Veena Mounika Ruddarraju (ID: 220269453)

Yarraguntla Saiswetha (ID:220268894)

CSc 215-01 Artificial Intelligence

Project #3

Due D Mini-Project 3: House Price Estimation using Both Visual and Textual Data

Date: 10/23/2019

Problem Statement

Predicting House Price is quiet an interesting and popular project that people work on. The project we did will differentiate the conventional project by introducing the visual data into the dataset. The goal of our project is to build a novel automatic house price estimation system by using both textual and visual inputs, other than only using textual information such as area, neighborhood, and number of bedrooms.

Methodology

We have used Google Colab platform for this project. Initially, we started data preprocessing for our textual data. We considered houses with price < 50000 and price>900000 as outliers, and removed the outliers from the data. Then we are reduced from 535 rows to 453 rows. At this moment we collected the outliers index data as outliers_index for future use. We then checked if there are any missing values in the data and found that there are no missing values in the data. In our textual data, we have a column called ‘Zipcode’ which contains categorical values, so we performed one hot encoding on that column. All the remaining columns contains numeric values. Except the output variable, we performed normalization using z-score for the remaining input variables. After completing all the data preprocessing steps on the textual data, we split the textual data into 80% training data and 20% testing data. We have prepared the textual data to give as an input to the dense layer.

After handling textual data, we moved on to visual data. In visual data, we have 4 different images of each house. They are images of bedroom, bathroom, kitchen and frontal image of the house. With the help of professor’s instructions, we have converted all the image data into a dataframe called ‘img’ that contains four series which represents each kind of image of a house. Then we removed images of the outliers from ‘img’ with the help of outlier_index. Then we are reduced from 535 rows to 453 rows in the ‘img’ dataframe. After removing outliers, we have created 2 different dataframes called ‘train_img’ and ‘test_img’ for training and testing images based on training and testing value indexes. Then we have considered 4 images of each house in both training and testing image datframes and appended them into a single image. After doing these steps we have converted the image dataframes to numpy arrays that could be given to CNN layer to train the model.

In this project, the model in trained using both textual and visual data. The dense layer takes the textual input and the CNN layer gets the visual input. The output of CNN layer is flattened and it along with the dense layer output is concatenated, later given as input to another dense layer which gives the output of predicted house price. Then we build a model with inputs as cnn_input and dnn_input with output as the final layer output. Then we compile it and start training it. While training we give both textual and image training data and output be the price, using the optimizer and activation function on the layers. We used early stopping and model checkpoint for better result, then calculated the MSE and RMSE values and also plotted the lift chart.

Experimental Results and Analysis:
Model Summary

```
Model: "model_1"
___________________________________________________________________________
_______________________
Layer (type)                    Output Shape         Param #     Connected 
to
===========================================================================
=======================
input_2 (InputLayer)            (None, 128, 128, 3)  0
___________________________________________________________________________
_______________________
conv2d_1 (Conv2D)               (None, 125, 125, 32) 1568
input_2[0][0]
___________________________________________________________________________
_______________________
max_pooling2d_1 (MaxPooling2D)  (None, 62, 62, 32)   0
conv2d_1[0][0]
___________________________________________________________________________
_______________________
input_1 (InputLayer)            (None, 44)           0
___________________________________________________________________________
_______________________
conv2d_2 (Conv2D)               (None, 59, 59, 16)   8208
max_pooling2d_1[0][0]
___________________________________________________________________________
_______________________
dense_1 (Dense)                 (None, 10)           450
input_1[0][0]
___________________________________________________________________________
_______________________
max_pooling2d_2 (MaxPooling2D)  (None, 29, 29, 16)   0
conv2d_2[0][0]
___________________________________________________________________________
_______________________
dense_2 (Dense)                 (None, 20)           220
dense_1[0][0]
___________________________________________________________________________
_______________________
flatten_1 (Flatten)             (None, 13456)        0
max_pooling2d_2[0][0]
```

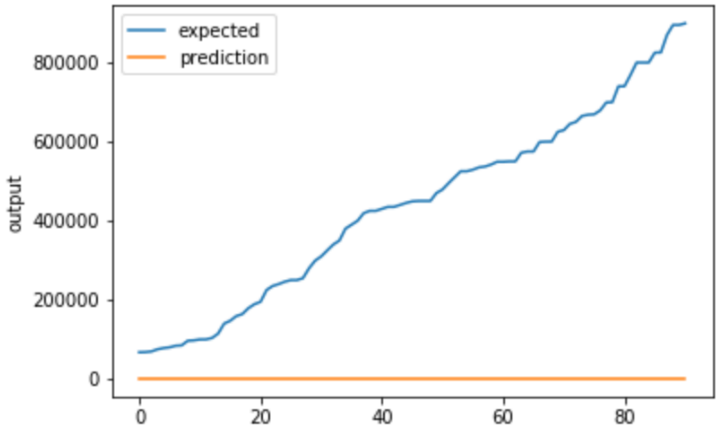
```
__________________________________________________________________________
________________________
dense_3 (Dense)                 (None, 10)           210
dense_2[0][0]
__________________________________________________________________________
________________________
dense_5 (Dense)                 (None, 10)           134570
flatten_1[0][0]
__________________________________________________________________________
________________________
dense_4 (Dense)                 (None, 1)            11
dense_3[0][0]
__________________________________________________________________________
________________________
dense_6 (Dense)                 (None, 1)            11
dense_5[0][0]
__________________________________________________________________________
________________________
concatenate_1 (Concatenate)     (None, 2)            0
dense_4[0][0]

dense_6[0][0]
__________________________________________________________________________
________________________
dense_7 (Dense)                 (None, 4)            12
concatenate_1[0][0]
__________________________________________________________________________
________________________
dense_8 (Dense)                 (None, 1)            5
dense_7[0][0]
==========================================================================
========================
Total params: 145,265
Trainable params: 145,265
Non-trainable params: 0
_________________________
```

MSE and RMSE values:

```
Final score (MSE): 247060354790.2901
Final score (RMSE): 497051.66209388146
```

Plot:



Task Division and Project Reflection:

Veena Mounika: Data Preprocessing

Swetha: Building model

Challenges:

After removing the outliers in the textual data and image data we faced a challenge like how to match the textual data with the image data

What we have learned from this project?

1. We learnt how to define complex models using functional API like multiple input and multiple output.

2. In this project we have learnt how to train the model using both textual and visual data.

3. We learnt how to work in google collab platform.