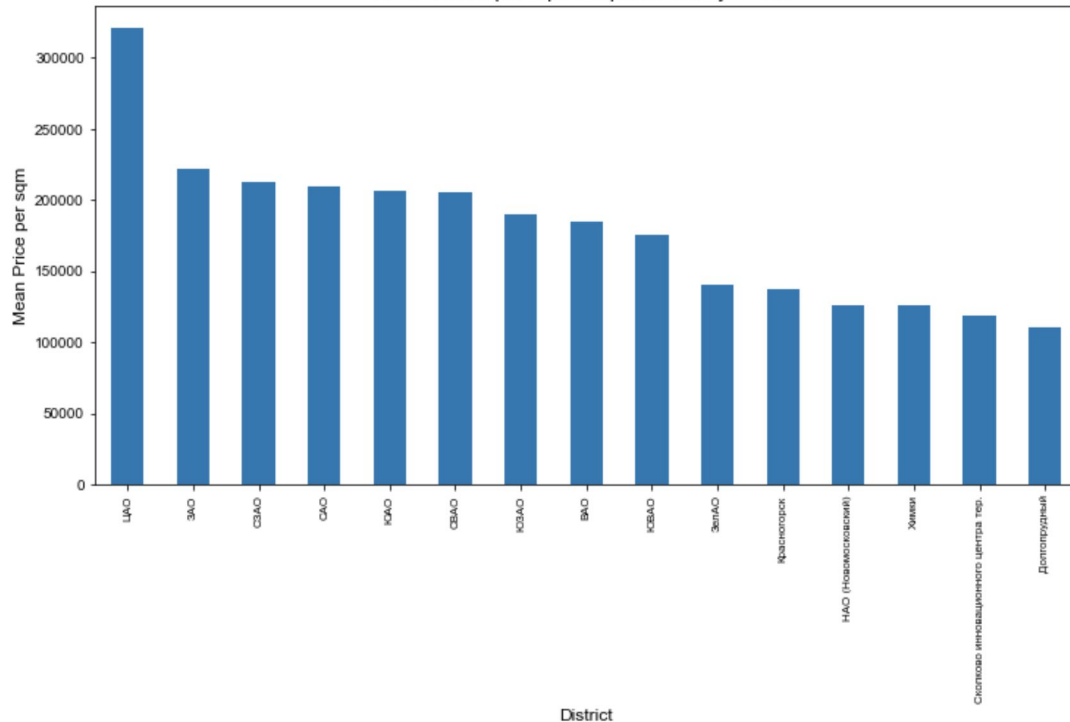


Data Analysis



Data Analysis

Mean Prices per sqm of Apartments by District



Thank You For Attention!

