**House Price Prediction**

**Group Name - Data Ninjas**

**Group Members**

1. Rajvi Shah
2. Yash Kamtekar
3. Sarjak Patel

**Project Detail**

Our project has been selected to answer the following questions given below

1. What houses should I invest in?
2. What is the price trend in real estate?
3. Which properties are over-priced/under-priced currently in the market?
4. How are real-estate prices affecting the economy?
5. Which factors are most/least affecting the house price?
6. How will properties perform in the future? (based on market's & property's past performance)
7. How will the market grow on a quarterly & annual basis?
8. Which area/region is best/worst for investment?

The features of our data set and what each value means has been stored in a [text file](https://drive.google.com/file/d/1zfk2aiL7t2JXRhHkJQGx6g4ICD5XjWFj/view?usp=sharing).

We have created a dashboard where the user can select all the features from a list and select a target feature. On this basis our model will predict the value of the target selected by the user.

On our dataset we have applied **clustering techniques** such as

1. K Means
2. Hierarchical clustering
3. DBscan clustering
4. Gaussian clustering
5. Agglomerative clustering
6. Mean shift clustering
7. Spectral clustering
8. Birch clustering

**Classification techniques**

1. Decision Tree
2. Random Forest
3. Logistic Regression
4. Support Vector Machines
5. Naive Bayes
6. k Nearest Neighbors"

We tried to implement classification algorithms on multi-label features.

* Instead of keeping numerical values of features - OverallQual, we can convert it to a binary classification problem. How? OverallQual > 5 -> Good quality; OverallQual < 5 -> Not Good quality.
* This can help in getting house quality by entering some features like Sales Price. Assuming the user enters the house budget and location, he/she can get an idea about the house quality.

**Regression models**

1. MLP Regressor
2. Random Forest Regressor
3. KNeighborsRegressor
4. Gradient Boosting Regressor
5. XGBoost Regressor

**Upsampling of data**

Upsampling or Oversampling refers to the technique to create artificial or duplicate data points or of the minority class sample to balance the class label. There are various oversampling techniques that can be used to create artificial data points.

**Downsampling of data**

Downsampling or Undersampling refers to removing or reducing the majority of class samples to balance the class label. There are various undersampling techniques implemented in the imblearn package including:

* Random Under Sampling
* Tomek Links
* NearMiss Sampling
* ENN (Edited Nearest Neighbors)

Data imbalance is predominant and inherent in the real world. Data often demonstrates skewed distributions with a long tail. However, most of the machine learning algorithms currently in use were designed around the assumption of a uniform distribution over each target category (classification).

**SMOTE** stands for Synthetic Minority Over-Sampling Technique. SMOTE is performing the same basic task as basic resampling (creating new data points for the minority class) but instead of simply duplicating observations, it creates new observations along the lines of a randomly chosen point and its nearest neighbors. Basically you’re simulating some additional variation in the data (within the established bounds of your minority class), reducing the danger of overfitting (although not eliminating it).

**Advantages**

* It improves the overfitting caused by random oversampling as synthetic examples are generated rather than a copy of existing examples.
* No loss of information.
* It’s simple.

**Disadvantages**

* While generating synthetic examples, SMOTE does not take into consideration neighboring examples that can be from other classes. This can increase the overlapping of classes and can introduce additional noise.
* SMOTE is not very practical for high-dimensional data.

**Smogn**

The key idea of SMOGN algorithm is to combine both SMOTER and Gaussian Noise strategies for generating synthetic examples to simultaneously limit the risks that SMOTER can incur such as lack of diverse examples by using the more conservative strategy of introducing Gaussian Noise because SMOTER will not use the most distant examples in the interpolation process. It works by generating new synthetic examples with SMOTER only when the seed example and the k-nearest neighbor selected are close enough and using the Gaussian noise when the two examples are more distant

The top algorithms for our data set are

1. **KNN**

The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems.

The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.

1. **Random Forest Regressor**

The Decision Tree is an easily understood and interpreted algorithm and hence a single tree may not be enough for the model to learn the features from it. On the other hand, Random Forest is also a “Tree”-based algorithm that uses the qualities features of multiple Decision Trees for making decisions.

Therefore, it can be referred to as a ‘Forest’ of trees and hence the name “Random Forest”. The term ‘Random’ is due to the fact that this algorithm is a forest of ‘Randomly created Decision Trees’.

The Decision Tree algorithm has a major disadvantage in that it causes over-fitting. This problem can be limited by implementing the Random Forest Regression in place of the Decision Tree Regression. Additionally, the Random Forest algorithm is also very fast and robust than other regression models.

1. **Linear Regression**

Simple linear regression is useful for finding relationships between two continuous variables. One is a predictor or independent variable and the other is a response or dependent variable. It looks for statistical relationships but not deterministic relationships. Relationship between two variables is said to be deterministic if one variable can be accurately expressed by the other. For example, using temperature in degrees Celsius it is possible to accurately predict Fahrenheit. Statistical relationship is not accurate in determining relationship between two variables.

The core idea is to obtain a line that best fits the data. The best fit line is the one for which total prediction errors (all data points) are as small as possible. Error is the distance between the point to the regression line.

**Best Algorithm is K nearest neighbor Regression**