Analysis and Prediction of house price, Loan default and Bank churn using a machine learning model

Ayusha Eknath Kashilkar
MSC in Data Analytics
National College of Ireland
Dublin, Ireland
x20239343@student.ncirl.ie

***Abstract*— Houses Prices has been changing and mostly increasing day by day. Some locations are hyped which causes the house prices to increase without even the main factors affecting it. The house price should be decided on the main factors which could be done by prediction. The income of the financial institutes comes from the loan interest and EMI paid by the customers. Many loans are approved on high interests and without securities provided to bank. With the use of machine learning model we can predict the loan defaulters by customer related data and be provided to bank for use of loan approvals. Bank churn has been a major issue to financial banks/institute. The bank and customer have no contractual agreement on period of time. The customer may leave the bank for another which might provide good interest rate, services and benefits. This affects bank, as they could predict and retain the customer before the customer could leave and provide a better service to the loyal customers.**

Keywords—prediction, Analysis, Machine Learning, KNN, Logistic regression

I. INTRODUCTION

Data is been generated every day, through machine learning models the data can be transformed to analyse research questions and major daily questions. Through machine learning model, predictions can be made via data provided to resolve a question or create insights of predictions. In this research paper, 2 Regression model – Linear Regression and Decision Tree Regression is used and 3 classification model – Random Forest, KNN and Logistic Regression.

As for the dataset we have considered the Beijing, China as our location of houses, which we will be predicting the price for. Real estate has never been transparent industry in the world. Many factors related to houses were never considered in pricing , rather some time the hyped area, high rates affects the pricing of the house. The basic parameters like square foot, floor, kitchen, drawing rooms, construction time etc is used in this paper to predict the price.

Banks providing the loans and lending business has been a great promotion to financial institutes and economy. As great it is, there has been issues with loan lending and loan defaulters of bank. Bank major gains from loan EMI and Interest paid on loan, If there are loan defaults it causes financial issue to banks. Using machine learning to predict the defaulter the bank can save time and workload and increase accuracy and resolve this issue. The model used are KNN, Random forest classifier and Logistic regression.

Financial Sector is also having problems due to Bank Churn. Many financial banks have provided great offers to customer and new joining customers, which makes the customer to leave the current bank and move assets to another bank which provides good interest rate, services and benefits while joining. To predict using machine learning algorithm if customer would exit the bank or not, this would help bank retain the customer which are about to leave and also come up with better beneficiary plans for customer to retain and provide services

A *RESEARCH QUESTION*

1. What is the correct price of the house effected by multiple parameters?

2. Will the customer default on loan or not?

3. Is the customer going to leave the bank or not?

II. RELATED WORK

The author [1] in this paper have used classification models to forecast the loan defaulters. The author has used CatBoost classifier to get highest accuracy and also Gradient boosting has same accuracy. The results were calculated via precision and F1 score of the model where logistic regression has performed extremely low. In this paper [2] using the Azure ML platform , two machine learning algorithm were used – Decision Jungle and Decision Forest for predicting the loan borrowers. SMOTE and Bragging methods were applied on the data ste to handle the missing and imbalanced data. Data discretization is used for data pre-processing. The Author of the paper [3] has implement and compared 3 algorithms i.e Decision Tree, Neural Networks and Logistic Regression using OptiML algorithm from BigML. As logistic Regression gives most accuracy value compared to others. The results have been calculated on F1 score, recall and precision of models. The author [4] has used 3 machine learning algorithm – Naïve Bayes, Decision tree and random forest to predict loan default. All the algorithm were runed 4 times, All the iterations used different feature selections and one of them used unprocessed dataThe author [5] has predicted the budget of the houses as the focus is on efficient pricing of houses and real estates. They have analysed the previous trends of industry and ranges of prices to predict the future price. The algorithms applied were logistic regression, Random Forest, Naïve Bayes and Decision tree. Where Ada boost and decision tree gives good accuracy than other models. The author [6] confirms that CNN can also be used to perform the time series prediction as the R square is higher than 0.945. The author confirms that for predicting the house prices a five month time series span is suitable. The author [7] has used CNN to predict the house price and to compare the effectiveness GM model and XGBoost model has been used. The parameters were optimised by grid search and results were calculated by accuracy and cross checking method. The authors main focus

was to differentiate features from housing and macroeconomy dataset.The author [8] has used 5 models to predict the house prices in Bengaluru i.e XGBoost, SVR, Lasso, OLS and Ridge regression model. With evaluation matrix the data was compared for study. The data was collected form 2016.The author [9] states when a client is lost the bank is at loss as serving a loyal long ter customer is less costly. With use of Machine learning we can predict the clients who are probably going to churn. The models applied were Decision tree , KNN, LR, RF to predict the customer who is going to leave. Comparision is provided through evaluation metrics like accuracy, recall etc.The author [10] states in this paper used Lightgbm , Random forest , Catboost algorithm to predict if the customer would churn.The author [11] has used XGBoost, LightGBM and CATBoost were applied over the default dataset and then on hyperparameters. Hyperparameters were derived from Grid search. The LightGBM performs the best model in comparison to others with accuracy of 94.8%.The author [12]uses KNN, SVM, Decision Tree and Random Forest models to predict the customer churn in Bank. Feature selection method has been used to verify system performance. The Random forest model after oversampling is better in performance as compared to others

III. DATA MINING METHODOLOGY

KDD is an iterative process, in which evaluation can be improved , mining can be optimized and new data can be combined and transformed in order to get new results.

A. Data Sources and Understanding

The first data source (The House Price Prediction dataset) consists of 24 columns and 318819 rows including dependent variable. This dataset contains information on House price in Beijing such as square, construction time, renovation, building structure, bedrooms, kitchen etc. It has 24 columns and 318819 rows represented in Table 1.

The dependent variable in the dataset is price which contains the price of house according to the mentioned features

TABLE I. HOUSE PRICE PREDICTION

Column name	***Description***
Lng	Lng longitude
Lat	latitude
CID	community id
Tradetime	the time of transaction
DOM	active days on market
Followers	the number of people follow the transaction
totalPrice	the total price
Price	the average price by square
Square	the square of house
livingRoom	the number of living room
Drawingroom	the number of drawing room
Kitchen	the number of kitchen
bathroom	the number of bathroom
Floor	the height of the house
buildingType	including tower(1) , bungalow(2)combination of plate and tower(3), plate(4).
constructionTime	the time of construction
renovation Cndition	including other(1), rough(2),Simplicity(3), hardcover(4)
buildingStructure	including unknow(1), mixed(2), brick and wood(3), brick and concrete(4),steel(5) and steel-concrete composite (6)
Ladder ratio	the proportion between number of residents on the same floor and number of elevator of ladder. It describes how many ladders a resident have on average.
elevator	have (1) or not have elevator(0)
Five year property	if the owner have the property for less than 5 years,
Subway	If subway is there or not . 1 – yes , 0 – no
District	The district of house
Community Average	Community average score

The second data set is Loan default to predict if the customer would default on loan or not. Th dataset contains 13 columns and 252000 rows, as shown in Table II with column names and description. The dataset contains information about the customer such as age, experience, married or single, house, car ownership, profession, state etc including the dependent variable risk flag

TABLE II. LOAN DEFAULT PREDICTION

Column name	***Description***
ID	ID
Income	Income of the customer
Age	Age of customer
Experience	Experience of the customer
Married /Single	Married or single status
House_Ownership	Type of house - rented or owned
Car_Ownership	If the customer has car or not
Profession	Customer profession
CITY	City currently living
State	Current state of customer
Current Job Yrs	job years in current company
Current House Yrs	Number of year customer has lived in current house
Risk Flag	If the customer would default or not

The Third dataset is of Bank Churn Dataset (Customer would leave or not) , it contains 14 columns and 10000 rows as shown in Table III. The dataset includes information of customer like CustomerID, surname, creditscroe, geography, Gender, Age, Tenure., Balance, NumOf Products,etc also included the dependent variable.

TABLE III. BANK CHURN PREDICTION

Column name	***Description***
RowNumber	Row Numbers from 1 to 10000
Surname	Customer's last name
CreditScore	Credit score of the customer
Geography	The country from which the customer belongs
Gender	Male or Female
Age	Age of the customer
Tenure	Number of years for which the customer has been with the bank
Balance	Bank balance of the customer
NumOfProducts	Number of bank products the customer is utilising
HasCrCard	Binary Flag for whether the customer holds a credit card with the bank or not
IsActiveMember	Binary Flag for whether the customer is an active member with the bank or not
EstimatedSalary	Estimated salary of the customer in Dollars
Exited	Binary flag 1 if the customer closed account with bank and 0 if the customer is retained

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

B. *Data Pre-Processing*

Data preprocessing is a crucial step in data mining process. Misleading results might be produced from analyzing data which is not processed first. The data from any data source contains missing values, null values, misleading data and impossible data combinations. Before we could apply the machine learning model , the data is cleaned and processed to make the a fit for the model.

Dataset 1 – House Price Prediction

- The dataset contained 318819 rows and 26 columns.
- The data was loaded into Python using the pandas read_csv function and 3 columns were dropped as they are not required i.e url, tradeTime and id, so we have 24 columns and 318819 rows.
- Starting with data cleaning, is to check the null values in the dataset. With isnull() function in python we had checked the null values present and used sum() to get the total count of null values. There were 4 columns containing null values i.e DOM, Building Type, ConstructionTime and communityAverage as shown in Fig 1

```
In [27]: datas.isnull().sum()
Out[27]: Lng                    0
         Lat                    0
         Cid                    0
         tradeTime              0
         DOM               157970
         followers              0
         totalPrice             0
         square                 0
         livingRoom             0
         drawingRoom            0
         kitchen                0
         bathRoom               0
         floor                  0
         buildingType        2021
         constructionTime   19283
         renovationCondition    0
         buildingStructure      0
         ladderRatio            0
         elevator               0
         fiveYearsProperty      0
         subway                 0
         district               0
         communityAverage     463
         price                  0
         dtype: int64
```

Fig 1 : Null Values of Dataset 1

- To fill in Null Values, the fillna() function was used and mean values of the column were filled in places of null. This was done only for DOM, CommunityAverage and building Type, as Building construction time could not be calculated as mean of all the time. After this only construction time contained 19283 null values which were dropped by using dropna() function. All the null values were removed after data cleaning.
- For feature selection of data, corrwith() function was used with method pearson to find the correlation between the dependent variable and independent variables. As price being the value 1 , the other highly dependent variables were removed. Features with greater than value 0.2 were followers, total price, construction time , renovation condition , subway and community average

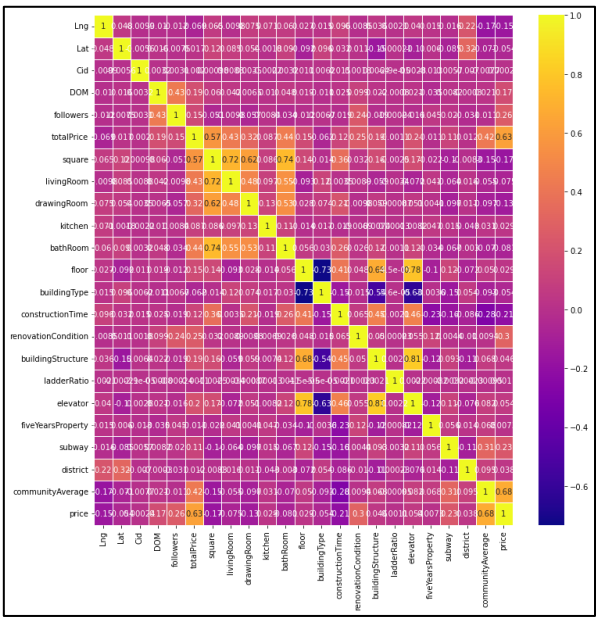

Fig 2 Correlation matrix - House price attributes

```
In [17]: cor = datas.corr()
         cor_target = abs(cor["price"])
         features = cor_target[cor_target>0.2]
         print(features)

followers              0.258917
totalPrice             0.626717
constructionTime       0.210991
renovationCondition    0.298168
subway                 0.231634
communityAverage       0.681681
price                  1.000000
Name: price, dtype: float64
```

Fig 3 Correlation values over 0.2 with price

- Box plots were created for features to check the outliers which would affect the results. There were multiple outliers in DOM, followers, TotalPrice, Squares , Community Average, Price.

  To remove the outliers from the data , The InterQuartile Range was used. Interquartile range is calculates the difference between Quartile 3 (75%) and Quartile 1 (25%)

  Formula :-

  Outlier = (Dataset[column] < (Qua1 - 1.5 * IQR) | (Dataset[column] > (Qua3 + 1.5 * IQR)

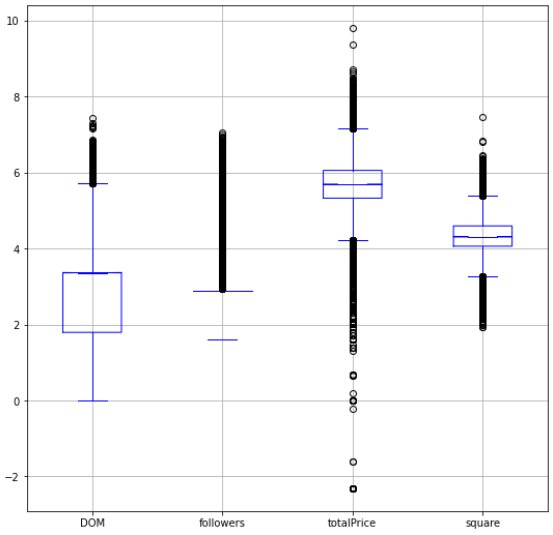

Fig 4 Box plot for checking outliers

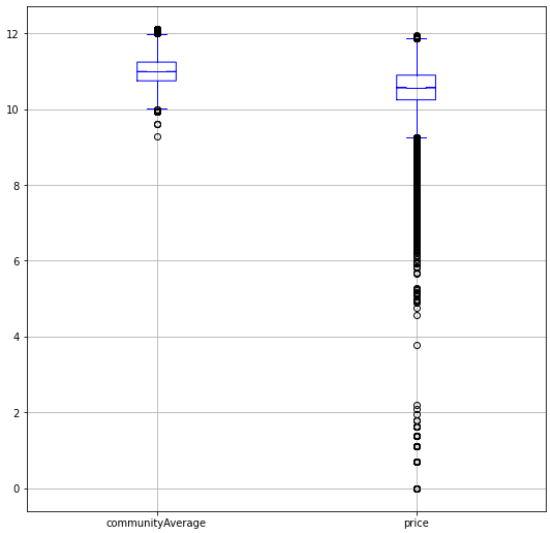

Fig 5 Box plot for checking outliers

Dataset 2 – Loan Default Prediction

- The csv was loaded into dataframe in python using the read_csv function with encoding of cp1252.
- Checked the data set information through info() function for null values and the data type of the columns as shown in Fig .

```
In [10]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 252000 entries, 0 to 251999
Data columns (total 13 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   Id                 252000 non-null  int64
 1   Income             252000 non-null  int64
 2   Age                252000 non-null  int64
 3   Experience         252000 non-null  int64
 4   Married/Single     252000 non-null  object
 5   House_Ownership    252000 non-null  object
 6   Car_Ownership      252000 non-null  object
 7   Profession         252000 non-null  object
 8   CITY               252000 non-null  object
 9   STATE              252000 non-null  object
 10  CURRENT_JOB_YRS    252000 non-null  int64
 11  CURRENT_HOUSE_YRS  252000 non-null  int64
 12  Risk_Flag          252000 non-null  int64
dtypes: int64(7), object(6)
memory usage: 25.0+ MB
```

Fig 6 Dataset information and null values

- There are No null values present in the data columns. The data types present are 7 columns with int64 and 6 columns with object datatype
- Checked the number of 1's and 0's in dependent variable for overfiiting and underfitting issues. The number of 0's is 221004 and the number of 1's is 30996. SMOTE was used for oversampling issue

```
In [10]: data["Risk_Flag"].value_counts()

Out[10]: 0    221004
         1     30996
         Name: Risk_Flag, dtype: int64
```

Fig 7 Number of 0's and 1's in Dependent variable

- Using the Corr() function , correlation matrix was created to check the relation between the dependent and independent variables

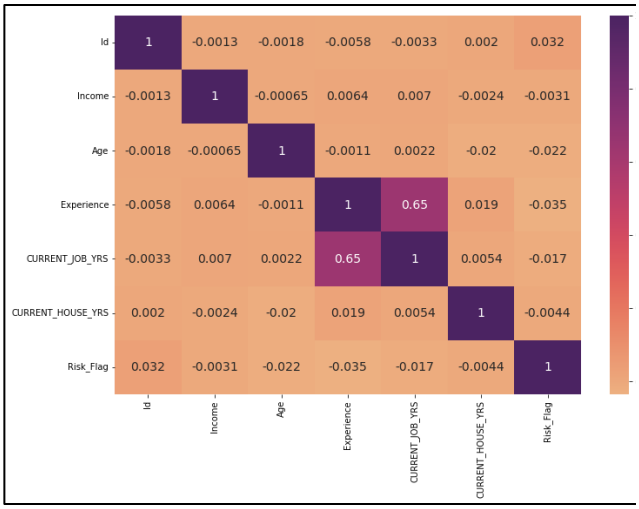

Fig 8 Correlation matrix of Loan default attributes

- There were 3 columns with categorical variable. The Married/ Single and Car_ownership columns were transformed with label encoder into 1 and 0 values respectively. The House_Ownership colum was transformed using Onehotencoder function in Python.
- As Profession , City and State column are not required to analyse the loan default, they were dropped.

Dataset 3 – Bank Churn Prediction

- The dataset was loaded into python through read_csv function. There are total 10000 rows and 14 columns
- The dataset contains no null values. The sum of null values were checked through isnull() function. As shown in fig

```
In [7]: datas.isnull().sum()

Out[7]: RowNumber          0
        CustomerId         0
        Surname            0
        CreditScore        0
        Geography          0
        Gender             0
        Age                0
        Tenure             0
        Balance            0
        NumOfProducts      0
        HasCrCard          0
        IsActiveMember     0
        EstimatedSalary    0
        Exited             0
        dtype: int64
```

Fig 9 Checking null values in Bank churn attributes

- The correlation of independent variables with dependent variable (Exited) was check through corr() function as shown in figv

```
In [12]: mutual_info=datas.corr()["Exited"].sort_values(ascending=False)
         mutual_info

Out[12]: Exited             1.000000
         Age                0.285323
         Balance            0.118533
         EstimatedSalary    0.012097
         HasCrCard         -0.007138
         Tenure            -0.014001
         CreditScore       -0.027094
         NumOfProducts     -0.047820
         IsActiveMember    -0.156128
         Name: Exited, dtype: float64
```

Fig 10 : Correlation values (pearson)

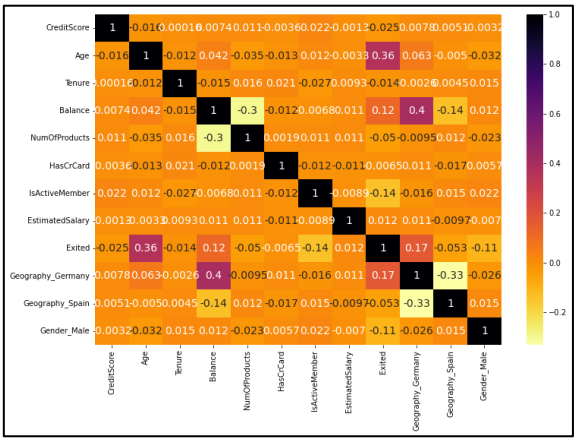

Fig 11 Correlation matrix

- Checked the number of unique values in every column through unique() function.

```
In [10]: datas.nunique()

Out[10]: RowNumber          10000
         CustomerId         10000
         Surname             2932
         CreditScore          460
         Geography              3
         Gender                 2
         Age                   70
         Tenure                11
         Balance             6382
         NumOfProducts          4
         HasCrCard              2
         IsActiveMember         2
         EstimatedSalary     9999
         Exited                 2
         dtype: int64
```

Fig 12 Checking unqiue values in attributes

- Then Checked the number of 1's and 0's in the dependent variable – Exited column . It contains 7963 number of 0's and 2037 number of 1's. SMOTE is used for oversampling.

```
In [14]: datas["Exited"].value_counts()

Out[14]: 0    7963
         1    2037
         Name: Exited, dtype: int64
```

Fig 13 Number of 1's and 0's in Dependent variable

- There are 2 categorical columns i.e Geography and Gender. Dummy variables were created through get_dummy() function and the original columns were dropped.
- The Age column contained multiple outliers, As shown in Fig through box plot. The outliers were removed by IQR(Interquartile Range) creating the lower bound value and upper bound value and mapping the age between the values and dropping the values outside the lower and upper bound

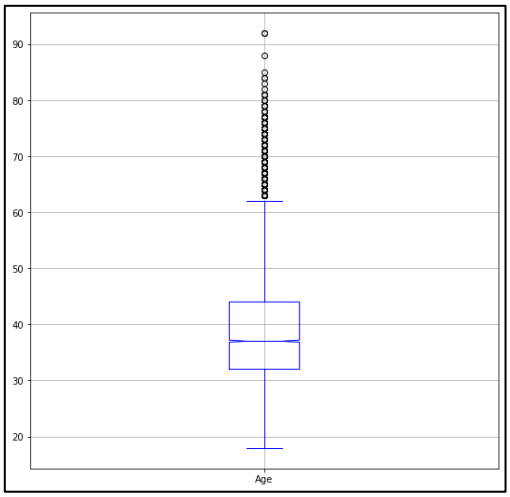

Fig 14 Box plot for checking outliers

IV. DATA MODELING

Dataset 1 – House price prediction

- After data cleaning and data processing the data was divided into 2 dataset i.e X and Y, where X contained independent variables and Y contained dependent variable i.e Price
- Through train_test_split library the data was divided into 4 dataset X_train, Xtest, Ytrain and Ytest. The model will be trained on Xtrain and Ytrain. The test dataset size is 20% of the dataset therefore the train dataset will be 80 % of the dataset
- Linear Regression – The LinearRegression() function was used from sklearn.linear_model in python to apply the Linear Regression model. Normalize was set to true , so X reressors will be normalised before the regression by subtracting the mean and dividing by the l2 norm. Then the model was applied to X_train and y_train by using the fit() function of sklearn.
- Decision Tree Regressor – The Decision tree Regression() function was used from the sklearn.tree library in python to apply to Decision Tree Regressor model. The random state is 0. The model was used on X_train and Y_train. Using the fit() function.

Dataset 2 – Loan Default Prediction

- After data cleaning and data processing the data was divided into 2 dataset i.e X and Y, where X contained independent variables and Y contained dependent variable i.e Risk Flag. Through train_test_spit library the data was divided into 4 dataset X_train,Xtest , Ytraina nd Y test. The data was split 20 % in test and 80 % in train.
- The SMOTE (Synthetic Minority Oversampling Technique) is used to for oversampling classification dataset. The SMOTE function was used from the imblearn library.
- Random Forest Classifier – The RandomeForestClasssifier function was used from the sklearn.ensemble library for model. The parameters passed in models were – the criterion was gini, by default it takes gini, it is for measuring the quality of split. Bootstrap was true, by default it is True – it is used to estimate the generalized score by using the out of bag samples if set to True. The random_state is set to 100. Then the model was fit using fit () function to Xtraina nd y train.
- K Neighbor Classifier – The KNeighborsClassifier function was used from the sklearn.neighbors library for the model. The model paramaeters were n_eightbors set to 5 i.e it is the number of neighbors to use by default for kneighbors queries. The metrics is set to minkowski and p value is 2 which is equivalent to the standard Euclidean metric.
- Logistic Regressor – The LogisticRegression function was used from sklearn.linear function. The model parameter passed was random state value 0. The model was then fit to x_train and y_train dataset.

Dataset 3 – Bank Churn Prediction

- The dataset was divided into dependent variable and independent variable in X and Y variables. These X dataset was then split into x_train and x_test, Y dataset was split into y_train and y_test.
- The StandardScaler function from sklearn.preprocessing was used to scale the X_train and X_test dataset.
- The SMOTE() function from imblearn.over_sampling library was used for oversampling issue. As shown in fig

```
In [25]: from imblearn.over_sampling import SMOTE
         oversampling=SMOTE()
         X_train,y_train=oversampling.fit_resample(X_train,y_train)

         y_train.value_counts()

Out[25]: 0    6105
         1    6105
         Name: Exited, dtype: int64
```

Fig 15 SMOTE - for oversampling

- Logistic Regressor – The LogisticRegression function was used from sklearn.linear function. No parameters were passed, so the default parameters were taken. The model was then fit to x_train and y_train dataset.

- Random Forest Classifier – The RandomForestClassifier() function from the sklearn.ensemble library. No parameters were passed in function, the function took the default parameter values . Then the model was fit using fit() function on X_train , Y_train dataset.
- Using the hyperparameter tunning is used with the GridSearchCV function from the sklearn.model_selection library. The parameters passed were model as randomforestclassifier with min_sample_leaf of 10 and n_jobs is -1. The n_estimators were 100,150,200,250,300,350 and max_depth values were 6,9,12,15,18,21. The cv was set to 10 and scoring was neg_mean_squared_error.
- Then the Grid model was fit on the X_train and Y_train dataset using the fit function. Through the grid.best_params_ we found out the best parameter for the RandomForestClassifier to be passed
- The best parameters were passed through the model and model was applied to the X_test dataset.

V. Model Evaluation

Dataset 1 – House Price Prediction

Regression was performed using Linear Regression and Decision Tree Regression to predict the price of the House in Beijing.

Linear Regression – the R square value of the Linear Regression model with 12 predictors is 0.85646. This means the model accounts for nearly 85.64% variation in dependent variable. The RSME of the model is 9926 and MAE is 8004 The below output shows the predicted values vs the true values price of the house

```
In [240]: df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
          df

Out[240]:
```

	Actual	Predicted
191395	241508.387235	233219.168168
275220	237064.490168	234658.179728
165052	226068.221685	231675.820888
104910	210616.402738	208404.728398
254297	260726.869493	261199.198257
...	...	...
154568	234721.138757	243582.160778
249926	253495.089004	245157.298898
247340	229171.734655	232296.632619
247812	266780.383841	267664.853065
237876	250092.657915	233903.721023

56121 rows × 2 columns

Fig 16 Actual price vs Predicted price of linear regression

Decision Tree Regressor – The R square value of the Decision Tree Regressor with same 12 predictors is 0.9240. that means the accuracy of the model is 92 %. The model has RMSE as 7219 and MAE as 5396The Fig shows the Actual price of the house vs the predicted price of the house.

```
In [248]: df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
          df.head(5)

Out[248]:
```

	Actual	Predicted
191395	241508.387235	233219.168168
275220	237064.490168	234658.179728
165052	226068.221685	231675.820888
104910	210616.402738	208404.728398
254297	260726.869493	261199.198257

Fig 17 Actual price vs predicted price of decision tree

Dataset 2 – Loan Default Prediction

Classification was performed on the Loan default dataset using Random forest, KNN and Logistic Regression. The model was applied to 12 predictors.

Random Forest Classifier – The Accuracy of the model is 88.75 %. The precision of the model is 54.30 and AUC score is 73.72. As shown in Fig shows the confusion matrix and classification report and accuracy

```
[[41395  2806]
 [ 2864  3335]]
              precision    recall  f1-score   support

           0       0.94      0.94      0.94     44201
           1       0.54      0.54      0.54      6199

    accuracy                           0.89     50400
   macro avg       0.74      0.74      0.74     50400
weighted avg       0.89      0.89      0.89     50400

0.8875
```

Fig 18 Confusion matrix and Classification report - Random Forest

KNeightbors Classifier – The Accuracy of the model on Loan default is 0.8876 , which means it is 88.76%. The precision of the model is 0.88 and AUC of 50.0. As shown in Fig

```
[[41866  2335]
 [ 3328  2871]]
              precision    recall  f1-score   support

           0       0.93      0.95      0.94     44201
           1       0.55      0.46      0.50      6199

    accuracy                           0.89     50400
   macro avg       0.74      0.71      0.72     50400
weighted avg       0.88      0.89      0.88     50400

0.8876388888888889
```

Fig 19 Confusion matrix and Classification report - KNN

Logistic Regression – The Accuracy of the model is 0.8770 which is 87.70 %. The weighted average precision is 0.77. As shown in the fig the confusion matrix and classification report of the model

```
[[44201     0]
 [ 6199     0]]
              precision    recall  f1-score   support

           0       0.88      1.00      0.93     44201
           1       0.00      0.00      0.00      6199

    accuracy                           0.88     50400
   macro avg       0.44      0.50      0.47     50400
weighted avg       0.77      0.88      0.82     50400

0.8770039682539682
```

Fig 20 Confusion matrix and Classification report - Logistic Regression

Dataset 3 – Bank Churn Prediction

Classification was performed on The Bank Churn dataset with Logistic Regression and Random Forest Classification.

Logistic Regression – The model has accuracy of 0.72 i.e 72% on the dataset. The weighted average precision is 0.80 and the f1 score for weighted average is 0.74.

The fig shows the confusion matrix and classification report for the logistic regression

```
[[1103  428]
 [ 115  272]]
              precision    recall  f1-score   support

           0       0.91      0.72      0.80      1531
           1       0.39      0.70      0.50       387

    accuracy                           0.72      1918
   macro avg       0.65      0.71      0.65      1918
weighted avg       0.80      0.72      0.74      1918

0.7168925964546402
```

Fig 21 Confusion matrix and classification report - logistic regression

Random Forest Classification –

The accuracy of the model before applying the hyper parameter tunning is 0.7168 which is 71.68% rounding to 72%. The weighted precision is 0.80 and weighted average F1 score is 0.74

```
[[1103  428]
 [ 115  272]]
              precision    recall  f1-score   support

           0       0.91      0.72      0.80      1531
           1       0.39      0.70      0.50       387

    accuracy                           0.72      1918
   macro avg       0.65      0.71      0.65      1918
weighted avg       0.80      0.72      0.74      1918

0.7168925964546402
```

Fig 22 Confusion matrix and classification report - Random Forest before hyperparametric tunning

The Accuracy of the model on Bank Churn dataset after applying the hyperparameter tunning is 0.8248 i.e 82.48%. The weighted average precision is 0.84 and weighted average f1-score is 0.83. The Fig show the Confusion matrix and classification report.

```
[[1318  213]
 [ 123  264]]
              precision    recall  f1-score   support

           0       0.91      0.86      0.89      1531
           1       0.55      0.68      0.61       387

    accuracy                           0.82      1918
   macro avg       0.73      0.77      0.75      1918
weighted avg       0.84      0.82      0.83      1918

0.8248175182481752
```

Fig 23 Confusion matrix and Classification report - Random Forest after hyperparametric tunning

VI. Conclusion

Five machine learning models were successfully built to perform regression and classification to predict the values. As per the results of the dataset 1 – house price prediction – the Decision Tree Regressor outperforms Linear Regression with 92% accuracy.

In dataset 2 – the loan Default classification all three models – Random forest classifier, KNN and Logistic Regression, Both Random Forest Classifier and KNN almost predicted the same output with same accuracy which is better than the accuracy of logistic regression.

In dataset 3 – Bank Churn the two models Logistic Regression and Random Forest Classification, the random forest classification outperforms Logistic regression with the accuracy of 82%.

The accuracy has been increased by applying the hyperparameter tunning

VII. Future Scope

House Price Prediction:

- This study was only limited to Beijing Houses. Data from different cities in China could provide a better understanding of costs of houses over China and which city is better to stay with basic cost
- In Beijing with the location of property, analyses of location factor over price could be done.

Loan Default Classification:

- Multiple factors like job type, medical history, customer tenure etc could be taken into consideration to predict if the customer would default or not.
- With addition to customer, reasons of default can be analyzed with loan amount and period of time.

- Multiple banks loan defaulters' data can be merged to check if the customer was a defaulter in previous or another bank.

Bank Churn Classification:

- Multiple bank data could be taken and predict why customer left the bank i.e for better service, interest rate etc
- Analyses of bank schemes for customer and number of customer left can provide scheme issues and bank to rectify it.

REFERENCES

[1] B. Patel, H. Patil, J. Hembram, and S. Jaswal, "Loan default forecasting using data mining," *2020 International Conference for Emerging Technology, INCET 2020*, Jun. 2020, doi: 10.1109/INCET49848.2020.9154100.

[2] K. Alshouiliy, A. Alghamdi, and D. P. Agrawal, "AzureML based analysis and prediction loan borrowers creditworthy," *Proceedings - 3rd International Conference on Information and Computer Technologies, ICICT 2020*, pp. 302–306, Mar. 2020, doi: 10.1109/ICICT50521.2020.00053.

[3] Z. Ereiz, "Predicting Default Loans Using Machine Learning (OptiML)," *27th Telecommunications Forum, TELFOR 2019*, Nov. 2019, doi: 10.1109/TELFOR48224.2019.8971110.

[4] A. Al-Qerem, G. Al-Naymat, and M. Alhasan, "Loan default prediction model improvement through comprehensive preprocessing and features selection," *Proceedings - 2019 International Arab Conference on Information Technology, ACIT 2019*, pp. 235–240, Dec. 2019, doi: 10.1109/ACIT47987.2019.8991084.

[5] P. Durganjali and M. V. Pujitha, "House Resale Price Prediction Using Classification Algorithms," *6th IEEE International Conference on "Smart Structures and Systems", ICSSS 2019*, Mar. 2019, doi: 10.1109/ICSSS.2019.8882842.

[6] C. Zhan, Z. Wu, Y. Liu, Z. Xie, and W. Chen, "Housing prices prediction with deep learning: An application for the real estate market in Taiwan," *IEEE International Conference on Industrial Informatics (INDIN)*, vol. 2020-July, pp. 719–724, Jul. 2020, doi: 10.1109/INDIN45582.2020.9442244.

[7] Y. Piao, A. Chen, and Z. Shang, "Housing Price Prediction Based on CNN," *9th International Conference on Information Science and Technology, ICIST 2019*, pp. 491–495, Aug. 2019, doi: 10.1109/ICIST.2019.8836731.

[8] J. Manasa, R. Gupta, and N. S. Narahari, "Machine Learning based Predicting House Prices using Regression Techniques," *2nd International Conference on Innovative Mechanisms for Industry Applications, ICIMIA 2020 - Conference Proceedings*, pp. 624–630, Mar. 2020, doi: 10.1109/ICIMIA48430.2020.9074952.

[9] I. Kaur and J. Kaur, "Customer churn analysis and prediction in banking industry using machine learning," *PDGC 2020 - 2020 6th International Conference on Parallel, Distributed and Grid Computing*, pp. 434–437, Nov. 2020, doi: 10.1109/PDGC50313.2020.9315761.

[10] Y. Deng, D. Li, L. Yang, J. Tang, and J. Zhao, "Analysis and prediction of bank user churn based on ensemble learning algorithm," *Proceedings of 2021 IEEE International Conference on Power Electronics, Computer Applications, ICPECA 2021*, pp. 288–291, Jan. 2021, doi: 10.1109/ICPECA51329.2021.9362520.

[11] N. T. M. Sagala and S. D. Permai, "Enhanced Churn Prediction Model with Boosted Trees Algorithms in the Banking Sector," *2021 International Conference on Data Science and Its Applications, ICoDSA 2021*, pp. 240–245, 2021, doi: 10.1109/ICODSA53588.2021.9617503.

[12] M. Rahman and V. Kumar, "Machine Learning Based Customer Churn Prediction in Banking," *Proceedings of the 4th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2020*, pp. 1196–1201, Nov. 2020, doi: 10.1109/ICECA49313.2020.9297529.