SURPRISE HOUSING CASE STUDY

ABSTRACT:

The focus of this project is Surprise Housing, a US-based housing company that wants to grow by entering the Australian real estate industry. The organization looks to find possible properties to buy by using data analytics to buy them below market value and then sell them for a profit. It is suggested to use regularization techniques in a regression model to aid in making well-informed decisions. This model will help Surprise Housing assess whether investments are feasible by forecasting the true values of potential properties. The company intends to strategically enter the Australian housing industry by optimizing its decision-making process through the application of sophisticated statistical methods. The research entails data analysis, model creation, and validation in order to improve prediction accuracy and reliability, which in turn helps Surprise Housing make wise investment decisions.

OBJECTIVE:

Creating a solid regression model with regularization techniques is the main objective of this project for Surprise Housing, a US-based business that intends to join the Australian real estate market. The objective is to precisely forecast the true values of potential attributes by utilizing a dataset that has been provided. With the help of the model, Surprise Housing will be able to make well-informed investment decisions by finding homes that may be bought for less than their market worth and then sold for a profit. The project aims to optimize the company's investment strategy in the Australian housing market by improving prediction accuracy and reliability through the utilization of advanced data analytics and regularization techniques.

INTRODUCTION:

For businesses looking to strategically enter the real estate sector, the ever-changing face of the industry offers both opportunities and obstacles. In this regard, the well-known US housing provider Surprise Housing's consideration of entering the Australian real estate market stands out. Surprise Housing, with a strong focus on predictive analytics and a data-driven methodology, wants to use regularization techniques in regression modelling to maximize its potential. The goal is apparent: to provide accurate estimates of future property prices, enabling wise investment choices in a market renowned for its subtleties and intricacies.

The primary focus of Surprise Housing is to purchase homes at a discount to their market prices, then take advantage of market movements to optimize profits. The company's primary focus is the Australian real estate market, and it uses both historical data and sophisticated statistical techniques to find the best investment prospects. Potential financial rewards are important, but so is the larger picture of adjusting to changing market conditions and guaranteeing steady growth.

Using a properly chosen dataset to provide a thorough depiction of the Australian real estate market is the foundation of this project. This dataset serves as the basis for the creation of a regression model that uses regularization strategies. Our goal in analyzing this dataset is to find trends, correlations, and patterns that will enable Surprise Housing to make well-informed judgments about the purchase of real estate.

Due to the complexity of the Australian real estate industry, a sophisticated and data-driven strategy is required. With this research, we hope to assist Surprise Housing in making strategic decisions by providing insights into the elements affecting property values in addition to developing a predictive model. It is expected that this project would produce benefits beyond short-term financial gain and act as a model for similar projects in unexplored real estate markets in the future. As we continue our investigation, it becomes clear that Surprise Housing's success in its quest of a profitable and sustainable entry into the Australian market is largely due to the combination of data science with practical business goals.

METHODOLOGY:

1. ### Importing Data / Reading CSV:

   Importing the required data into the data analysis environment of our choice is the first stage in our data analysis and modelling process. Here, the relevant data regarding the Australian real estate market is contained in a CSV (Comma-Separated Values) file.

2. ### Data Cleaning:

   - **Identify Missing Values:** Identify and assess the extent of missing values in the dataset using methods such as `isnull()` or `info()`.
   - **Handle Missing Values:** Decide on a strategy for handling missing values, whether through imputation, deletion of rows/columns, or other methods based on the nature and impact of the missing data.
   - **Detect and Handle Outliers:** Identify outliers that may skew analysis and decide on appropriate methods for handling them, such as removing outliers or transforming values.
   - **Check and Correct Datatypes:** Verify that each variable has the correct data type (e.g., numerical, categorical) and convert as needed.
   - **Remove Duplicate Records:** Identify and remove duplicate records to avoid redundancy in the dataset.
   - **Validate Cleaned Dataset:** Conduct final checks to ensure that the dataset is free from errors, inconsistencies, and missing values.

3. ### Exploratory Data Analysis:

   Exploratory Data Analysis is a crucial phase in understanding the characteristics of a dataset, identifying patterns, and generating insights. The methodology involves the following steps:

   - **Summary Statistics:** Compute and analyze descriptive statistics, such as mean, median, standard deviation, and quartiles, to gain an initial understanding of the central tendencies and variability in the dataset.
   - **Data Visualization:** Create visualizations, such as histograms, box plots, and scatter plots, to visualize the distribution of individual variables and explore relationships between them.
   - **Correlation Analysis:** Examine the correlation between variables using correlation matrices or heatmaps to identify potential relationships and dependencies.

- **Categorical Data Exploration:** Explore categorical variables through frequency tables, bar charts, or pie charts to understand the distribution of categories.
- **Outlier Detection:** Identify outliers using visualizations or statistical methods to assess their impact on data distribution and consider appropriate actions.
- **Pattern Recognition:** Look for patterns or trends in the data that may inform subsequent modelling decisions or hypothesis formulation.

4. Data Preparation:

Data preparation is a critical phase in the data analysis and modelling process, ensuring that the dataset is well-structured, cleaned, and formatted for effective analysis. The methodology involves the following steps:

- **Handling Missing Data:** Identify and address missing values through imputation, deletion, or other suitable methods to ensure completeness and reliability.
- **Dealing with Outliers:** Evaluate and handle outliers using appropriate methods such as trimming, winsorizing, or transforming values to prevent skewed analysis.
- **Encoding Categorical Variables:** Convert categorical variables into numerical format using techniques like one-hot encoding or label encoding to make them suitable for modelling.
- **Feature Scaling:** Standardize or normalize numerical features to a common scale, enhancing the performance of certain machine learning algorithms.
- **Data Splitting:** If building predictive models, split the dataset into training and testing sets to evaluate model performance on unseen data.
- **Final Dataset Check:** Conduct a final check on the prepared dataset to ensure that it is free from errors, inconsistencies, and is appropriately formatted for analysis.

5. Building Machine Learning Model:

When constructing a machine learning model, particularly when dealing with regression tasks, Ridge and Lasso regularization techniques can be employed to enhance model performance and prevent overfitting. Here is a concise methodology for building a machine learning model using Ridge and Lasso regularization:

- **Model Selection:** Choose the appropriate regression model. In this case, consider Ridge Regression and Lasso Regression, which are extensions of linear regression with regularization.
- **Model Training - Ridge Regression:** Train the Ridge Regression model using the training data. Adjust the regularization strength (Here, alpha=0.0001) based on cross-validation or other tuning methods.
- **Model Training - Lasso Regression:** Train the Lasso Regression model similarly, adjusting the regularization strength as needed.
- **Model Evaluation:** Assess the performance of the models on the testing set using appropriate evaluation metrics such as Mean Squared Error (MSE), R-squared, or others.
- **Model Interpretation:** Interpret the coefficients of the Ridge and Lasso models, which are influenced by regularization. This step provides insights into the impact of each feature on the target variable.

CODE:

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

import numpy as np
import pandas as pd

pd.set_option("display.max_columns",None)
pd.set_option("display.max_rows",None)

# Step 1: Import and Inspect Dataset
housing = pd.read_csv("train.csv")
housing.head()
housing.shape
housing.describe()
housing.info()
housing.isnull().sum()/housing.shape[0] * 100

# Step 2: Data Cleaning
cols = ['Alley', 'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1',
'BsmtFinType2', 'FireplaceQu', 'GarageType', 'GarageFinish', 'GarageQual',
'GarageCond', 'PoolQC', 'Fence', 'MiscFeature']
for i in cols:
    housing[i].fillna("None",inplace = True)
housing.info()

import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

plt.figure(figsize =[6,6])
sns.distplot(housing['SalePrice'])
plt.show()

print("Skewness: ", housing['SalePrice'].skew())
print("Kurtosis: ", housing['SalePrice'].kurt())

#Log Transformation
housing['SalePrice'] = np.log(housing['SalePrice'])
plt.figure(figsize =[6,6])
sns.distplot(housing['SalePrice'])
plt.show()
print("Skewness: ", housing['SalePrice'].skew())
print("Kurtosis: ", housing['SalePrice'].kurt())

housing.drop("Id",axis=1, inplace=True)
```

```python
housing[['MSSubClass', 'OverallQual', 'OverallCond']] = housing[['MSSubClass',
'OverallQual', 'OverallCond']].astype('object')
housing['LotFrontage'] = pd.to_numeric(housing['LotFrontage'], errors =
'coerce')
housing['MasVnrArea'] = pd.to_numeric(housing['MasVnrArea'], errors =
'coerce')
housing.info()

null_cols = housing.columns[housing.isnull().any()]
null_cols
for i in null_cols:
    if (housing[i].dtype == np.float64 or housing[i].dtype == np.int64):
        housing[i].fillna(housing[i].mean(), inplace = True)
    else:
        housing[i].fillna(housing[i].mode()[0], inplace = True)
housing.isnull().sum()

# Step 3: Exploratory Data Analysis on Dataset

# list of categorical Columns
cat_cols = housing.select_dtypes(include = 'object').columns
cat_cols

# List of numerical columns
num_cols = housing.select_dtypes(include = ['int64','float64']).columns
num_cols

## Univariant analysis
# Numerical Columns
#Plotting Boxplots to visualize the distribution and check for outliers
for i in num_cols:
    plt.figure(figsize=[8,5])
    print(i)
    sns.boxplot(housing[i])
    plt.show()

#Categorical Columns
# Plotting Pie plots to visualize the values distribution in each category
for i in cat_cols:
    print(housing[i].value_counts(normalize=True))
    plt.figure(figsize=[5,5])
    housing[i].value_counts(normalize=True).plot.pie(labeldistance = None,
autopct = '%1.2f%%')
    plt.legend()
    plt.show()
    print("------------------------------------------------------------------
----------")
```

```
# Bivariate/ Multivariate Analysis on the Dataset
#Plot of MSZoning vs LotFrontage
sns.barplot(x='MSZoning', y='LotFrontage', data= housing)
plt.show

#Plot of MSSubClass vs LotFrontage
sns.barplot(x='MSSubClass', y='LotFrontage', data= housing)
plt.show

# plot of HouseStyle vs SalePrice based on Street
sns.barplot(x='HouseStyle', y='SalePrice',hue= 'Street', data = housing)
plt.show

# Plot of BldgType vs SalePrice
sns.barplot(x='BldgType', y='SalePrice', data=housing)
plt.show()

# Plot of BsmtQual vs SalePrice
sns.barplot(x='BsmtQual', y='SalePrice', data=housing)
plt.show()

# Calculating Age of the Property
housing["Age"] = housing["YrSold"] - housing["YearBuilt"]
housing["Age"].head()

# Dropping YrSold and YearBuilt
housing.drop(columns=['YrSold','YearBuilt'], axis=1, inplace=True)
housing.head()
plt.figure(figsize = [25,25])
sns.heatmap(housing.corr(),annot=True, cmap= 'BuPu')
plt.title("Correlation of Numeric Values")
k =10
plt.figure(figsize=[15,15])
cols = housing.corr().nlargest(k,"SalePrice").index
cm = np.corrcoef(housing[cols].values.T)
sns.heatmap(cm,annot=True,square=True, fmt='.2f', cbar = True,
annot_kws={'size':10},yticklabels=cols.values,xticklabels=cols.values)
plt.show()

# Pairplot for Numerical Columns
cols =
["SalePrice","OverallQual","GrLivArea","GarageCars","TotalBsmtSF","Age"]
plt.figure(figsize=[20,20])
sns.pairplot(housing[cols])
plt.show()

# Step 4: Data Preparation
```

```python
housing_num = housing.select_dtypes(include =['int64','float64'])
housing_cat = housing.select_dtypes(include = 'object')
housing_cat
housing_cat_dm = pd.get_dummies(housing_cat, drop_first=True, dtype=int)
housing_cat_dm
house = pd.concat([housing_num,housing_cat_dm], axis=1)
house.head()
house.shape

#Split into Target and feature variables
X = house.drop(["SalePrice"],axis=1).copy()
y = house["SalePrice"].copy()
X.head()
y.head()
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,
random_state=42)
X_train.shape
y_train.shape

# Scaling the dataset with Standard Scaler
num_cols = list(X_train.select_dtypes(include=['int64','float64']).columns)
scaler = StandardScaler()
X_train[num_cols] = scaler.fit_transform(X_train[num_cols])
X_test[num_cols] = scaler.fit_transform(X_test[num_cols])

# Building a function to calculate evaluation metrics
def eval_metrics(y_train, y_train_pred, y_test, y_pred):

    #r2 values for train and test data
    print("r2 score (train) = ", '%.2f' % r2_score(y_train, y_train_pred))
    print("r2 score (test) = ", "%.2f" % r2_score(y_test, y_pred))

    # RMSE for train and test data
    mse_train = mean_squared_error(y_train, y_train_pred)
    mse_test = mean_squared_error(y_test, y_pred)
    rmse_train = mse_train ** 0.5
    rmse_test = mse_test ** 0.5

    print("RMSE(Train) = ", "%.2f" % rmse_train)
    print("RMSE(Test) = ", "%.2f" % rmse_test)

# Step 5: Build ML Model

# Import ML Libraries
import sklearn
from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error
from sklearn.linear_model import Ridge, Lasso
from sklearn.model_selection import GridSearchCV

# Applying Ridge regression with varying the hyperparameter 'lambda'
params = {'alpha':
             [0.0001,0.001,0.01,0.05,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,2
.0
             ,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10,20,50,100,500,1000]}
ridge = Ridge()
ridgeCV = GridSearchCV(estimator=ridge, param_grid = params,
scoring="neg_mean_absolute_error", cv = 5, return_train_score=True, verbose=1,
n_jobs=-1)
ridgeCV.fit(X_train, y_train)
ridgeCV.best_params_
ridgeCV.cv_results_
ridge = Ridge(alpha=9)
ridge.fit(X_train, y_train)
ridge.coef_
y_train_pred = ridge.predict(X_train)
y_pred = ridge.predict(X_test)
eval_metrics(y_train, y_train_pred, y_test, y_pred)
ridgeCV_res = pd.DataFrame(ridgeCV.cv_results_)
ridgeCV_res.head()
plt.plot(ridgeCV_res['param_alpha'], ridgeCV_res['mean_train_score'], label =
'train')
plt.plot(ridgeCV_res['param_alpha'], ridgeCV_res['mean_test_score'], label =
'test')
plt.xlabel('alpha')
plt.ylabel('R2_score')
plt.xscale('log')
plt.legend()
plt.show()

# Applying Lasso regression with varying the hyperparameter 'lambda'
params = {'alpha':
             [0.0001,0.001,0.01,0.05,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,2
.0
             ,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10,20,50,100,500,1000]}
lasso = Lasso()
lassoCV = GridSearchCV(estimator=lasso, param_grid = params,
scoring="neg_mean_absolute_error", cv = 5, return_train_score=True, verbose=1,
n_jobs=-1)
lassoCV.fit(X_train, y_train)
lassoCV.best_params_
lasso = Lasso(alpha = 0.0001)
lasso.fit(X_train, y_train)
lasso.coef_
y_train_pred1 = lasso.predict(X_train)
y_pred1 = lasso.predict(X_test)
eval_metrics(y_train, y_train_pred1, y_test, y_pred1)
lassoCV_res = pd.DataFrame(lassoCV.cv_results_)
lassoCV_res.head()
plt.plot(lassoCV_res['param_alpha'], lassoCV_res['mean_train_score'], label =
'train')
plt.plot(lassoCV_res['param_alpha'], lassoCV_res['mean_test_score'], label =
'test')
plt.xlabel('alpha')
plt.ylabel('R2_score')
plt.xscale('log')
plt.legend()
plt.show()

# Feature Extraction/Elimination

betas = pd.DataFrame(index=X.columns) # Convert the columns to a Dataframe as
betas
betas.rows = X.columns

# Creating columns for Ridge and lasso coefficients against each feature
betas['Ridge'] = ridge.coef_
betas['Lasso'] = lasso.coef_
betas

# View the features removed by Lasso
lasso_cols_removed = list(betas[betas['Lasso']==0].index)
print(lasso_cols_removed)

# View the features selected by Lasso
lasso_cols_selected = list(betas[betas['Lasso']!=0].index)
print(lasso_cols_selected)
print(len(lasso_cols_removed)) # 179 features are removed by Lasso
print(len(lasso_cols_selected)) # 107 features are selected by Lasso

# View the top 10 coefficients of Ridge Regression in descending order
betas['Ridge'].sort_values(ascending=False)[:10]

# View the top 10 coefficients of Lasso in descending order
betas['Lasso'].sort_values(ascending=False)[:10]

# We have to take inverse log of betas to interpret the ridge coefficients in
terms of target variable
lasso_coeffs = np.exp(betas['Lasso'])
lasso_coeffs.sort_values(ascending=False)[:10]
```

CONCLUSION:

Finally, by applying a data-driven strategy and machine learning approaches, this case study explored Surprise Housing's strategic entry into the Australian real estate market. The trip started with thorough data exploration, which allowed for a better comprehension of the Australian real estate market and gave insights that helped with decision-making later on.

While data preparation involved encoding categorical variables, scaling features, and constructing derived features for optimal model performance, data cleaning protected the dataset's integrity by resolving missing values and outliers. A machine learning model was painstakingly constructed, using Lasso and Ridge regularization to reduce overfitting and improve prediction accuracy.

Using pertinent measures, the evaluation phase examined the model's performance and provided insight into its capacity to generalize to new data. Understanding the model's predictions and pinpointing areas for development was made easier with the use of visualizations. The best strategy was chosen through an iterative process of model comparison, hyperparameter tweaking, and fine-tuning.

Surprise Housing is a predictive algorithm that can evaluate the true worth of potential properties thanks to this thorough process. In a market renowned for its subtleties, the coefficients obtained from Ridge and Lasso offer insightful information on the importance of each aspect, assisting in the making of investment decisions.

This case study serves as a model for utilizing machine learning in strategic decision-making in addition to being a data-driven entry into the Australian real estate market. The methodology's iterative structure emphasizes the value of ongoing improvement and guarantees the model's flexibility in response to changing market conditions.

With a strong model in hand, Surprise Housing contemplates its next moves as the voyage comes to an end with the prospect of well-informed and well-timed forays into the Australian real estate market, laying the groundwork for long-term success and profitability.