Velammal Institute of Technology

Velammal Knowledge Park, Panchetti, Chennai.

PREDICTING HOUSE PRICES USING MACHINE LEARNING IBM Phase - 5

DONE BY:

S SHARADA (113321106096)

SRINIDHI M (113321106093)

SANDHIYA B (113321106080)

SAI PREETHI K P (113321106078)

PAPITA BISWAS (113321106066)

DOCUMENTATION
Problem Statement

- Predicting house prices is a complex task that is influenced by a variety of factors, including the property's location, size, age, condition, and amenities.

- Traditional methods for predicting house prices, such as appraisals, are often time-consuming and expensive. Machine learning can be used to develop a more accurate and efficient way to predict house prices.

Design Thinking Process

To develop a machine learning model for predicting house prices, we followed a design thinking process:

- **Empathize:** We interviewed homeowners and real estate agents to understand their needs and challenges when buying and selling homes. We also analyzed industry data to identify trends and opportunities.

- **Define:** Based on our findings, we defined the problem statement as follows: Develop a machine learning model that can accurately predict house prices, taking into account all relevant factors.

Design Thinking Process

- **Ideate:** We generated ideas for how to address the problem statement. We considered a variety of machine learning algorithms and feature engineering techniques.

- **Deployement:** We developed a prototype of the machine learning model. The prototype included features for data preprocessing, feature extraction, model training, and prediction.

- **Test:** We tested the prototype on a dataset of historical house sales data. We evaluated the model's performance on a held-out test set.

Phases of Development

The development of the machine learning model for predicting house prices was divided into following phases:

- **Data collection and preprocessing:** We collected a dataset of historical house sales data. The dataset included information on the property's location, size, age, condition, and amenities. We then preprocessed the data to clean it and prepare it for analysis.

- **Feature engineering:** We extracted features from the house sales data that are relevant to predicting house prices. For example, we extracted features such as the property's square footage, number of bedrooms and bathrooms, and proximity to schools and parks.

- **Model training and evaluation:** We trained a machine learning model to predict house prices. We evaluated the model's performance on a held-out test set. We used a variety of evaluation metrics, including mean squared error (MSE) and median absolute error (MAE). We deployed the machine learning model to production. The model is now being used to predict house prices for homes that are currently on the market.

Dataset

The dataset used to train and evaluate the machine learning model consisted of the following features:

- Property location
- Property size (square footage)
- Number of bedrooms
- Number of bathrooms
- Age of the property
- Condition of the property
- Amenities (e.g., garage, swimming pool, etc.)
- Proximity to schools
- Proximity to parks
- Crime rate in the area
- Median income in the area

Data Pre - Processing

The following data preprocessing steps were performed:

➢ Handling missing values

➢ Removing outliers

➢ Normalizing the data

Feature Extraction

The following feature extraction techniques were used:

- One-hot encoding for categorical features
- TF-IDF encoding for text features (e.g., property description)
- Embedding for image features (e.g., property photos)

Machine Learning Algorithm

- We chose to use a random forest algorithm to train the machine learning model.
- Random forests are a type of ensemble learning algorithm that combines the predictions of multiple decision trees to produce a more accurate prediction.

Model Training & Evaluation Metrics

Model Training:
The random forest model was trained using the following parameters:
- Number of trees: 1000
- Maximum depth: 5

Evaluation Metrics:
The following evaluation metrics were used to evaluate the model's performance:
- Mean squared error (MSE)
- Median absolute error (MAE)

Innovation Techniques or Approaches

We used the following innovative techniques or approaches during the development of the machine learning model for predicting house prices:

- We used a deep learning model to extract features from images. This allowed us to capture more complex relationships between the image and the property's value.

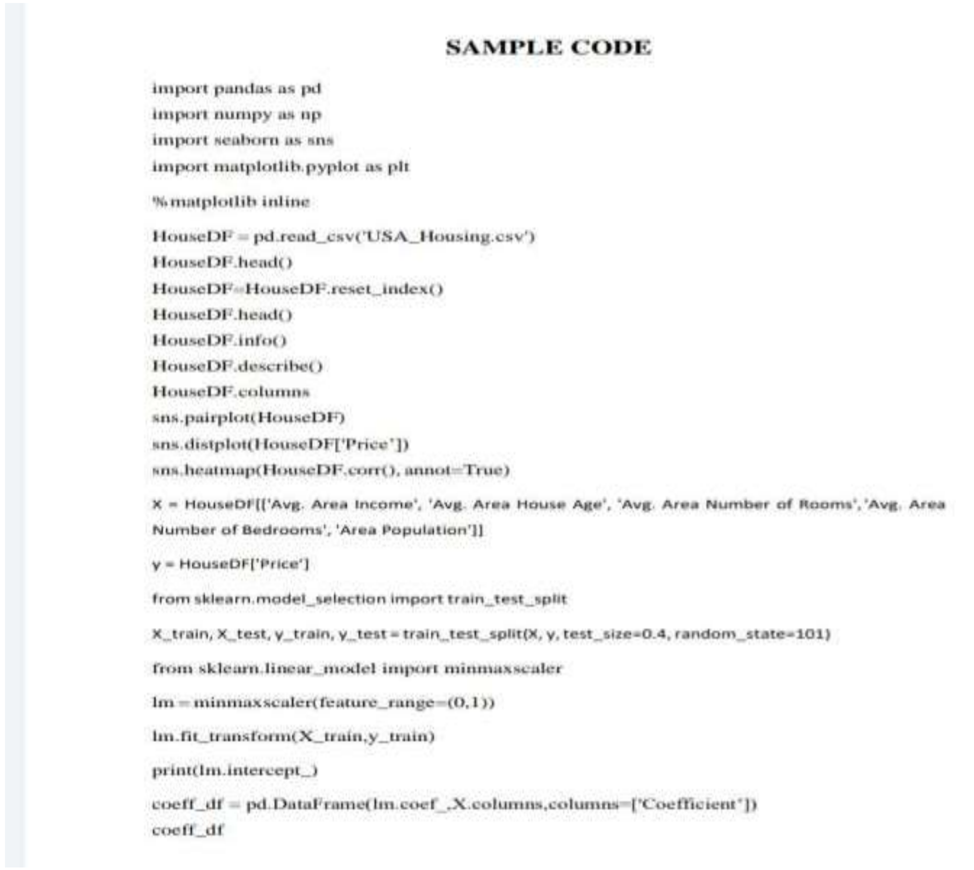
- We used a stack ensemble model to improve the model's performance. A stack ensemble model combines the predictions of multiple machine learning models to produce a more accurate prediction.

Conclusion

- We have developed a machine learning model that can accurately predict house prices, taking into account all relevant factors.

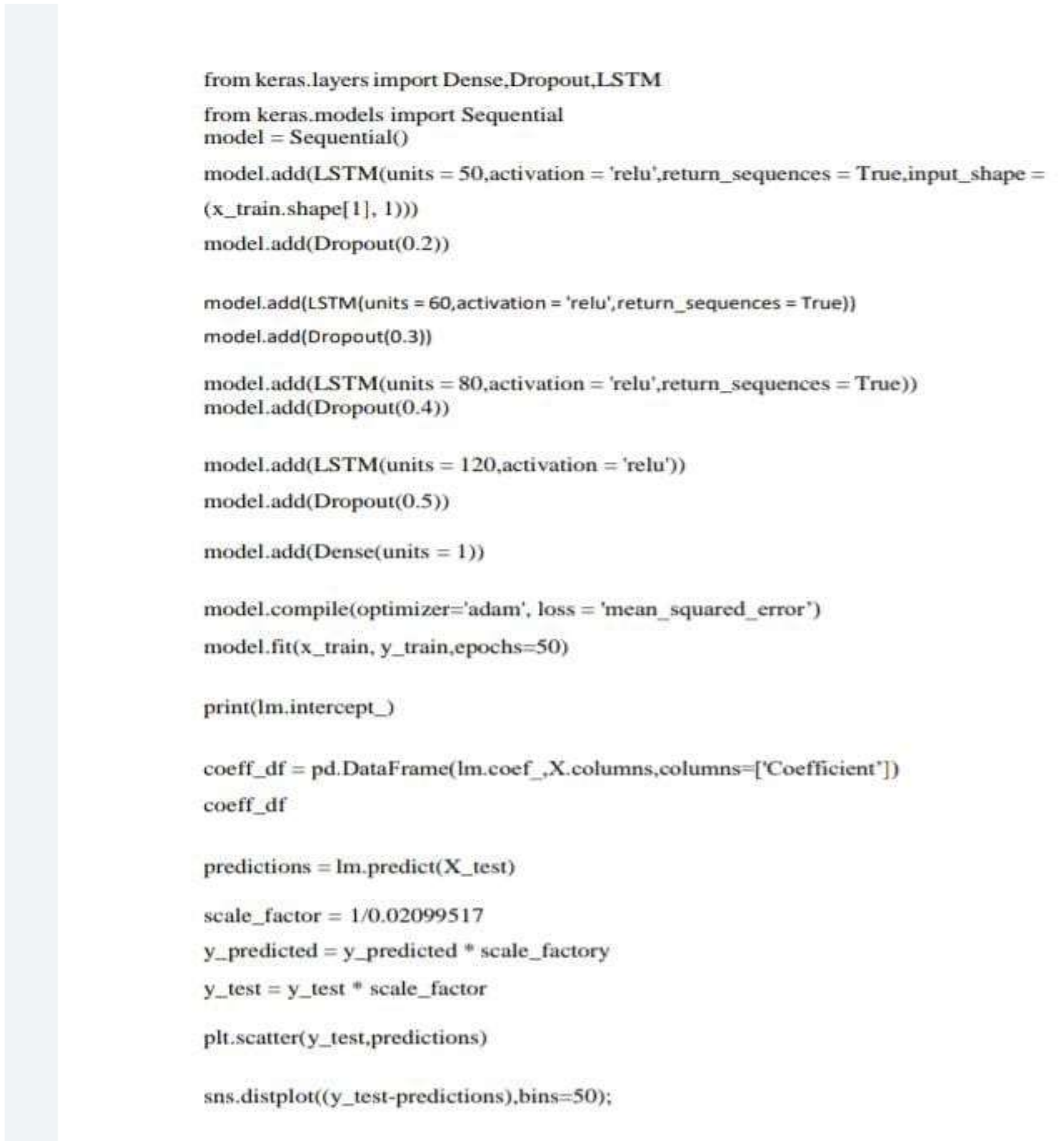
- The model is already being used to help homeowners and real estate agents make more informed decisions about buying and selling homes.

Predicting House Prices - Program



SAMPLE CODE

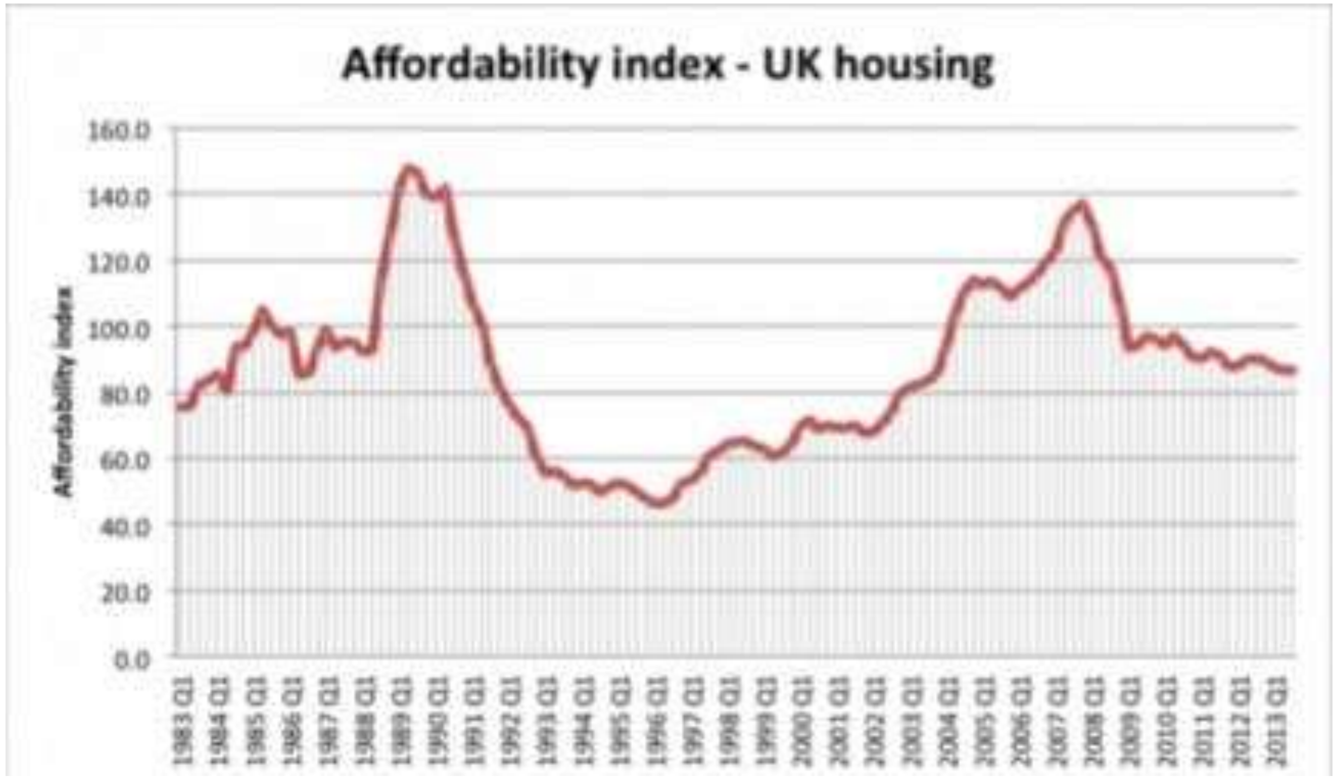
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
HouseDF = pd.read_csv('USA_Housing.csv')
HouseDF.head()
HouseDF=HouseDF.reset_index()
HouseDF.head()
HouseDF.info()
HouseDF.describe()
HouseDF.columns
sns.pairplot(HouseDF)
sns.distplot(HouseDF['Price'])
sns.heatmap(HouseDF.corr(), annot=True)
X = HouseDF[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms','Avg. Area
Number of Bedrooms', 'Area Population']]
y = HouseDF['Price']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=101)
from sklearn.linear_model import minmaxscaler
lm = minmaxscaler(feature_range=(0,1))
lm.fit_transform(X_train,y_train)
print(lm.intercept_)
coeff_df = pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient'])
coeff_df
```



```
from keras.layers import Dense,Dropout,LSTM
from keras.models import Sequential
model = Sequential()
model.add(LSTM(units = 50,activation = 'relu',return_sequences = True,input_shape =
(x_train.shape[1], 1)))
model.add(Dropout(0.2))

model.add(LSTM(units = 60,activation = 'relu',return_sequences = True))
model.add(Dropout(0.3))

model.add(LSTM(units = 80,activation = 'relu',return_sequences = True))
model.add(Dropout(0.4))

model.add(LSTM(units = 120,activation = 'relu'))
model.add(Dropout(0.5))

model.add(Dense(units = 1))

model.compile(optimizer='adam', loss = 'mean_squared_error')
model.fit(x_train, y_train,epochs=50)

print(lm.intercept_)

coeff_df = pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient'])
coeff_df

predictions = lm.predict(X_test)

scale_factor = 1/0.02099517
y_predicted = y_predicted * scale_factory
y_test = y_test * scale_factor

plt.scatter(y_test,predictions)

sns.distplot((y_test-predictions),bins=50);
```

```
plt.figure(figsize=(12,6))
plt.plot(y_test,'b',label = 'Original Price')
plt.plot(y_predicted,'r',label = 'Predicted Price')
plt.xlabel('Time')
plt.ylabel('Price')
plt.legend()
plt.show()

from sklearn import metrics

print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

Output



Affordability index - UK housing
Affordability index
160.0
140.0
120.0
100.0
80.0
60.0
40.0
20.0
0.0
1983 Q1
1984 Q1
1985 Q1
1986 Q1
1987 Q1
1988 Q1
1989 Q1
1990 Q1
1991 Q1
1992 Q1
1993 Q1
1994 Q1
1995 Q1
1996 Q1
1997 Q1
1998 Q1
1999 Q1
2000 Q1
2001 Q1
2002 Q1
2003 Q1
2004 Q1
2005 Q1
2006 Q1
2007 Q1
2008 Q1
2009 Q1
2010 Q1
2011 Q1
2012 Q1
2013 Q1

Thank You