ACKNOWLEDGMENT

This includes mentioning of all the references, research papers, data sources, professionals and other resources that helped you and guided you in completion of the project.

• I would like to thank FlipRobo Technologies for providing me this opportunity and guidance throughout the project and all the steps that are implemented.

• I have primarily referred to various articles scattered across various websites for the purpose of getting an idea on Housing project.

• I have referred to various articles in Towards Data Science and Kaggle

INTRODUCTION

• Business Problem Framing

• Houses are one of the necessary need of each and every person around the globe and therefore housing and real estate market is one of the markets which is one of the major contributors in the world's economy. It is a very large market and there are various companies working in the domain. Data science comes as a very important tool to solve problems in the domain to help the companies increase their overall revenue, profits, improving their marketing strategies and focusing on changing trends in house sales and purchases. Predictive modelling, Market mix modelling, recommendation systems are some of the machine learning techniques used for achieving the business goals for housing companies. Our problem is related to one such housing company.

• A US-based housing company named Surprise Housing has decided to enter the Australian market. The company uses data analytics to purchase houses at a price below their actual values and flip them at a higher price. For the same purpose, the company has collected a data set from the sale of houses in Australia. The data is provided in CSV

• The company is looking at prospective properties to buy houses to enter the market. You are required to build a model using Machine Learning in order to predict the actual value of the prospective properties and decide whether to invest in them or not. For this company wants to know:

• Which variables are important to predict the price of variable?

• How do these variables describe the price of the house? Business Goal:

• You are required to model the price of houses with the available independent variables. This model will then be used by the management to understand how exactly the prices vary with the variables. They can accordingly manipulate the strategy of the firm and concentrate on areas that will yield high returns. Further, the model will be a good way for the management to understand the pricing dynamics of a new market.

• Conceptual Background of the Domain Problem Estimating the sale prices of houses is one of the basic projects to have on your Data Science CV. By finishing this article, you will be able to predict continuous variables using various types of linear regression algorithm: (Linear regression is an algorithm used to predict values that are continuous in nature. It became more popular because it is the best algorithm to start with if you are a newbie to ML.) Technical Requirements:

• Data contains 1460 entries each having 81 variables.

• Data contains Null values.Data treatment using domain knowledge, understanding.

• Extensive EDA has to be performed to gain relationships of important variable and price.

• Data contains numerical as well as categorical variable, handle them accordingly.

• Machine Learning models, apply regularization and determine values of Hyper Parameters.

• You need to find important features which affect the price positively or negatively.

• Two datasets are being provided (test.csv, train.csv), train model on train.csv dataset and predict on test.csv file.

• Review of Literature Housing is one out of the 3 basic needs for human survival hence research about housing and all that relates to it can never be over emphasized. In this section of this research work I will examine different papers or research work that have been published previously as regards housing and prices in the past and the research gap noticed which forms the basis for my own research work. In this study, we will use a housing dataset presented by De Cock (2011). This dataset describes the sales of residential units in Ames, Iowa starting from 2006 until 2010. The dataset contains a large number of variables that are involved in determining a house price. The performance of the model were evaluated using mean absolute percentage error (MAPE) performance metric. MAPE was calculated using this formula: MAPE=∑ri=1(abs(yi−pi)/yi)r×100MAPE=∑i=1r(abs(yi−pi)/yi)r×100 where pipi is the predicted stock price on day ii, yiyi is the actual stock price on day, ii, and rr is the number of trading days. Steps : ⎫ Importing the required packages into our python environment ⎫ Importing the house price data and do some EDA on it ⎫ Data Visualization on the house price data ⎫ Feature Selection & Data Split ⎫ Modelling the data using the algorithms ⎫ Evaluating the built model using the evaluation metrics Finally, we conclude which model is best suitable for the given case by evaluating each of them using the evaluation metrics provided by the scikit-learn package.

• Motivation for the Problem Undertaken

• As we know that housing and real estate market is one of the markets which is one of the major contributors in the world's economy. The model is use by the management to understand how exactly the prices vary with the variables.

• Analytics data will help company to purchase houses at a price below their actual values and flip them at a higher price. For the same purpose, the company has collected a data set from the sale of houses in Australia.

• Record of low home loan rates and deficiency of stock is helpful to keep the US real market solid, home costs have been flooding month-by-month and breaking new records.

• Cost are ascending as there is a lot of capital uninvolved, just as exceptionally modest home loan rates, as well as new gaint are also getting involved and for longer period of time home costs have been set in mid-single digits.

• As a less number house owners in different reasons; due to developing expenses and land financial backers gathering , starter houses lodging supply is by and by at its most minimal level.

- As houses are one of the necessary need of each and every person around the globe and therefore housing and real estate market is needed new people are migrating so there is requirement of new houses as per their requirements. Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

- We are going to work with the house price dataset that contains various features and information about the house and its sale price. Using the 'read_csv' function provided by the Pandas package, we can import the data into our python environment. After importing the data, we can use the 'head' function to get a glimpse of our dataset.

- A US-based housing company named Surprise Housing has decided to enter the Australian market,
- Housing is one of the fundamental essential of every living thing hence the reason for continuous research in this sector. This project simply examines a dataset, which consists 1460 observations and 80 features that contribute to the sale price of the houses. Dataset was cleaned and transformed and some explorations were done on it to answer some basic questions that anybody would like to ask about housing. Feature engineering was performed on the transformed data using Principal Component Analysis (PCA) and encoding and this is to ensure our dataset is ready in the right form with the right variables to be used in the algorithms, which results in improved model accuracy. Different ensemble algorithms were used on the dataset in this project. The overall result of this project shows that the most important variables that determine the price of a house being sold. Keywords: Housing price, Principal Component Analysis, encoding, Ensemble Algorithms and Feature Engineering.

- Data Sources and their formats

- The dataset has been provided by the FlipRobo technologies only for academic use, not for any commercial.

- The dataset describe data related to housing with 1460 records.

- The dataset is in csv. Format which contains train and test data.

- This dataset is to use simply examines data, which consists 1460 observations and 80 features for model predication.

- The dataset is in both numerical as well as categorical data.

- Data Preprocessing Done The dataset received from FlipRobo technologies, data describe about housing for client US based company which wishes to get in Australian market based on dataset model. Data Pre-processing and Transformation: The extracted CSV file was loaded then, Structure of the data is investigated as against the meta data provided. It was discovered that some variables were numeric instead of factors so that was converted to factors. The ID column in the dataset was dropped as it isn't necessary for the prediction Missing values in the numerical variables were replaced by the mean of the column while for factor variables it was replaced with 'No_' and this is because the explanation in the metadata says where there is a missing value it is because that feature doesn't exist for that house. Specificity was done whereby all the levels in the factor columns were correctly represented as it is in the meta data Standard deviation of the dependent variable was calculated and it showed that the independent variables aren't to far away from the mean of the sale price. It shows that the values in the dataset are normally distributed. There is no noticeable outlier to be worried about as checked; Multicollinearity was used to check for the correlation between the independent variables.

Important package required

Our primary packages for this project are going to be pandas for data processing, NumPy to work with arrays, matplotlib & seaborn for data visualizations, and finally scikit-learn for building an evaluating our ML model.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [24]:
```
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import ExtraTreesRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.experimental import enable_hist_gradient_boosting
from sklearn.ensemble import HistGradientBoostingRegressor
from sklearn.ensemble import GradientBoostingRegressor

from sklearn.metrics import mean_squared_error, mean_absolute_error
from statsmodels.stats.outliers_influence import variance_inflation_factor

from sklearn.model_selection import train_test_split
from scipy.stats import zscore
from sklearn.metrics import r2_score

import warnings
warnings.filterwarnings("ignore")
```

Importing csv dataset:

In [2]:
```
df01=pd.read_csv('HousingProject_TESTDATA')
```

In [4]:
```
df02=pd.read_csv('HousingProject_TRAINDATA')
df02
```

• Data Inputs- Logic- Output Relationships

Exploratory Data Analysis (EDA)

o This section shows the exploration done on the dataset, which is what motivated the use of the algorithm. The following are the questions explored in this project and for the sake of writing I will only show some of the visuals here while I will provide the codes that shows the full visualization of all the questions explored.

o Is there a significant relationship between sale price and building's age? It was used to check for this and we can see that there is a relationship between how much old the building is and how much it was sold for.

o What is the average sale price based on overall condition of the house, year it was built, condition1 - proximity to social amenities and sale condition? For overall condition we have levels 1-9 with 1

been the lowest and 9 been the highest and the average of each level is shown. For the others the result is in the code provided.

o What is the sale price distribution based on the overall quality of the house?

o What categories of house (based on age built) have the highest sale price? Here it was obvious that houses built below 50 years have sales price higher than $700,000

o Sale Price versus Month it was sold. From here we saw that house price increases more during winter than autumn, spring and summer.

o What sale type has the highest sale price? There are 9 different types of sales type that was considered against the sale price.

o Price distribution and season. The bar charts shows there is higher percentage of people buy houses across all seasons at less than 200k.

o At what price will people buy more even with garage attached. Density of garage type has a high peak at claim size about 160k$. It tells us that people are liable to buy houses at that point regardless of the sale price as long as a garage is attached to the house

o Sales per seasons. The probability of people buying houses is higher in summer and spring is more than autumn and winter.

In [18]:
```
# Fill the columns with mean as its continous data

df02["LotFrontage"].fillna(df02["LotFrontage"].mean(), inplace=True)
df02["MasVnrArea"].fillna(df02["MasVnrArea"].mean(), inplace=True)
```

In [19]:
```
# Fill the columns with mode as its categorical data

df02["MasVnrType"].fillna(df02["MasVnrType"].value_counts().index[0], inplace=True)
df02["BsmtQual"].fillna(df02["BsmtQual"].value_counts().index[0], inplace=True)
df02["BsmtCond"].fillna(df02["BsmtCond"].value_counts().index[0], inplace=True)
df02["BsmtExposure"].fillna(df02["BsmtExposure"].value_counts().index[0], inplace=True)
df02["BsmtFinType1"].fillna(df02["BsmtFinType1"].value_counts().index[0], inplace=True)
df02["BsmtFinType2"].fillna(df02["BsmtFinType2"].value_counts().index[0], inplace=True)
df02["FireplaceQu"].fillna(df02["FireplaceQu"].value_counts().index[0], inplace=True)
df02["GarageType"].fillna(df02["GarageType"].value_counts().index[0], inplace=True)
df02["GarageYrBlt"].fillna(df02["GarageYrBlt"].value_counts().index[0], inplace=True)
df02["GarageFinish"].fillna(df02["GarageFinish"].value_counts().index[0], inplace=True)
df02["GarageQual"].fillna(df02["GarageQual"].value_counts().index[0], inplace=True)
df02["GarageCond"].fillna(df02["GarageCond"].value_counts().index[0], inplace=True)
```

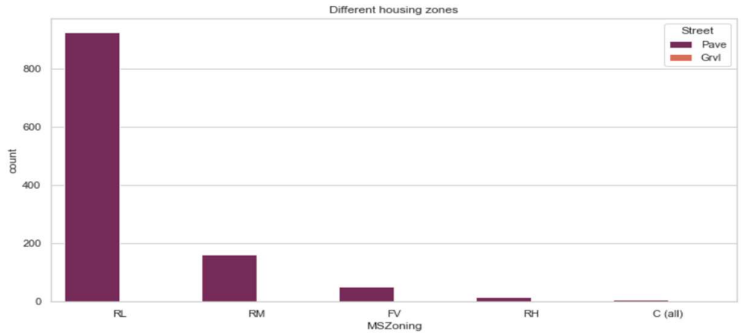
In [20]:
```
df02.drop(columns = ["Alley", "PoolQC", "Fence", "MiscFeature"], axis=1, inplace=True)
df01.drop(columns = ["Alley", "PoolQC", "Fence", "MiscFeature"], axis=1, inplace=True)
```

Data visualization:

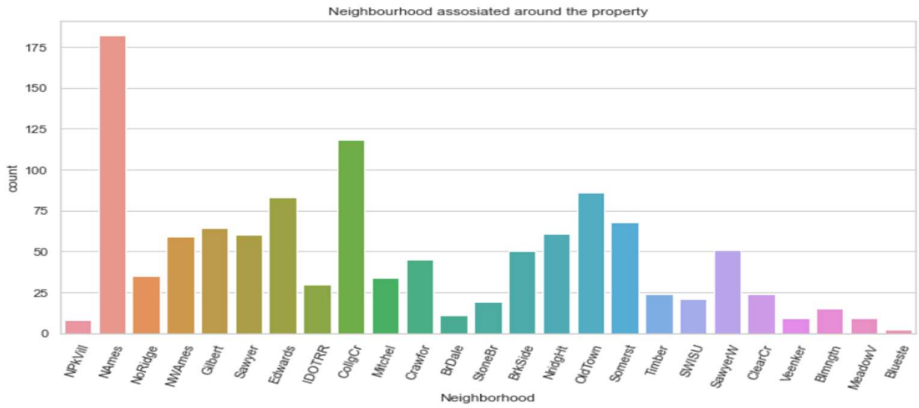
In [24]:
```
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import ExtraTreesRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.experimental import enable_hist_gradient_boosting
from sklearn.ensemble import HistGradientBoostingRegressor
from sklearn.ensemble import GradientBoostingRegressor

from sklearn.metrics import mean_squared_error, mean_absolute_error
from statsmodels.stats.outliers_influence import variance_inflation_factor

from sklearn.model_selection import train_test_split
from scipy.stats import zscore
from sklearn.metrics import r2_score

import warnings
warnings.filterwarnings("ignore")
```

In [25]:
```
# Plot below can identify the zoning area of the properties / apartments that are up for sale.

plt.figure(figsize=(12, 6))
sns.set_theme(style="whitegrid")
ax = sns.countplot("MSZoning", data=df02, hue= "Street", palette="rocket").set(title='Different housing zones')
plt.show()
```

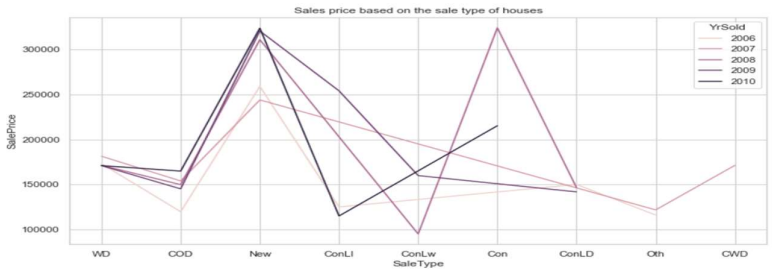
In [25]:
```
# Plot below can identify the zoning area of the properties / apartments that are up for sale.

plt.figure(figsize=(12, 6))
sns.set_theme(style="whitegrid")
ax = sns.countplot("MSZoning", data=df02, hue= "Street", palette="rocket").set(title='Different housing zones')
plt.show()
```



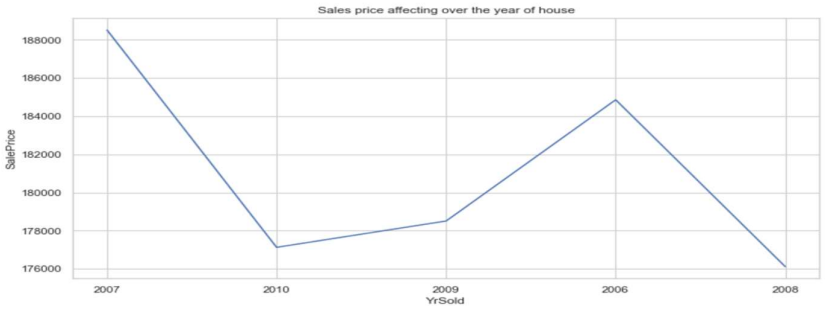
In [27]:
```
# Plot gives idea on the locality of neughbourhood

plt.figure(figsize=(12, 6))
sns.set_theme(style="whitegrid")
ax = sns.countplot("Neighborhood", data=df02).set(title='Neighbourhood assosiated around the property')
plt.xticks(rotation=70)
plt.show()
```



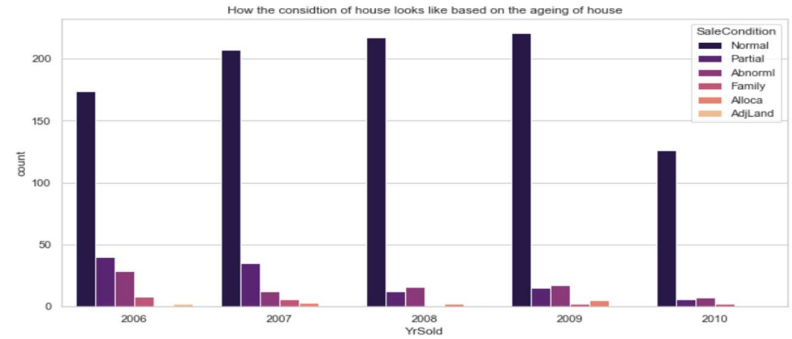
In [28]:
```
# Relation between SaleType and SalePrice.

plt.figure(figsize=(12, 6))
sns.set_theme(style="whitegrid")
sns.lineplot(data=df02, x="SaleType", y="SalePrice", hue = "YrSold", ci=None).set(title='Sales price based on the sale type
plt.show()
```



In [29]:
```
# Plot shows us the years when properties were sold the highest.

year_new = df02['YrSold'].astype(str)

plt.figure(figsize=(12, 6))
sns.set_theme(style="whitegrid")
sns.lineplot(data=df02, x=year_new, y="SalePrice", ci=None).set(title='Sales price affecting over the year of house')
plt.show()
```



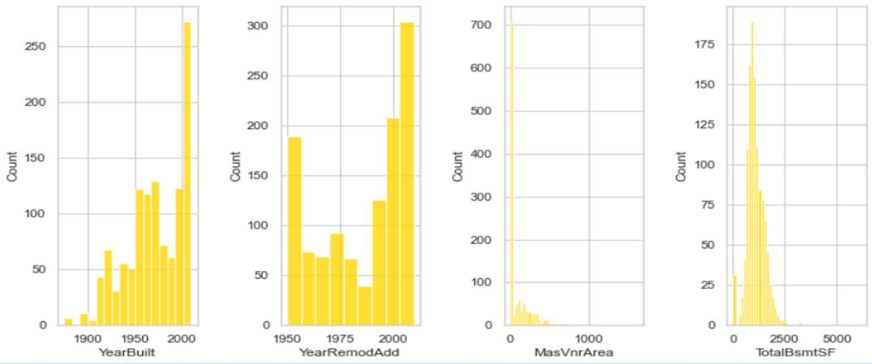
In [30]:
```
plt.figure(figsize=(12, 6))
sns.set_theme(style="whitegrid")
ax = sns.countplot("YrSold", data=df02, palette= "magma",
                   hue="SaleCondition").set(title='How the considtion of house looks like based on the ageing of house')
plt.show()
```



In [31]:
```
# It appears Warranty Deed seems to be the most sought after payment or purchasing method ,it's clearly in favour of both th

gap_of_sale = df02["YrSold"] - df02["YearBuilt"]

plt.figure(figsize=(12, 6))
sns.set_theme(style="whitegrid")
sns.scatterplot(data=df02, x=gap_of_sale,
                y="SalePrice", hue="SaleType").set(title='Sale price of house baed on sale of year')
plt.show()
```

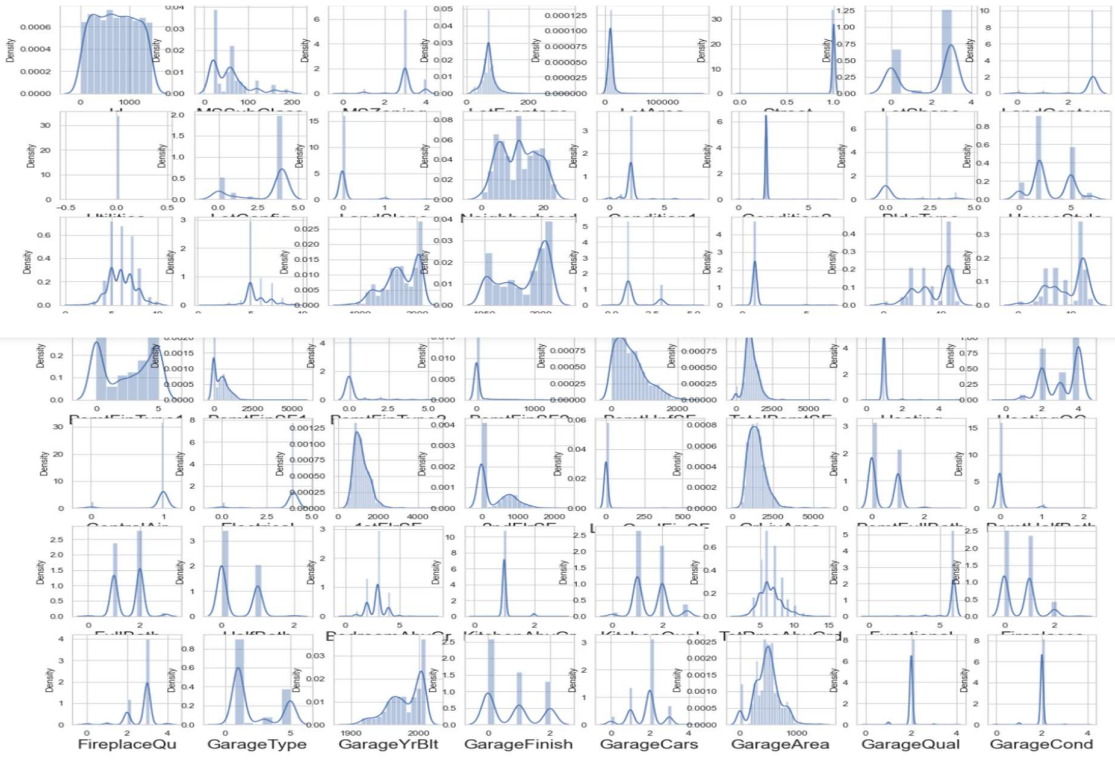
In [41]:

```
#  Relationship between the target variable and the variables that are positively correlated with it.

fig, axes = plt.subplots(1, 4, figsize=(12,6))
fig.subplots_adjust(hspace=0.5, wspace=0.6)
for ax, v in zip(axes.flat, ["YearBuilt", "YearRemodAdd",
                             "MasVnrArea", "TotalBsmtSF"]):
    sns.distplot(df02[v], kde=False, color= 'Gold',
                 hist_kws={"alpha": 0.8}, ax=ax)
    ax.set(ylabel="Count");
```



In [49]:

```
# Distribution of all the columns in the dataset:

# let us now see the distribution of the "Train dataset"

plt.figure(figsize=(20,25), facecolor="white")
plotnumber = 1

for column in df02:
    if plotnumber <=77:
        ax = plt.subplot(8,8, plotnumber)
        sns.distplot(df02[column])
        plt.xlabel(column, fontsize=20)
    plotnumber+=1
plt.tight_layout()
```



• Model/s Development and Evaluation

• Identification of possible problem-solving approaches (methods)

• The factors need to be found which can impact the housing price . This can be done by analysing the various factors and the store the respondent prefers. This will be done by checking each of the factors impacts the respondents decision making.

In [47]:
```
# Encode the training dataset

df02.MSZoning = encoder.fit_transform(df02.MSZoning)
df02.Street = encoder.fit_transform(df02.Street)
df02.LotShape = encoder.fit_transform(df02.LotShape)
df02.LandContour = encoder.fit_transform(df02.LandContour)
df02.Utilities = encoder.fit_transform(df02.Utilities)
df02.LotConfig = encoder.fit_transform(df02.LotConfig)
df02.LandSlope = encoder.fit_transform(df02.LandSlope)
df02.Neighborhood = encoder.fit_transform(df02.Neighborhood)
df02.Condition1 = encoder.fit_transform(df02.Condition1)
df02.Condition2 = encoder.fit_transform(df02.Condition2)
df02.BldgType = encoder.fit_transform(df02.BldgType)
df02.HouseStyle = encoder.fit_transform(df02.HouseStyle)
df02.RoofStyle = encoder.fit_transform(df02.RoofStyle)
df02.RoofMatl = encoder.fit_transform(df02.RoofMatl)
df02.Exterior1st = encoder.fit_transform(df02.Exterior1st)
df02.Exterior2nd = encoder.fit_transform(df02.Exterior2nd)
df02.MasVnrType = encoder.fit_transform(df02.MasVnrType)
df02.ExterQual = encoder.fit_transform(df02.ExterQual)
df02.ExterCond = encoder.fit_transform(df02.ExterCond)
df02.Foundation = encoder.fit_transform(df02.Foundation)
df02.BsmtQual = encoder.fit_transform(df02.BsmtQual)
df02.BsmtCond = encoder.fit_transform(df02.BsmtCond)
df02.BsmtExposure = encoder.fit_transform(df02.BsmtExposure)
df02.BsmtFinType1 = encoder.fit_transform(df02.BsmtFinType1)
df02.BsmtFinType2 = encoder.fit_transform(df02.BsmtFinType2)
df02.Heating = encoder.fit_transform(df02.Heating)
df02.HeatingQC = encoder.fit_transform(df02.HeatingQC)
df02.CentralAir = encoder.fit_transform(df02.CentralAir)
df02.Electrical = encoder.fit_transform(df02.Electrical)
df02.KitchenQual = encoder.fit_transform(df02.KitchenQual)
df02.Functional = encoder.fit_transform(df02.Functional)
df02.FireplaceQu = encoder.fit_transform(df02.FireplaceQu)
df02.GarageType = encoder.fit_transform(df02.GarageType)
```

In [56]:
```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
x_scaled = scaler.fit_transform(x)

```

In [57]:
```
x_train, x_test, y_train, y_test = train_test_split(x_scaled, y, test_size=0.30, random_state = 200)
```

KNN:

In [58]:
```
from sklearn.neighbors import KNeighborsRegressor

k_neigh = KNeighborsRegressor()
k_neigh.fit(x_train,y_train)

y_pred = k_neigh.predict(x_test)

print("Adjusted R2 squared : ",k_neigh.score(x_train,y_train))
print("Mean Absolute Error (MAE): ", mean_absolute_error(y_test, y_pred))
print("Mean Squared Error (MSE): ",mean_squared_error(y_test, y_pred))
print("Root Mean Squared Error (RMSE): ",np.sqrt(mean_squared_error(y_test, y_pred)))
```

```
Adjusted R2 squared :  0.8667901832681217
Mean Absolute Error (MAE):  23323.041025641025
Mean Squared Error (MSE):  1683800580.468718
Root Mean Squared Error (RMSE):  41034.13920711287
```

DTR:

In [59]:
```
from sklearn.tree import DecisionTreeRegressor

dt_reg = DecisionTreeRegressor()
dt_reg.fit(x_train,y_train)

y_pred = dt_reg.predict(x_test)

print("Adjusted R2 squared : ",dt_reg.score(x_train,y_train))
print("Mean Absolute Error (MAE): ", mean_absolute_error(y_test, y_pred))
print("Mean Squared Error (MSE): ",mean_squared_error(y_test, y_pred))
print("Root Mean Squared Error (RMSE): ",np.sqrt(mean_squared_error(y_test, y_pred)))
```

```
Adjusted R2 squared :  1.0
Mean Absolute Error (MAE):  29726.11396011396
Mean Squared Error (MSE):  2167883538.7236466
Root Mean Squared Error (RMSE):  46560.53628045586
```

RFR:

In [60]:
```
from sklearn.ensemble import RandomForestRegressor

rf_reg = RandomForestRegressor()
rf_reg.fit(x_train,y_train)

y_pred = rf_reg.predict(x_test)

print("Adjusted R2 squared : ",rf_reg.score(x_train,y_train))
print("Mean Absolute Error (MAE): ", mean_absolute_error(y_test, y_pred))
print("Mean Squared Error (MSE): ",mean_squared_error(y_test, y_pred))
print("Root Mean Squared Error (RMSE): ",np.sqrt(mean_squared_error(y_test, y_pred)))
```

```
Adjusted R2 squared :  0.9756856910367635
Mean Absolute Error (MAE):  21662.56547008547
Mean Squared Error (MSE):  1463886499.469246
Root Mean Squared Error (RMSE):  38260.769718724245
```

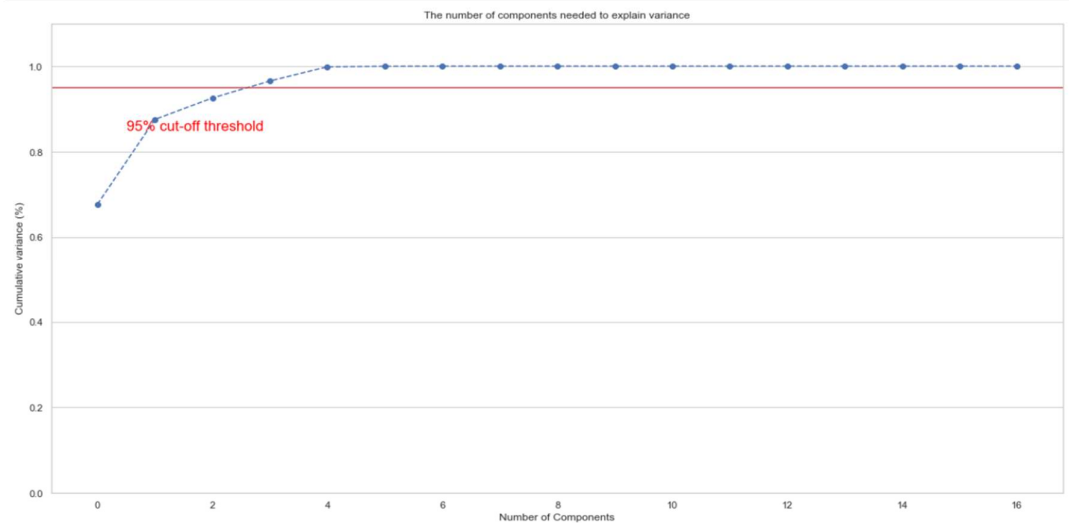
ETR:

In [61]:
```
from sklearn.ensemble import ExtraTreesRegressor

extra_reg = ExtraTreesRegressor()
extra_reg.fit(x_train,y_train)

y_pred = extra_reg.predict(x_test)

print("Adjusted R2 squared : ",extra_reg.score(x_train,y_train))
print("Mean Absolute Error (MAE): ", mean_absolute_error(y_test, y_pred))
print("Mean Squared Error (MSE): ",mean_squared_error(y_test, y_pred))
print("Root Mean Squared Error (RMSE): ",np.sqrt(mean_squared_error(y_test, y_pred)))
```

```
Adjusted R2 squared :  1.0
Mean Absolute Error (MAE):  22482.002478632483
Mean Squared Error (MSE):  1584047891.1128936
Root Mean Squared Error (RMSE):  39800.09913446063
```

In [71]:
```
import numpy as np
from sklearn.decomposition import PCA
```

In [72]:
```
pca = PCA().fit(x)
plt.rcParams["figure.figsize"] = (20,10)

fig, ax = plt.subplots()

y = np.cumsum(pca.explained_variance_ratio_)

plt.ylim(0.0,1.1)
plt.plot(y, marker='o', linestyle='--', color='b')

plt.xlabel('Number of Components')
plt.ylabel('Cumulative variance (%)')
plt.title('The number of components needed to explain variance')

plt.axhline(y=0.95, color='r', linestyle='-')
plt.text(0.5, 0.85, '95% cut-off threshold', color = 'red', fontsize=16)

ax.grid(axis='x')
plt.show()
```



Saving the model

```
In [73]:
import joblib
joblib.dump(rf_reg_tuned,"Housing_Pred.pkl")
```

```
Out[73]:
['Housing_Pred.pkl']
```

```
In [74]:
model = joblib.load("Housing_Pred.pkl")

prediction = model.predict(x_test)

prediction=pd.DataFrame(prediction)
prediction
```

Out[74]:

	0
0	118786.16
1	155695.24
2	222500.10
3	194214.77
4	280763.24
...	...
346	134296.20
347	89534.93
348	302932.87
349	215098.58
350	319634.11

351 rows × 1 columns

Interpretation of the Results

o In this research, two experiments were performed, the first experiment was conducted using all the variables available in the dataset after pre-processing, while the second experiment was conducted using most important variables and the goal of this is to be able to improve the model's performance using fewer variables.

o There requirement of train and test and building of many models to get accuracy of the model.

o There are multiple of matric which decide the best fit model like as : R-squared ,RMSE value, etc.

o Database helped in making perfect model and will help in understanding Australian market

CONCLUSION

In conclusion after reviewing the above papers the first thing I noticed with the dataset is that a lot of research has been done in the past on US housing sector hence I got data for other country, Australian houses. Also different algorithms have been used to predict housing prices but the most efficient one is regression of which my research work will perform on this dataset using different regression algorithms and present the result to know which gives the best accuracy using the R-squared and RMSE value.