Predicting Future Medical Cost using Linear Regression

Erick Pereyra

CSCI 380 Machine Learning

5/21/2019

Prof. Johnson

Abstract

This project is going to try to accurately predict future medical costs for patients. I will be figuring out which is the dependent and independent variables in the dataset to use on the model. The models that are going to be used to accurately predict the cost is the Linear Regression model and we will be finding the regression score of the model as well. I will be using train_test_split to be able to split the data into training and testing data. Finally, I represent the data in a scatter plot graph.

Introduction

Everyone knows that the cost of medical bills can ruin a person's life. Depending on the circumstances, it can really determine if your medical bill can be at an all-time high. In this project I plan to try and accurately predict patients' future medical costs. It will be a great to have that understanding much more of an understanding with the visualizations that will be provided. This is will give us much more insight on how the much insurance can really cost to someone.

Dataset

For this project, I will be using the Medical Cost Personal Dataset [1]. This dataset has 1338 rows and 7 columns

- sex: Whether the person is a male or female.
- bmi: Body mass index (BMI),is a measure of body fat based on height and weight that applies to males and females. (kg / m ^ 2) using the ratio of height to weight.
- children: How many children they have if any.
- smoker: Smoking / non smoking
- region: In what region the patients live in
- charges: Medical bill costs according to the patients.

Methods and Results

Linear Regression Model

I used sklearn Linear Regression model on the insurance.csv dataset that was provided. I used this method instead because sklearn has great built in functions that can do most of the work for you in a sense. Instead of having to write the algorithms out, since I have a general understanding of what it does, I should be able to interpret the information that is given.

What I did first was to find the correlation between the columns:

```
In [50]: #Trying to see the correlation between the columns
         corr1 = data.corr()
         sns.heatmap(corr1, annot = True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff2305b7dd8>
```

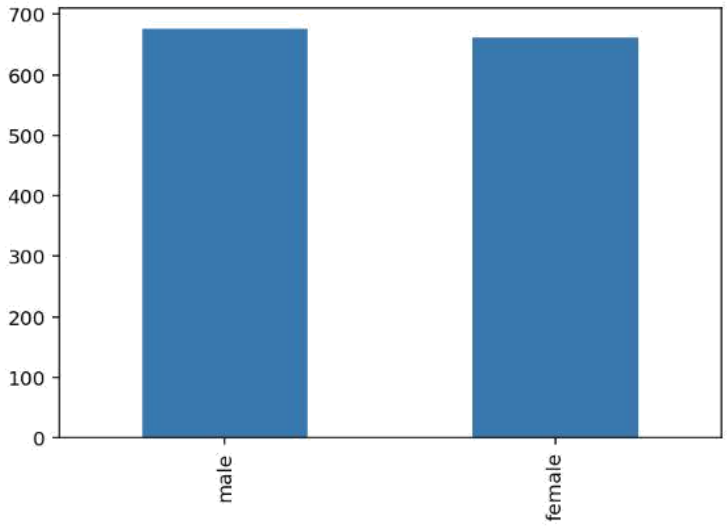
Out[50]:

With this is able to see the 1 to 1 correlations of the columns with black having the least amount of correlation. I wanted to make sure the sex column wasn't skewed:

In [6]:
```
#Checking to see if the sex column is skewed or not.
#In this case it's not.

data['sex'].value_counts()
data['sex'].value_counts().plot(kind='bar')
```

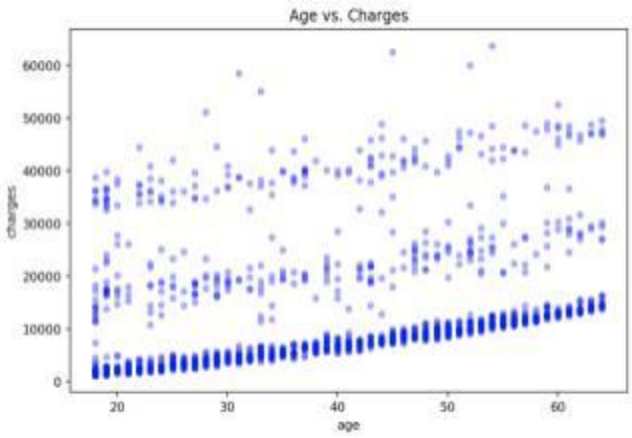
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6f64b79320>
```

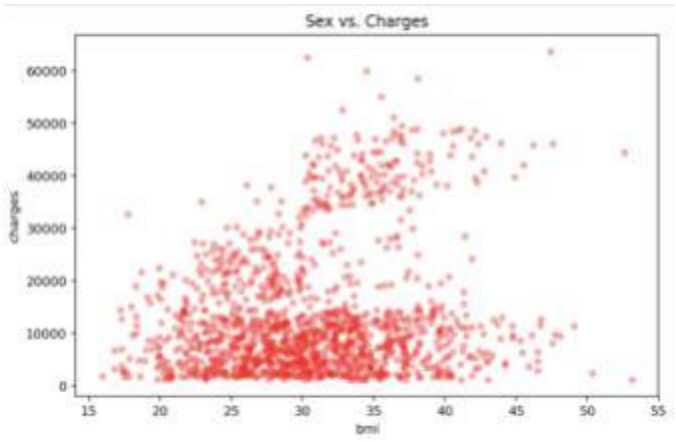


Out[6]:

In this case the column isn't skewed. It has close to the same amount of values together male and female.

After this, I proceeded and made a graph from the data and made a graph of "Age vs Charges" and "Sex vs Charges":





This is an important part of the data because It gives us a nice visualization of the data and it gives us a better understanding of the trend of the data. The graph on the left is showing Age vs Charges. As you can see that the charges seem to increase as the patient is older. In our graph its showing the charges based of the patient's BMI. What I tried to achieve here was try

and see if a patient's bmi had something to do with the charges. Most of the charges occur within 20 - 40 BMI and the $20,000 range.

Since there is columns that contain just strings and not integers, I ended up converting them by using panda's get_dummies () function. This is a type of encoding to be able to generate values for a dataset. In my case, it replaced columns with Nan values, with dummy categories with a value instead of a NaN.

In [8]:
```
obj_data = pd.get_dummies(obj_data)
obj_data.head()
```
0.066 seconds

Out[8]:

sex_female	sex_male	smoker_no	smoker_yes	region_northeast	region_northwest	region_southeast	region_sc
1	0	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	0	0	0	1	0
0	1	1	0	0	1	0	0
0	1	1	0	0	1	0	0

Train_test_split

Why should we use cross validation? We use this to try and prevent overfitting of the data. We want to prevent overfitting or underfitting because you don't want the data to train too well or too poorly. It won't give us an accurate probability reading of our model if any of the two occur.

I created an X and Y variables, stored all the columns except "charges" into X and only "charges into Y. I ran sklearn train_test_split () function to split the data into training and testing data used the Linear Regression model and got an R^2score of 0.7394054641456173which is approximately 74%.

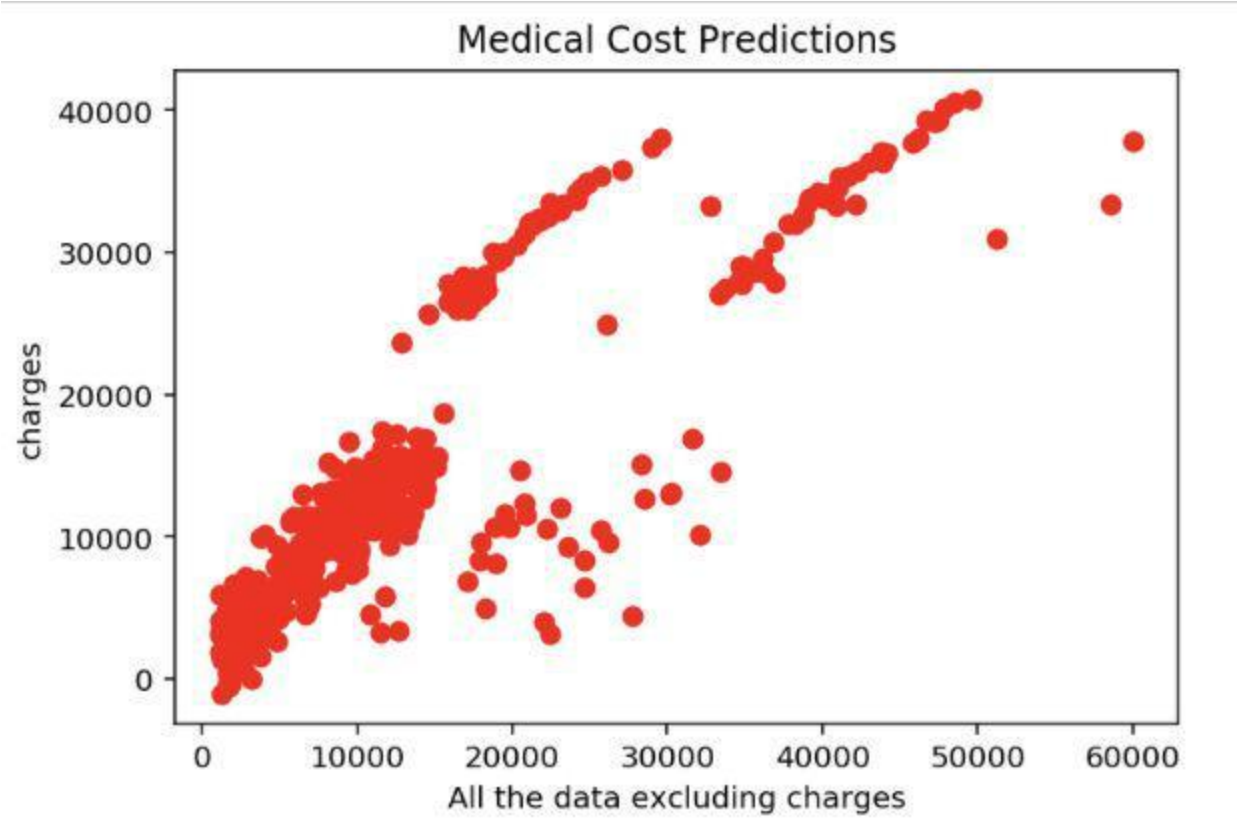
In [14]:
```
X = data.drop('charges', axis =1).values
Y = data[['charges']].values

x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size = 0.33, random_state = 1)
reg = LinearRegression()
reg.fit(x_train,y_train)
R2 = reg.score(x_test,y_test)
g = reg.predict(x_test)
MSE = mean_squared_error(g, y_test)

print('The Mean Square Error(MSE) is: ', MSE)
print('R2 score is %', R2* 100)
```

```
The Mean Square Error(MSE) is:  35585029.808486596
R2 score is % 73.94054641456172
```

At this point I was ready to plot the graph of the predictions that were made.



As you can see from this graph the predictions of the medical costs for patient's are increasing using PCA. This is should be an accurate representation of the predicted data.

Conclusion

This is was an interesting dataset to work on. As there is a rise of increases of medical debt, this showed how costs for datasets group of patient's debts is increasing. This is scary to think about because of how powerful medical and insurance companies are and how they are increasing the costs. At least by using Linear Regression on this dataset, we can see a simple representation of the data. There are other ways to represent the data that is given to us. Apply some on these methods to hospital records and you can see how powerful machine learning can be.

References/ Bibliography

[1] **M. Choi, "Medical Cost Personal Datasets,"** ***Kaggle*****,21-Feb-2018. [Online]. Available: https://www.kaggle.com/mirichoi0218/insurance. [Accessed: 20-May-2019].**

[2] J. VanderPlas, "In Depth: Principal Component Analysis," ***In Depth: Principal Component Analysis | Python Data Science Handbook*****.[Online]. Available: https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysi s.html. [Accessed: 20-May-2019].**

[3] Felipe, "One-Hot Encoding a Feature on a Pandas Dataframe: Examples," ***queirozf.com*****,22-Mar-2019. [Online]. Available: http://queirozf.com/entries/one-hot-encoding-a-feature-on-a-pandas-dataframe-an-example . [Accessed: 20-May-2019].**

[4] "pandas.get_dummies¶," ***pandas.get_dummies - pandas 0.17.0 documentation*****.[Online]. Available: https://pandas.pydata.org/pandas-docs/version/0.17.0/generated/pandas.get_dummies.html. [Accessed: 20-May-2019].**

[5] "sklearn.model_selection.train_test_split¶," ***scikit*****.[Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.ht ml. [Accessed: 20-May-2019].**

[6] A. Bronshtein and A. Bronshtein, "Train/Test Split and Cross Validation in Python," ***Towards Data Science*****,17-May-2017. [Online]. Available: https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4 b6. [Accessed: 20-May-2019].**