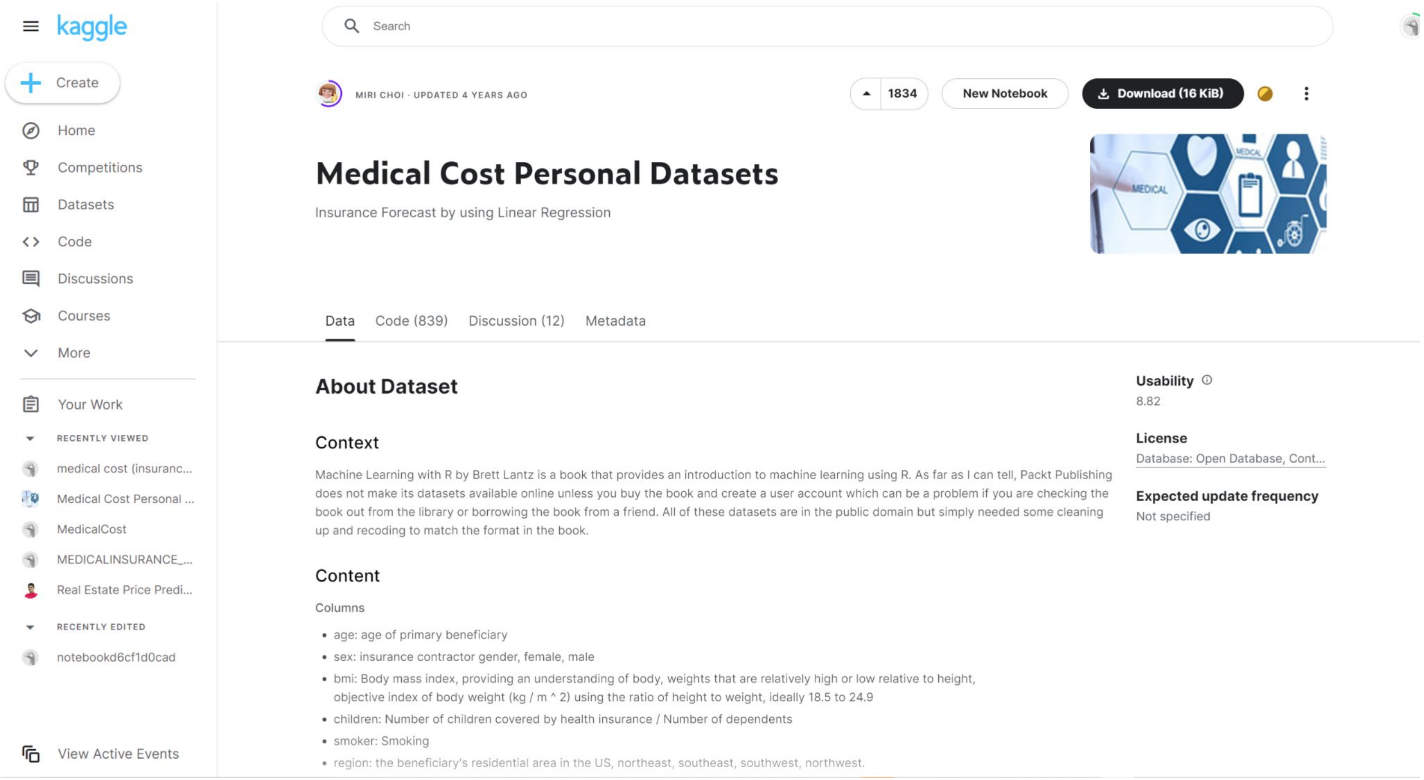
Medical Cost Report

Zhenglei Jiang, Qintian Huang, Sijia Li, Zhangde Song

Dataset



kaggle
Search
Create
Home
Competitions
Datasets
Code
Discussions
Courses
More
Your Work
RECENTLY VIEWED
medical cost (insuranc...
Medical Cost Personal ...
MedicalCost
MEDICALINSURANCE_...
Real Estate Price Predi...
RECENTLY EDITED
notebookd6cf1d0cad
View Active Events
MIRI CHOI · UPDATED 4 YEARS AGO
1834
New Notebook
Download (16 KiB)
Medical Cost Personal Datasets
Insurance Forecast by using Linear Regression
MEDICAL
Data
Code (839)
Discussion (12)
Metadata
About Dataset
Context
Machine Learning with R by Brett Lantz is a book that provides an introduction to machine learning using R. As far as I can tell, Packt Publishing does not make its datasets available online unless you buy the book and create a user account which can be a problem if you are checking the book out from the library or borrowing the book from a friend. All of these datasets are in the public domain but simply needed some cleaning up and recoding to match the format in the book.
Content
Columns
• age: age of primary beneficiary
• sex: insurance contractor gender, female, male
• bmi: Body mass index, providing an understanding of body, weights that are relatively high or low relative to height, objective index of body weight (kg / m ^ 2) using the ratio of height to weight, ideally 18.5 to 24.9
• children: Number of children covered by health insurance / Number of dependents
• smoker: Smoking
• region: the beneficiary's residential area in the US, northeast, southeast, southwest, northwest.
Usability
8.82
License
Database: Open Database, Cont...
Expected update frequency
Not specified

Dataset introduction

- **age:** age of primary beneficiary
- **sex:** insurance contractor gender, female, male
- **bmi:** Body mass index, providing an understanding of body, weights that are relatively high or low relative to height,
  objective index of body weight (kg / m ^ 2) using the ratio of height to weight, ideally 18.5 to 24.9
- **children:** Number of children covered by health insurance / Number of dependents
- **smoker:** Smoking
- **region:** the beneficiary's residential area in the US, northeast, southeast, southwest, northwest.
- **charges:** Individual medical costs billed by health insurance

Goals and Approaches

- Problem we are trying to solve
    a. Find the best model for predicting medical cost
- Approaches
    a. Linear regression
    b. Random forest regressor
    c. K-means clustering
    d. Logistic regression

Data Processing

- Encoding
- Deletions
- Train Test split (80 20)

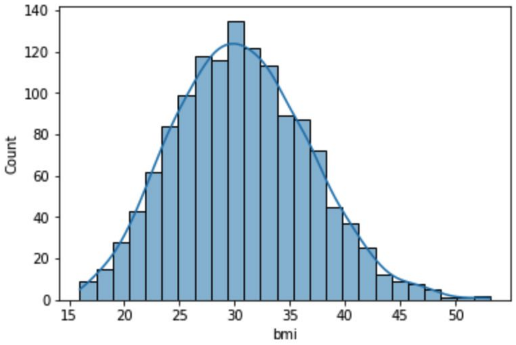
```
[12] # encode sex and smoker
label = preprocessing.LabelEncoder()
df1 = df
df1['sex'] = label.fit_transform(df['sex'])
df1['smoker'] = label.fit_transform(df['smoker'])
df1['region'] = label.fit_transform(df['region'])
df1.head()
```

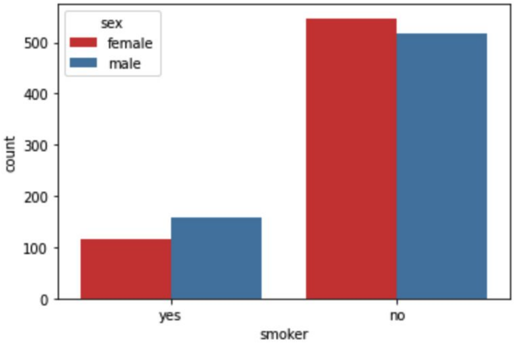
	age	sex	bmi	children	smoker	region	charges
0	19	0	27.900	0	1	3	16884.92400
1	18	1	33.770	1	0	2	1725.55230
2	28	1	33.000	3	0	2	4449.46200
3	33	1	22.705	0	0	1	21984.47061
4	32	1	28.880	0	0	1	3866.85520

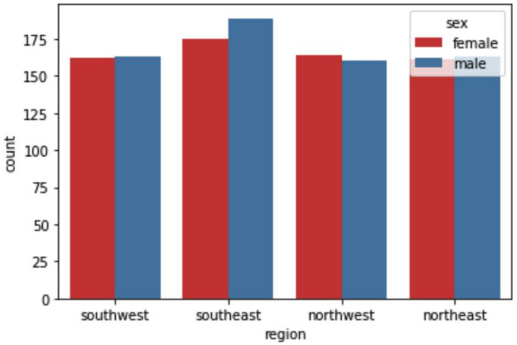
```
# splitting the datasets to train and test
from sklearn.model_selection import train_test_split

train_x,test_x,train_y,test_y = train_test_split(x,y)
```

Data Visualization

- Distribution of BMI
- Distribution of smoker
- Distribution of region

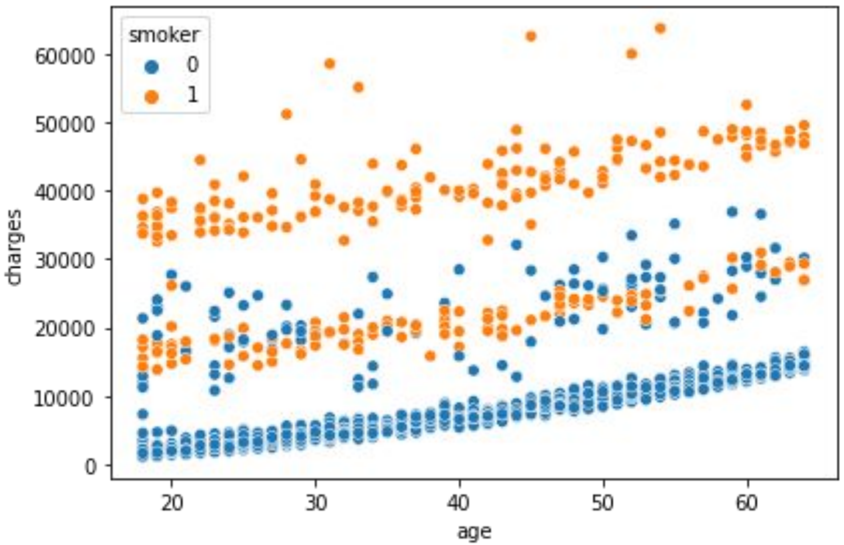


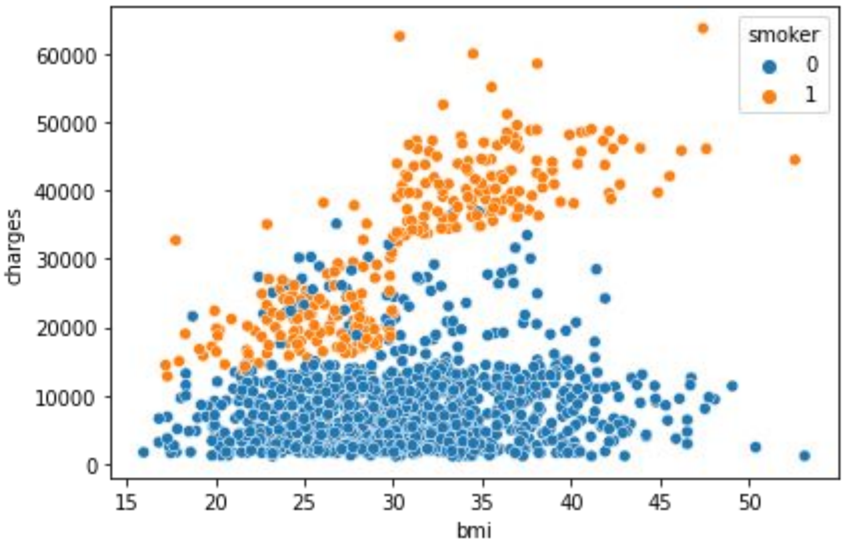




Data Visualization: categorical

- Age Vs. Charges & BMI Vs. charges separated by: Smoker
- Age Vs. Charges & BMI Vs. charges separated by: Sex
- Age Vs. Charges & BMI Vs. charges separated by: Region





Importance

- Age Vs. Charges & BMI Vs. charges separated by: Smoker
- Features are shuffled n times and the model refitted to estimate the importance of it
- In the below case, age (0.42) and bmi (0.33) are the values strongly related to charges

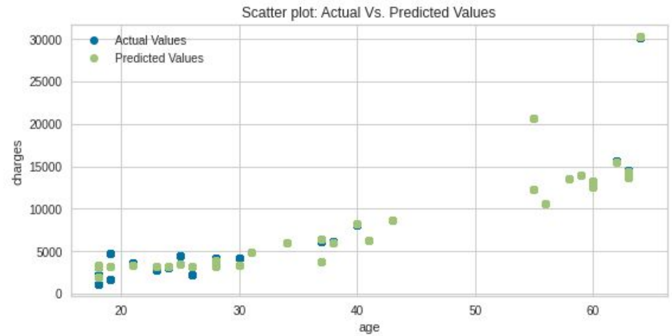
```
[25] fi_col = []
     fi = []

     for i,column in enumerate(df1.drop('charges', axis = 1)):
         print('The feature importance for {} is : {}'.format(column, dt.feature_importances_[i]))

         fi_col.append(column)
         fi.append(dt.feature_importances_[i])
```

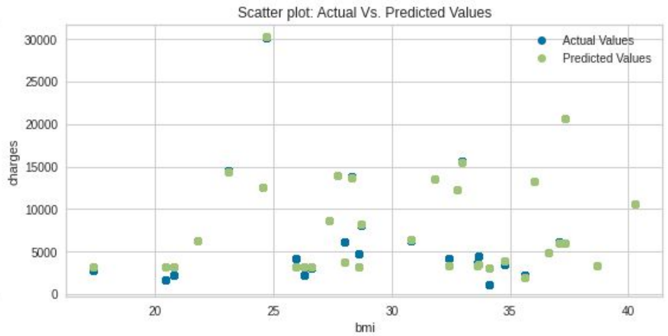
```
The feature importance for age is : 0.42175048342670046
The feature importance for sex is : 0.07534062309601461
The feature importance for bmi is : 0.3333674316537561
The feature importance for children is : 0.06787486959321948
The feature importance for smoker is : 0.019048641638405025
The feature importance for region is : 0.08261795059190426
```

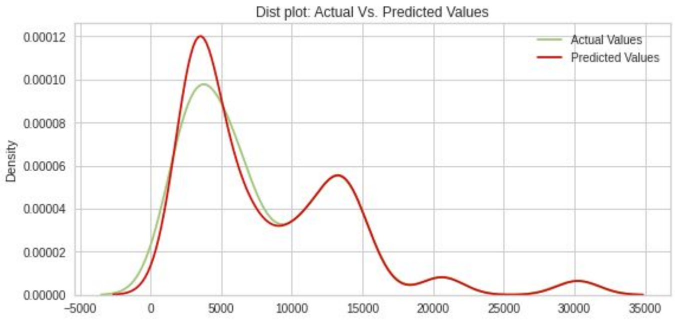
Linear Regression

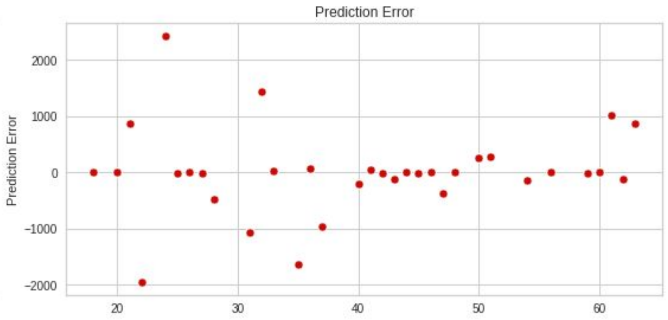
	Variables	Equation	Results	R^2 Value
Simple LR	Age	`y = ax+b`	`y = (192.37157976 * x) + 9455.3300572`	0.04
Simple LR	BMI	`y = ax+b`	`y = (864.12452383 * x) -10471.21685794`	0.11
Multiple LR	Age, BMI	`y = a1x1 + a2x2 + b`	`y = 172.5286568 * x1 - 835.17704282 * x2 -16151.04774268`	0.15
Multiple LR	Age, BMI, Smoker	`y = a1x1 + a2x2 + a3x3+ b`	`y = 320.0978024 * x1 - -22.52480488x2 + 0 * x3 + -3738.48465202`	0.76
Polynomial LR	Age, BMI, Smoker	`y = a1x1^n + a2x2^n + a3x3^n+ b`	/	0.98

Actual VS. Predicted, Predicted Error

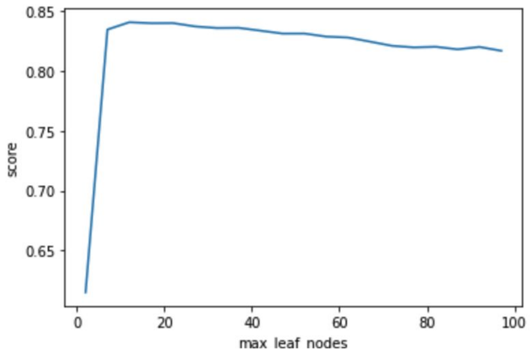


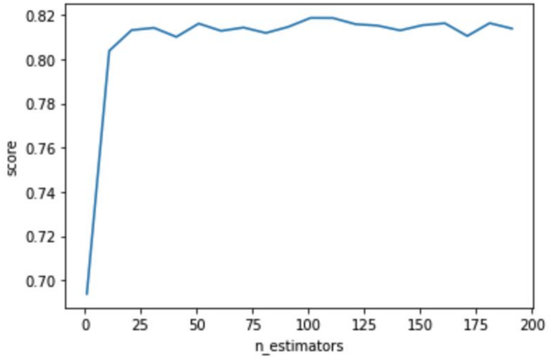


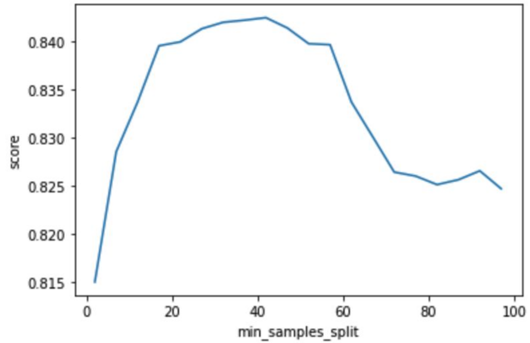


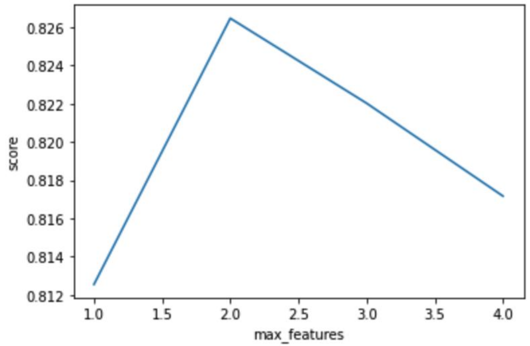


Random Forest Regression - Hyperparameter Tuning










```
# splitting the datasets to train and test
from sklearn.model_selection import train_test_split

train_x,test_x,train_y,test_y = train_test_split(x,y)
```

```
#RandomForestRegressor, using default parameters
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score

rf = RandomForestRegressor()
rf.fit(train_x, train_y)
print(rf.score(test_x, test_y))

predictions = rf.predict(test_x)

print("Mean absolute error: %.2f" % np.mean(np.absolute(predictions - test_y.values)))
print("Residual sum of squares (MSE): %.2f" % np.mean((predictions - test_y.values) ** 2))
print("R2-score: %.8f" % r2_score(test_y.values, predictions))
```

```
0.8108193770151562
Mean absolute error: 2978.12
Residual sum of squares (MSE): 30502273.67
R2-score: 0.81081938
```

RFRegressor - Sample of Prediction

```
#smoker prediction sample
for i in range(5):
  print("test set: " + str(i))
  print(test_y2.iloc[i])
  print(rf2.predict(test_x2)[i])
```

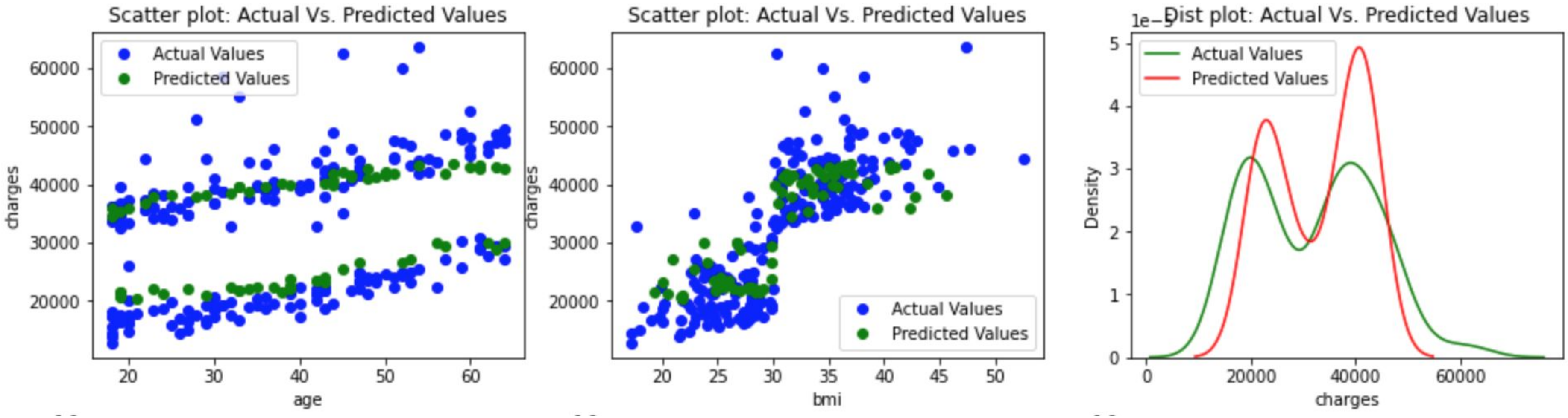
```
test set: 0
30184.9367
25583.0082284321
test set: 1
41919.097
43529.80503150785
test set: 2
35069.37452
22785.66123532648
test set: 3
42856.838
43529.80503150785
test set: 4
48824.45
47282.97624064615
```

```
#non-smoker prediction sample
for i in range(5):
  print("test set: " + str(i))
  print(test_y3.iloc[i])
  print(rf2.predict(test_x3)[i])
```

```
test set: 0
8606.2174
23534.4882971737
test set: 1
12485.8009
47184.83184563781
test set: 2
2457.502
17754.44256107169
test set: 3
11737.84884
20065.31115773501
test set: 4
2219.4451
37329.264324999145
```

- Intuitively, the model works better for smoker
- On the other hand, the error for non-smoker is more obvious
- May have over fitting or under fitting problem

Result Visualization, Actual VS. Predicted



RandomForestRegressor - Predict Smoker/Non-Smoker with 5 Fold Cross Validation

```
#smoker
df2 = df1.loc[df1['smoker'].isin([1])]

features = ["age","sex","bmi","children","smoker"]
x2 = df2[features]
y2 = df2["charges"]
train_x2,test_x2,train_y2,test_y2 = train_test_split(x2,y2)
df2.head()

rf2 = RandomForestRegressor(n_estimators=100, max_leaf_nodes=10, min_samples_split=35, max_features=2)
rf2.fit(train_x2, train_y2)
result = cross_val_score(rf2 , x, y, cv = kf)
print(np.mean(result))

predictions = rf2.predict(test_x2)

print("Mean absolute error: %.2f" % np.mean(np.absolute(predictions - test_y2.values)))
print("Residual sum of squares (MSE): %.2f" % np.mean((predictions - test_y2.values) ** 2))
print("R2-score: %.2f" % r2_score(test_y2.values, predictions))
```

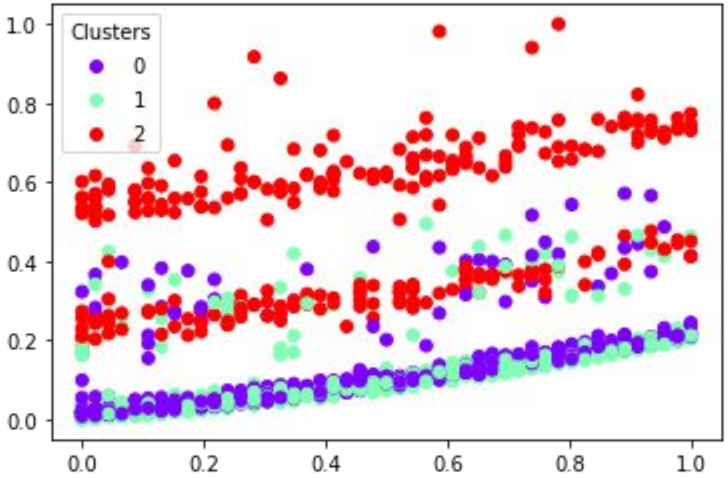
```
0.8181977541807042
Mean absolute error: 3575.23
Residual sum of squares (MSE): 20536841.45
R2-score: 0.82
```

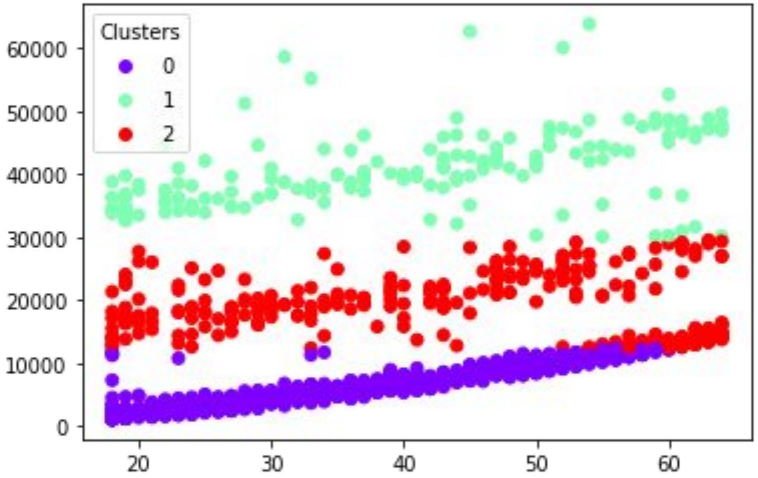
```
#not smoker
df3 = df1.loc[df1['smoker'].isin([0])]

features = ["age","sex","bmi","children","smoker"]
x3 = df3[features]
y3 = df3["charges"]
train_x3,test_x3,train_y3,test_y3 = train_test_split(x3,y3)
df3.head()

rf3 = RandomForestRegressor(n_estimators=100, max_leaf_nodes=10, min_samples_split=35, max_features=2)
rf3.fit(train_x3, train_y3)
result = cross_val_score(rf3 , x, y, cv = kf)
print(np.mean(result))

predictions = rf3.predict(test_x3)

print("Mean absolute error: %.2f" % np.mean(np.absolute(predictions - test_y3.values)))
print("Residual sum of squares (MSE): %.2f" % np.mean((predictions - test_y3.values) ** 2))
print("R2-score: %.2f" % r2_score(test_y3.values, predictions))
```

```
0.8154759496338368
Mean absolute error: 2437.28
Residual sum of squares (MSE): 20028484.13
R2-score: 0.39
```

Though there is large difference in R2-score, 5 cv fold shows an average level of performance

K-Means Clustering - 300_iter

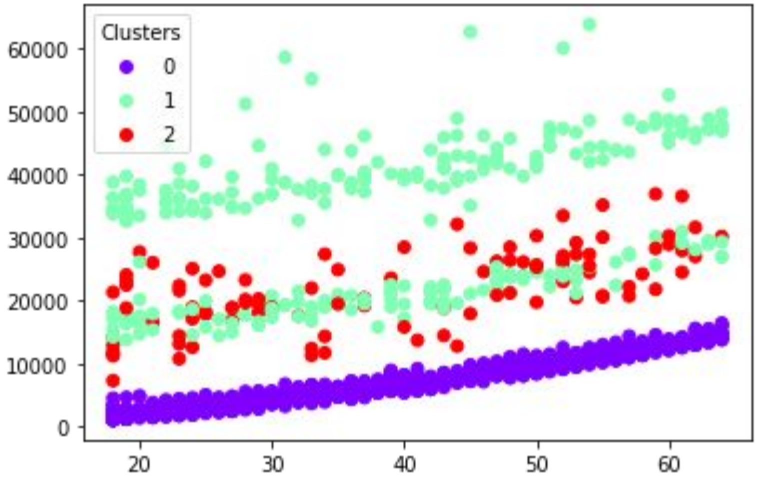




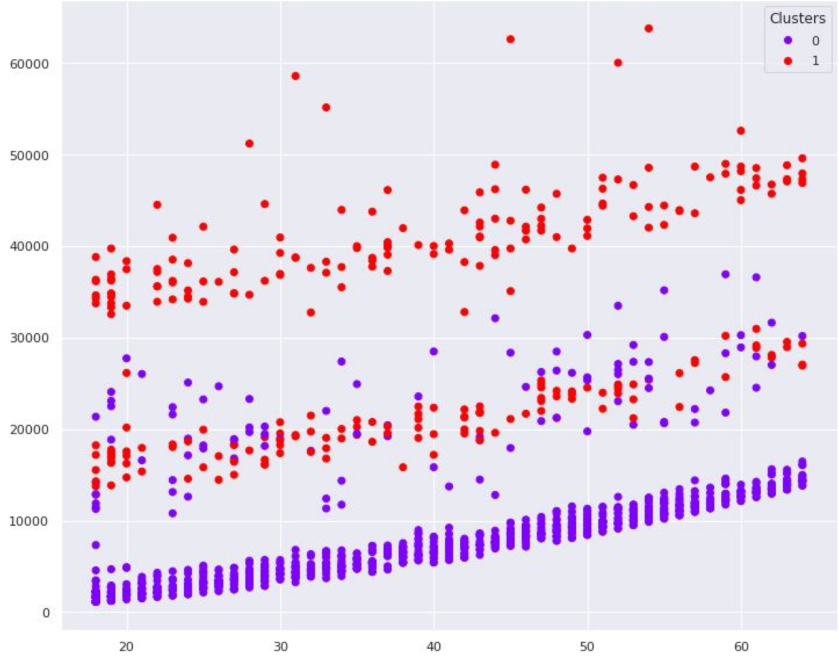
With
Normalization(0,1)

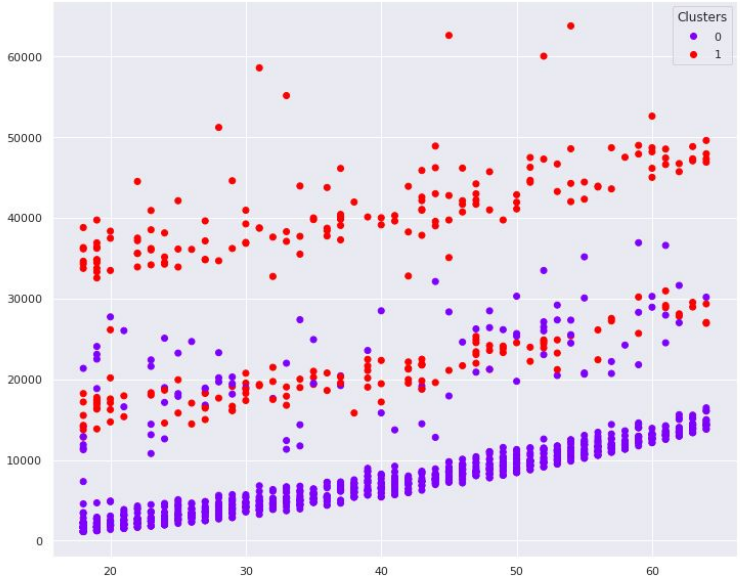
```
0: [36, 6296]
1: [40, 40761]
2: [45, 18505]
```

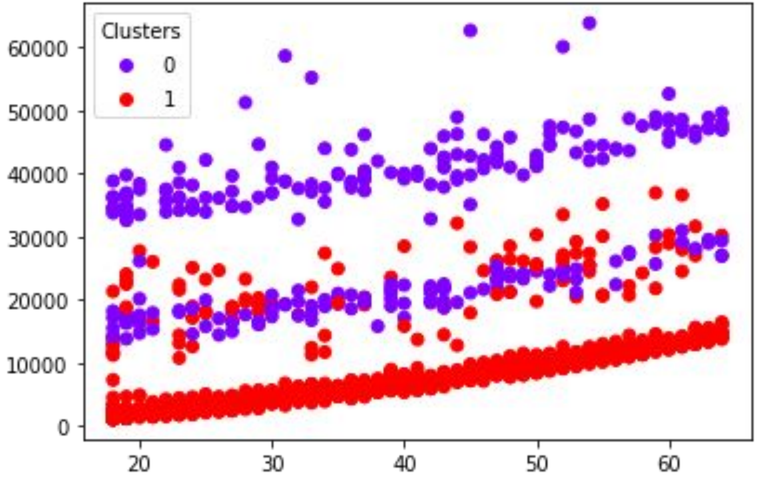
GMM Clustering - 300_iter



0: [36, 6296]
1: [40,
40761]
2: [45,
18505]

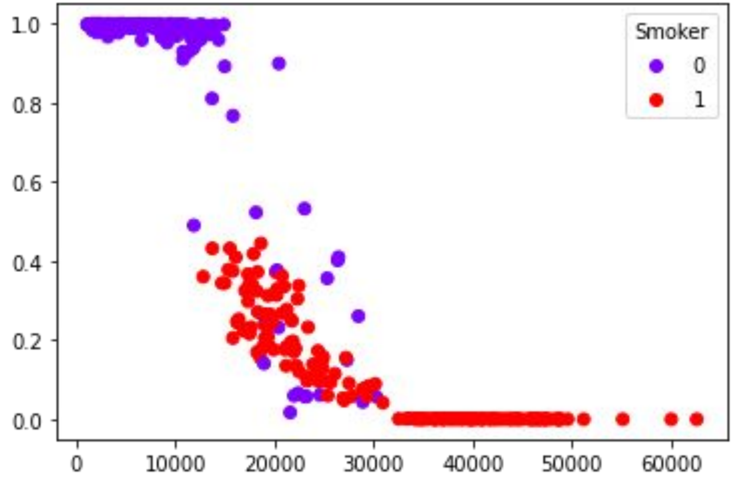


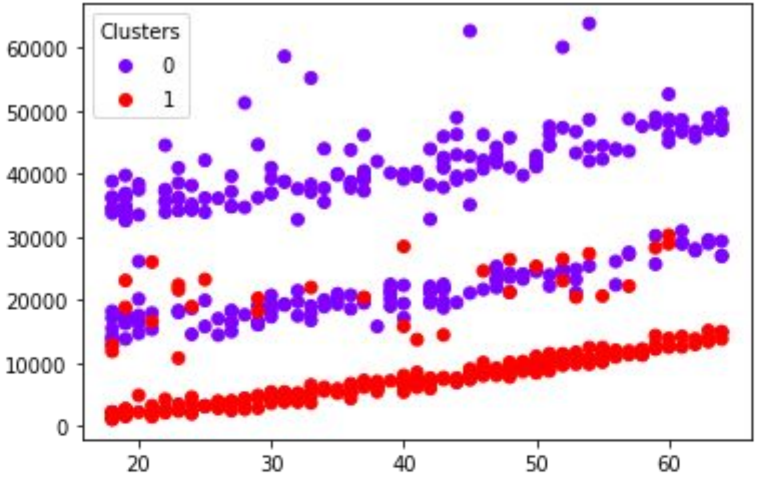




1: [3.9 7047]
0: [3.8 29368]]

Logistic Regression & GMM Model-Predict Smoker/Non-Smoker






```
[-0.0655244 ,  0.0863435 , -0.19792367,
-0.08792067,  0.36826806, 0.00040367]
```

```
The Training Accuracy is:  0.954
The Testing Accuracy is:  0.954
```

```
The Training Accuracy is:  0.771
The Testing Accuracy is:  0.704
```

Results

	R^2 Score	Mean Absolute Error	Mean Squared Error	Features
Linear Regression	0.04, 0.11	12626.98, 12021.34	211253607.93, 194873962.16	Age, BMI, Smoker
Polynomial Regression	0.98	412.32	566326.94	Age, BMI, Smoker
Random Forest Regression	0.8166825	3475.61	25532514.61	All features

Conclusion

- **Linear Regression Model** $R^2 = 0.98$
  - Efficient but
  - underfitting
- **Random Forest Model** $R^2 = 0.81$,
  - lower performance
  - less likely to suffer from fitting issues

- **Logistic Regression & Clustering**

  A **Significant Positive Correlation between <u>Smoking</u> and <u>High Medical Cost</u>** was Observed