Project Report

Introduction:

Title

Go to this link http://www.car.org/marketdata/data/housingdata/ and download the Excel file in the link titled, **'Median Prices of Existing Detached Homes.'** Based on the available data, predict the house price in various counties using linear regression. Plot house prices for two counties in separate graphs and also include the corresponding linear regression lines in the graphs.

Background:

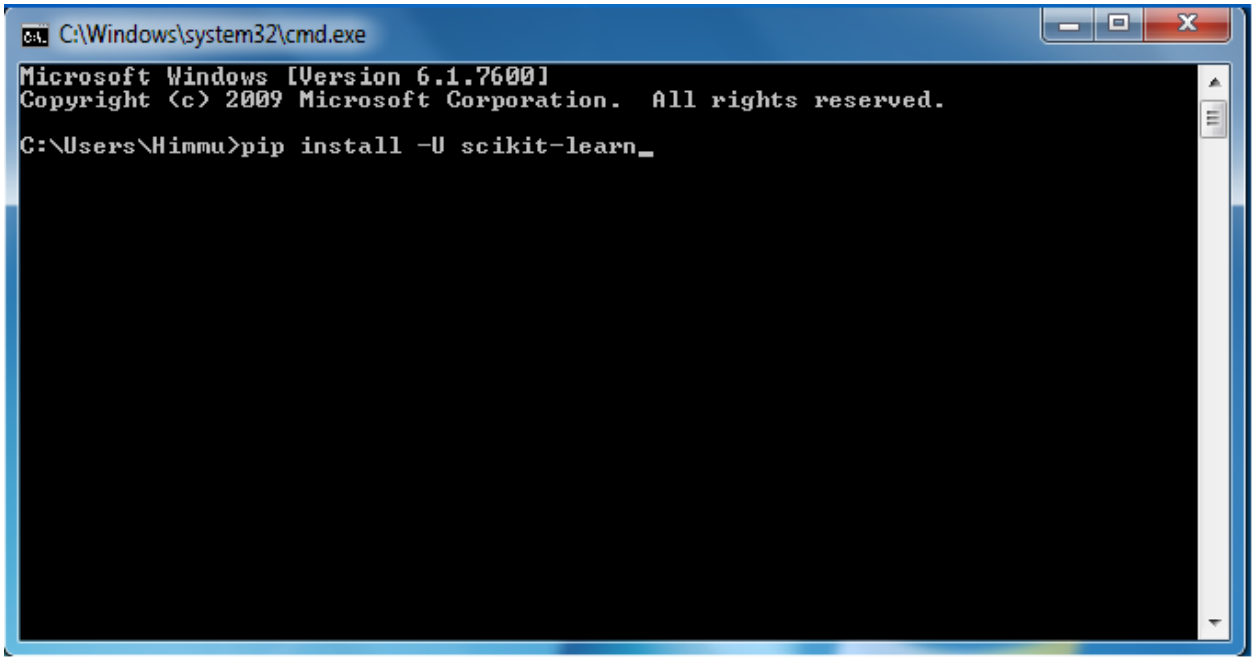
About Project

In this project, we will download the Excel File mention in the link. Save the file in the form CSV file format with extension (.csv). I have downloaded the file from the link and saved as: "HousepricesinCalifornia.csv". I will be using data from this particular file to execute advance functions as required. In this program I will compare house prices of two counties and plotting them on the graph (scattered). Afterwards I will predict the House prices after few years. I am going to fix the prediction value in this particular program. To perform all this functions different modules of Python.

Requirements:

- ✓ Enthought Canopy: We need to Download and Install it. Enthought Canopy is a comprehensive Python analysis environment that provides easy installation of over 450 core scientific analytic and Python packages, creating a robust platform you can explore, develop, and visualize on. When we open any python program using Canopy it will opened in "jupyter IPython Notebook". IPython offers a combination of convenient shell features, special commands and a history mechanism for both input (command history) and output (results caching, similar to Mathematic). It is intended to be a fully compatible replacement for the standard Python interpreter, while offering vastly improved functionality and flexibility. I will be using Python 2.7.11 | 32-bit.

- ✓ Scikit-learn : Scikit-learn is tested to work under Python 2.6, Python 2.7, and Python 3.4. (using the same codebase thanks to an embedded copy of six). It should also work with Python 3.3.



-Installation of scikit-learn using pip

We will be using other modules as well which is already installed. It in inbuilt with the Canopy .

- ✓ Numpy
- ✓ CSV
- ✓ Matploylib

Description:

Firstly, we need to import all the modules that will be required to run the program. So I imported different modules like:

- import csv
- import matplotlib.pyplot as plt
- import numpy as np
- from sklearn import linear_model
- import re

Then I am defining a function named predict_prices. In this function I will be predicting the Prices for the houses. Using linear Regression we can predict the prices. X_test will give the points for the linear line. Y_predict is used to predict the future points. I have fixed the Year for which we can predict the prices. plt.scatter is used is to draw a scatter graph of the given values in X axis and y axis. plt .plot is used to draw a graph with linear line. plt.title and plt.ylabel is used for labeling of graphs.

Now, we are giving 3 empty lists (lst1, lst2, lst3). Using the Dictreader we will open a file " House pricesinCalifornia2.csv". With the help of append function we will be inputting 3 column from the excel sheet and will be storing them in lst1, lst2, lst3. Now will be removing all the special characters using regular expression. Save the value of formatted lst in lst5 and lst6. Next step we have to convert the formatted lst into numeric format. With the help of list comprehension method, we will extract the length of a lst2 which is the variation ofMon-year of the house prices. Save the lst into new lst7. Convert the lst7 into matrix and save it in new lst8. Then take its transpose as it is required for plotting.

In the end we have to call the function named Predict_prices so it can take the imput and print the result. plt.show() is used to print the graph.

Screenshots:

```
Formated List of Prices in Alameda =  ['226149 ', '219306 ', '225162 ', '229333 ', '232291 ', '231250 ', '232916 ', '228467 ',
 '225000 ', '221093 ', '226126 ', '218161 ', '216532 ', '222896 ', '221768 ', '233192 ', '239885 ', '229129 ', '234042 ', '2315
51 ', '224811 ', '221656 ', '221098 ', '228195 ', '212500 ', '213194 ', '222837 ', '226946 ', '221220 ', '229885 ', '222153 ',
 '223511 ', '219496 ', '214925 ', '217514 ', '222321 ', '209583 ', '201699 ', '216313 ', '219999 ', '219897 ', '228279 ', '2154
08 ', '223888 ', '212005 ', '208918 ', '215714 ', '207994 ', '207822 ', '204389 ', '209117 ', '217890 ', '205494 ', '216705 ',
 '205912 ', '203947 ', '211482 ', '207390 ', '203550 ', '208654 ', '194528 ', '199258 ', '198888 ', '204610 ', '209335 ', '2140
90 ', '212316 ', '211904 ', '210893 ', '207414 ', '206743 ', '204362 ', '201739 ', '205250 ', '208046 ', '210130 ', '213709 ',
 '213414 ', '222612 ', '218452 ', '218251 ', '220375 ', '216186 ', '211695 ', '213461 ', '206818 ', '207143 ', '230023 ', '2303
87 ', '231725 ', '240294 ', '234529 ', '234023 ', '239244 ', '239999 ', '234375 ', '229687 ', '234772 ', '239595 ', '259606 ',
 '255440 ', '267806 ', '267601 ', '263517 ', '257258 ', '261234 ', '253846 ', '253846 ', '248195 ', '245121 ', '272404 ', '2903
34 ', '301751 ', '313474 ', '313182 ', '313481 ', '285163 ', '312879 ', '302842 ', '292804 ', '323795 ', '327747 ', '348034 ',
 '371064 ', '369383 ', '391288 ', '372433 ', '359120 ', '361250 ', '393343 ', '406364 ', '410952 ', '391303 ', '400962 ', '4215
42 ', '426373 ', '394444 ', '405389 ', '404872 ', '408836 ', '385210 ', '388753 ', '389887 ', '394957 ', '390576 ', '405181 ',
 '403448 ', '455749 ', '447773 ', '471222 ', '464999 ', '460958 ', '459832 ', '452391 ', '450990 ', '453079 ', '441025 ', '4390
80 ', '471632 ', '471715 ', '471005 ', '488474 ', '486363 ', '483883 ', '483461 ', '478347 ', '485191 ', '491495 ', '489603 ',
 '530746 ', '528871 ', '558392 ', '591700 ', '589285 ', '595273 ', '586409 ', '590576 ', '604464 ', '599397 ', '604392 ', '5843
60 ', '619320 ', '649200 ', '659260 ', '665320 ', '681540 ', '678650 ', '676080 ', '666730 ', '665100 ', '671490 ', '663480 ',
 '645600 ', '663950 ', '663640 ', '686820 ', '682690 ', '685580 ', '677310 ', '673460 ', '659110 ', '658650 ', '643840 ', '6590
40 ', '651220 ', '655090 ', '684310 ', '692900 ', '709420 ', '694780 ', '696330 ', '703180 ', '662560 ', '658420 ', '659550 ',
 '616220 ', '561360 ', '556300 ', '569370 ', '537990 ', '528890 ', '536310 ', '526000 ', '470250 ', '427040 ', '405400 ', '3902
70 ', '368240 ', '143587 ', '327370 ', '316440 ', '337100 ', '374850 ', '405260 ', '397460 ', '392260 ', '432440 ', '427180 ',
 '431980 ', '408590 ', '383330 ', '410230 ', '432720 ', '407830 ', '472370 ', '474520 ', '458180 ', '440480 ', '424770 ', '4446
70 ', '434420 ', '412210 ', '363750 ', '392660 ', '420420 ', '390960 ', '415830 ', '415300 ', '420150 ', '407420 ', '390440 ',
 '391710 ', '396990 ', '383430 ', '340600 ', '360670 ', '372220 ', '412590 ', '430340 ', '459760 ', '473600 ', '466760 ', '4541
70 ', '460670 ', '467840 ', '463010 ', '466830 ', '451580 ', '523280 ', '573000 ', '593160 ', '604600 ', '614810 ', '607240 ',
 '604100 ', '588600 ', '607900 ', '585440 ', '559110 ', '572630 ', '623110 ', '656140 ', '680020 ', '695210 ', '688870 ', '6915
00 ', '652610 ', '659660 ', '670320 ', '645680 ', '610130 ', '651950 ', '713060 ', '742340 ', '763740 ', '750000 ', '771460 ',
 '741460 ', '739580 ', '711990 ', '736870 ', '738790 ', '711900 ', '712990 ', '762570 ', '820080 ', '828000 ', '803000 ']
```

-**List of prices in Alameda**

```
Formated List Pf Prices In Butte =  ['102143 ', '83333 ', '100000 ', '108000 ', '100000 ', '123043 ', '113571 ', '113333 ', '11
7499 ', '127143 ', '117999 ', '125333 ', '118332 ', '121333 ', '123000 ', '118749 ', '127894 ', '120714 ', '113571 ', '120500
 ', '117856 ', '98333 ', '120625 ', '124545 ', '122308 ', '128888 ', '115454 ', '122692 ', '117368 ', '122609 ', '115555 ', '11
1578 ', '120588 ', '118124 ', '127647 ', '109500 ', '113333 ', '113999 ', '110999 ', '114999 ', '129166 ', '107500 ', '114210
 ', '124762 ', '123600 ', '119473 ', '114166 ', '108888 ', '122143 ', '115999 ', '114705 ', '124545 ', '128125 ', '117499 ', '1
17894 ', '114799 ', '115499 ', '122000 ', '111249 ', '121818 ', '110000 ', '123333 ', '123333 ', '122000 ', '106428 ', '108571
 ', '116922 ', '113529 ', '118888 ', '118076 ', '114615 ', '122222 ', '114210 ', '114210 ', '118234 ', '111363 ', '115861 ', '1
18749 ', '126111 ', '119499 ', '120000 ', '115428 ', '119166 ', '113124 ', '127500 ', '120000 ', '110000 ', '110454 ', '109130
 ', '115789 ', '122692 ', '120556 ', '113999 ', '115651 ', '113157 ', '117599 ', '111249 ', '109333 ', '113076 ', '118124 ', '1
19696 ', '121905 ', '123600 ', '119393 ', '124545 ', '122069 ', '120690 ', '121500 ', '110499 ', '125555 ', '124074 ', '126428
 ', '121111 ', '128750 ', '125238 ', '120645 ', '123809 ', '133870 ', '127333 ', '126071 ', '120000 ', '113076 ', '129655 ', '1
24400 ', '130666 ', '133333 ', '140952 ', '138332 ', '136538 ', '148181 ', '136249 ', '148095 ', '142222 ', '142308 ', '149166
 ', '141579 ', '149600 ', '153636 ', '154374 ', '153103 ', '158635 ', '149375 ', '154799 ', '152631 ', '158332 ', '163684 ', '1
74090 ', '165789 ', '171904 ', '176363 ', '185789 ', '187777 ', '185714 ', '182105 ', '192666 ', '200000 ', '186923 ', '189090
 ', '216428 ', '209259 ', '225581 ', '211363 ', '233749 ', '221078 ', '230920 ', '239903 ', '235714 ', '221969 ', '238970 ', '2
35483 ', '237244 ', '244999 ', '269230 ', '266827 ', '251744 ', '273913 ', '273611 ', '273125 ', '279999 ', '276744 ', '273148
 ', '277884 ', '297825 ', '317000 ', '317045 ', '324528 ', '327884 ', '340000 ', '336538 ', '348529 ', '329411 ', '325714 ', '2
97618 ', '318750 ', '337805 ', '323684 ', '327631 ', '357777 ', '336842 ', '317021 ', '344594 ', '320513 ', '297999 ', '307407
 ', '292104 ', '317187 ', '326923 ', '322916 ', '297916 ', '326250 ', '323684 ', '293548 ', '294230 ', '288749 ', '296874 ', '2
97058 ', '303125 ', '271875 ', '256250 ', '279999 ', '288461 ', '288749 ', '261363 ', '272222 ', '251471 ', '256579 ', '268750
 ', '255952 ', '143587 ', '238157 ', '247499 ', '238888 ', '242307 ', '258000 ', '267708 ', '227631 ', '245832 ', '236110 ', '2
55555 ', '245237 ', '250000 ', '219230 ', '250000 ', '238000 ', '236760 ', '229464 ', '235710 ', '243420 ', '225000 ', '244050
 ', '208930 ', '230000 ', '180000 ', '190000 ', '222370 ', '217040 ', '222620 ', '221050 ', '196430 ', '201320 ', '197690 ', '1
86150 ', '207950 ', '198750 ', '194000 ', '177860 ', '200000 ', '197620 ', '211290 ', '221050 ', '215740 ', '226090 ', '207140
 ', '210940 ', '223860 ', '236460 ', '208930 ', '211760 ', '250000 ', '234000 ', '238300 ', '265000 ', '239280 ', '281820 ', '2
50000 ', '248330 ', '255950 ', '231100 ', '227500 ', '217650 ', '221550 ', '231670 ', '237880 ', '245000 ', '276140 ', '241450
 ', '235710 ', '233780 ', '241910 ', '231820 ', '231730 ', '247320 ', '233930 ', '253850 ', '261110 ', '253660 ', '269290 ', '2
63640 ', '247320 ', '265240 ', '238540 ', '261670 ', '260580 ', '270830 ', '283870 ', '272220 ', '275000 ', '277980 ']
```

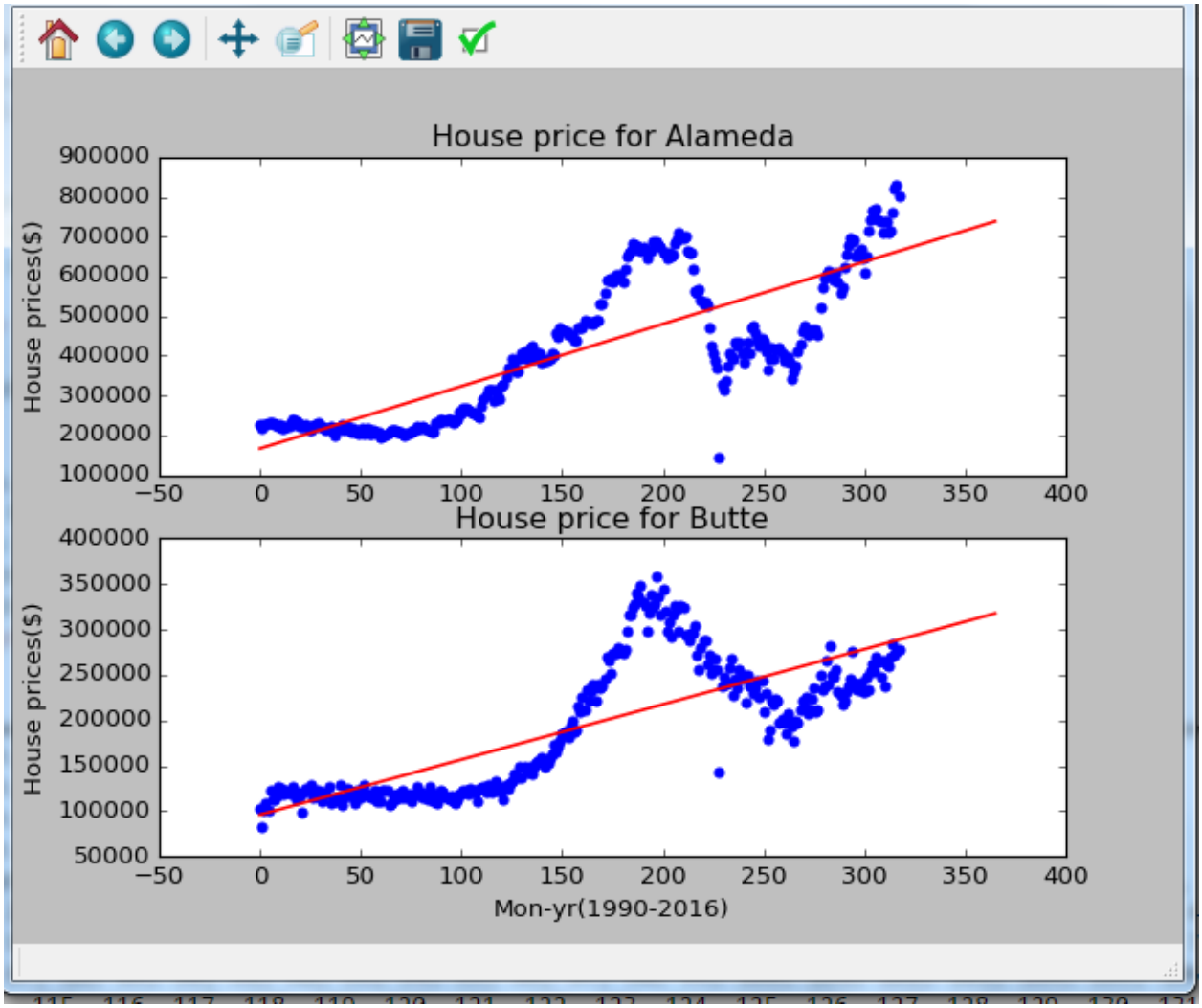
-List of House prices in butte

```
Month-Year varriation on x-axis for both counties [[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51,
 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82,
 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 1
11, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 1
36, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 1
61, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 1
86, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 2
11, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 2
36, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 2
61, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 2
86, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 3
11, 312, 313, 314, 315, 316, 317]]
```

-Time representation from 1990-2016 in one month intervals

```
Predited value after 4 years in Alameda [ 738376.85377743]
Predited value after 4 years in Butte [ 317596.90626547]
```

-Predicted values for both counties after 4 years



-Graph representation

Conclusion:

In this program, I have compare the two counties on the basis of their House prices using different python modules and plotted the scattered graph of it. Using liner regression I have drawn the linear line and predicted the value of house in further 4 years.

Program Code:

```
#importing of Python Modules Required
import csv
import matplotlib.pyplot as plt
matplotlib inline
import numpy as np
from sklearn import linear_model
import re
#function
def predict_price(Matrix1,y,lst8,subplotInstance,titleInstance):
    lm=linear_model.LinearRegression()
    plt.subplot(2,1,subplotInstance)
    lm.fit(Matrix1,y)
    x_test=[[0],[200],[341],[365]]
    y_predict=lm.predict(x_test)
    x_test2=[[365]]
    y_predict2=lm.predict(x_test2)
    print "Predited value after 4 years in " +  titleInstance ,y_predict2
    plt.scatter(lst8,y,color='blue')
    plt.plot(x_test,y_predict,color='red',linewidth=1.5)
    plt.title('House price for ' +  titleInstance)
    plt.ylabel('House prices($)')
#Main Code
lst1=[]
lst2=[]
```

```
lst3=[]
input_file=csv.DictReader(open("HousepricesinCalifornia2.csv"))
for line in input_file:
   lst1.append(line['Alameda'])
   lst2.append(line['Mon-Yr'])
   lst3.append(line['Butte'])
#print "House prices in Alameda=",lst1
#print "House price in Butte=",lst3
#use of regular expression
lst5 = [re.sub("[a-zA-Z$,]", "", elem) for elem in lst1] #removing of all the variables and special chracters
lst6 = [re.sub("[a-zA-Z$,]", "", elem) for elem in lst3] #removing of all the variables and special chracters
print "Formated List of Prices in Alameda = ",lst5

print "Formated List Pf Prices In Butte = ",lst6

y1=np.array(lst5).astype(np.float)
y2=np.array(lst6).astype(np.float)
#print y1,y2

#use of list comprehension
lst7=[x for x in range (len(lst2))]
#print "month-yr of Alameda",lst7
lst8=[lst7]
print "Month-Year varriation on x-axis for both counties",lst8
```

```
x1= np.asarray(lst8)
#print x1

Matrix1=np.matrix.transpose(x1)
#print "value of Matrix1= ",Matrix1
#call function to predict future price
predict_price(Matrix1,y1,lst8,1, 'Alameda')
predict_price(Matrix1,y2,lst8,2, 'Butte')
plt.xlabel('Mon-yr(1990-2016)')
plt.show()
```