Leitfaden für nachvollziehbare Schritte

1. Kurze Darstellung des Problembereichs / Aufriss des Themas

1.1 Inhaltlich

Kern der Untersuchung : Data Analysis and Biulding a Neural Network for predicting the medical cost.

Grobziele der Arbeit : For any healthe insurance company to make money ,it needs to collect more money than it spends on the medical care on its customers. As a result it needs to develop a model which accurately predicts the medical expenses for the customers.Also its difficult to estimate the expenses for certain segment of polpulation . For instance smokers are more likely to have lung cancer and the more obese people are likely to have heart deseases.Hence the goal of this analysis is to use the data and estimate the average medical expenses for such customers. And can be used to set the cost depending upon the expected health issues.

1.2 Begründung desThemas

Darstellung der Relevanz des Themas?

Warum ist das Thema wichtig und interessant und daher bearbeitungs- und förderungswürdig?

Health insurance makes a difference in whether and when people get necessary medical care, where they get their care, and ultimately, how healthy they are. Uninsured people are far more likely than those with insurance to postpone health care or forgo it altogether. The consequences can be severe, particularly when preventable conditions or chronic diseases go undetected.

Darstellung eines persönlichen Erkenntnisinteresses.

Dieser Abschnitt soll ein prägnanter Einstieg in die Projektarbeit / Seminararbeit sein.

Er soll beim Leser Interesse für das Thema und die Bereitschaft wecken oder verstärken, die Arbeit zu betreuen bzw. zu fördern und dient der Eigenmotivation.

To understand the factors that effect the charges of medical expenses can be learnt with the analysis and Moreover, without a proper understanding of the data, it is possible during the analysis and data interpretation to mistakenly interpret the correlation between variables as a causal relationship.

2. Nachvollziehbare Schritte

2.1 Der Stand der Forschung / Auswertung der vorhandenen Literatur / Tutorials ...

Welche Aspekte wurden untersucht und welche nicht?

In the given data set there are columns which does not correlate to the medical expenses

Incured by the customer .It would have been good to collect the data set which directly effect the charges .

Welche Kontroversen gab es und welche Methoden standen bis jetzt im Vordergrund?

Lösungswege strukturieren!

Importing important libraries

Load the dataset into a data frame using Pandas

Explore the number of rows & columns, ranges of values etc.

The data had no missing values hence there was no need for replacing any values

Plotting the graph using seaborn and plotly to study the relation between the data

Wichtigste (verwendete) wissenschaftliche Positionen zum ausgewählten Thema?

(Z.B. **Tutorials …**)

2.2 Fragestellung

What are the factors on which the medical charges depend?

2.4 Wissenslücke

There could be some coloums in the data set which really effect the charges like the occupation of the customer ,the health status and affordabity of the customer

2.5 Methode

Detaillierte nachvollziehbare Beschreibung der Vorgehensweise !!

Vgl. MUSTER-PROJEKTE in den Tutorials !!

- Pandas for reading and analyzing the data
- Seaborn and plotly for ploting the graphs
- Scikit-Learn for data preprocessing (encoding, scaling, train/test split)
- Tensorflow & Keras API to create the models

Importing the required libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from plotly.offline import plot
#%%
from sklearn import preprocessing
from sklearn.preprocessing import MinMaxScaler,LabelEncoder, OneHotEncoder
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
#%%
from sklearn.preprocessing import MinMaxScaler
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dense
from tensorflow.python.keras.wrappers.scikit_learn  import KerasRegressor
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.metrics import RootMeanSquaredError
```

As the data contains three categoricall column so imported Label Encoder and OnehHotEncoder from scikitlearn preprocessor . Also as we are predicting the charges of the medical insurance ,hence creating the regression model .

Reading the data using pandas and exploring the columns

```
#%%
# Reading the data and exploring the columns
df=pd.read_csv(r"C:\Users\Vaishu\Desktop\Work\Neural_Network\US_Insurance.csv")
print(df.head())
df.info()
df.describe()
df.age.describe()#distribution of age
df.columns
```

RangeIndex: 1338 entries, 0 to 1337

Data columns (total 7 columns):

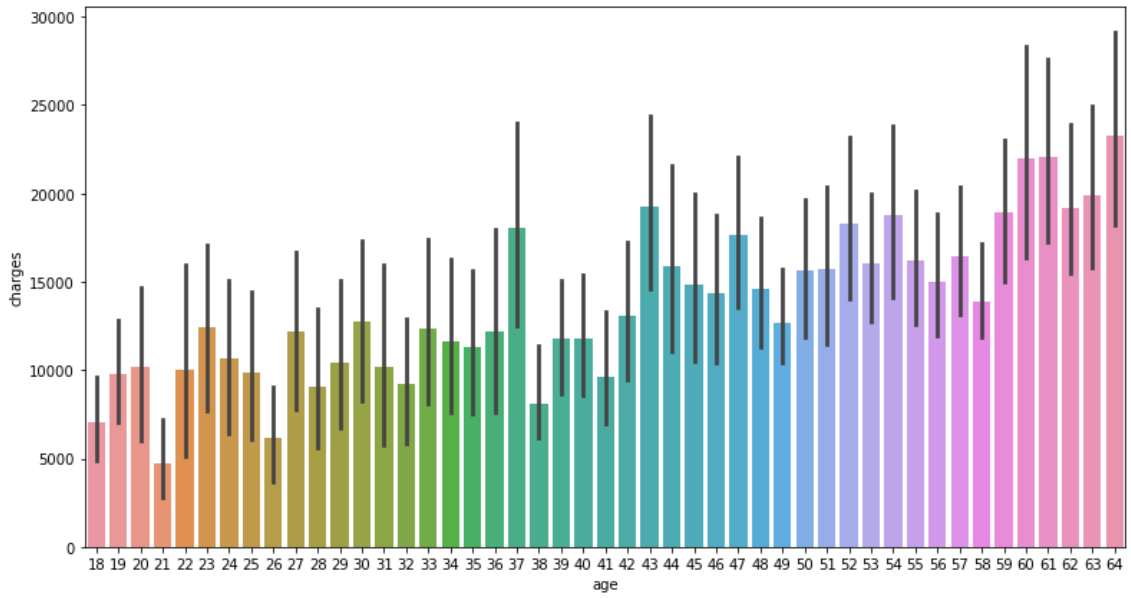
#	Column	Non-Null Count	Dtype
0	age	1338 non-null	int64
1	gender	1338 non-null	object
2	bmi	1338 non-null	float64
3	children	1338 non-null	int64
4	smoker	1338 non-null	object
5	region	1338 non-null	object
6	charges	1338 non-null	float64

dtypes: float64(2), int64(2), object(3)

memory usage: 73.3+ KB

Exploring the relation between the charges and age

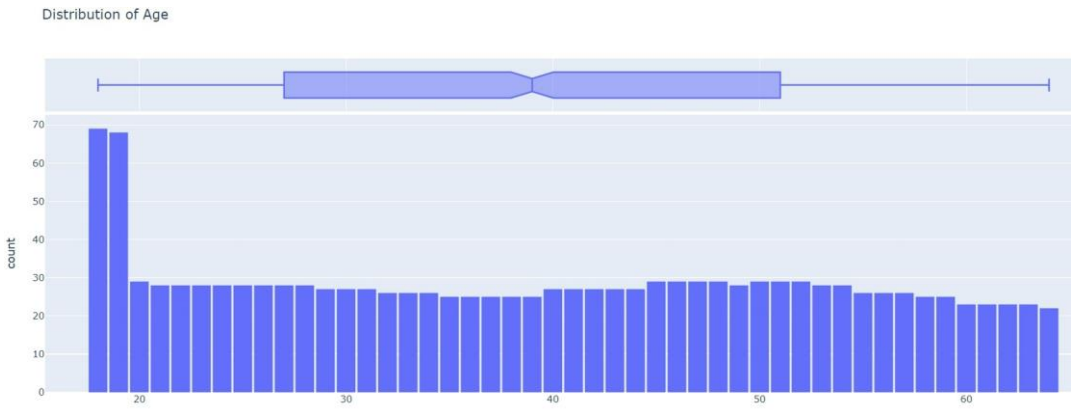
```
#%% Comparing charges with age


#sns.histplot(data=df, x="age",  binwidth=1)
plt.figure(figsize=(13,7))
sns.barplot( x = 'age', y = 'charges', data = df)
plt.show()
'''
There is a clear indication that with increase in  age the charges  increase

'''
```



There is a clear evidence that with increase in age the charges increase. As the age increase the medical expenses also increase due health issues

Comparision of the number of customers with age

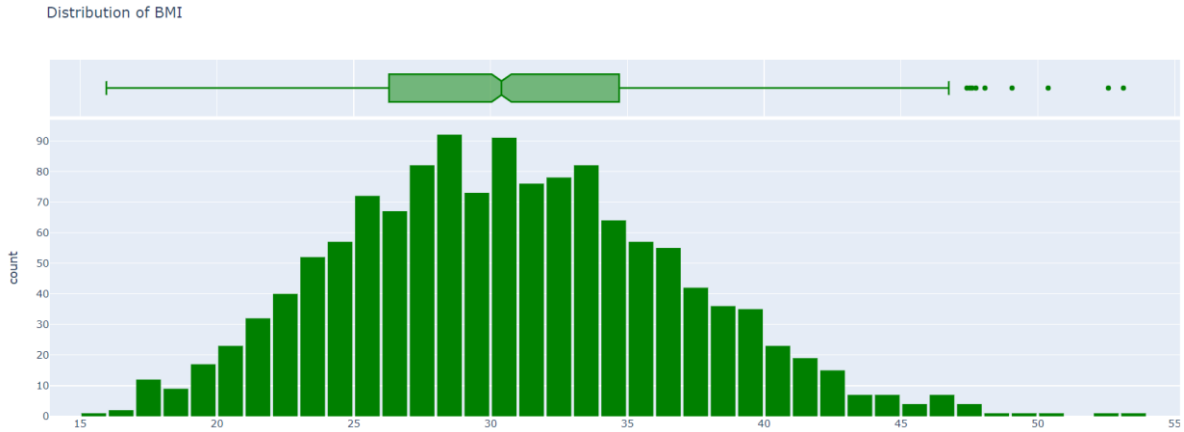
```
#%%Comparision of number of customers with age
fig = px.histogram(df, x='age', marginal='box', nbins=47, title='Distribution of Age')
fig.update_layout(bargap=0.1)
plot(fig)
```



The distribution of ages in the dataset is almost uniform, with 20-30 customers at every age, but for the ages 18 and 19 there are twice number of customers
Why there is increase in the number of customers in the age group 18 to 19??

Comparision of the number of customer w.r.to BMI

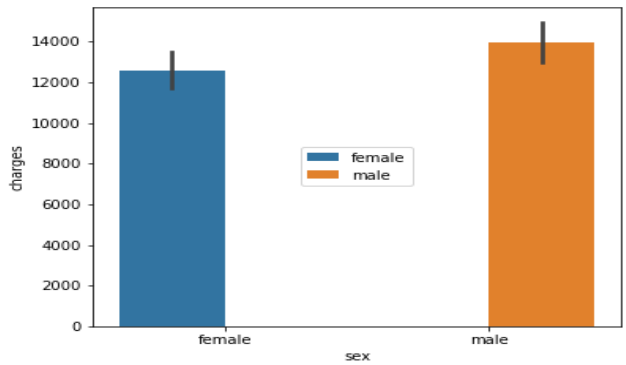
```
#%% Comparision of number of customers with bmi
fig = px.histogram(df, x='bmi', marginal='box', nbins=47,color_discrete_sequence=['green'] ,ti
fig.update_layout(bargap=0.1)
plot(fig)
'''
```



The distribution of BMIs forms a gaussian distribution unlike the distribution of age
It means most of the customers are having normal weight and BMI or slightly overweight (i.e., range from 18 to 30) but there are few outliers as well.

Comparing the charges according to Gender

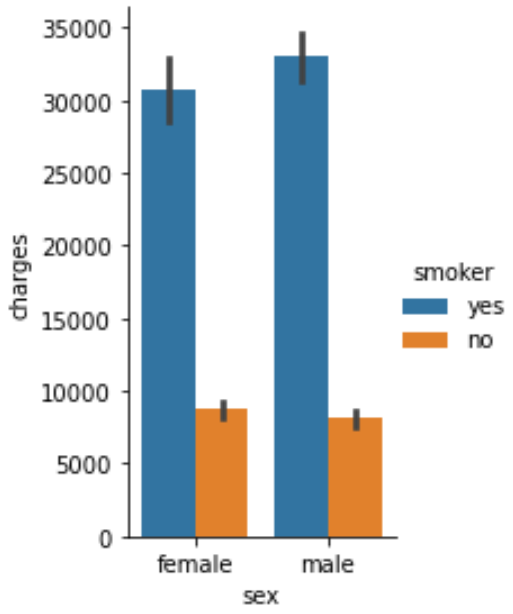
```
#%% Comparing charges according to gender
ax = sns.barplot(x="sex", y="charges", hue="sex", data=df)
plt.legend(loc="center")
'''
```



The graph clearly indicates that the gender of the customer affects slightly in the charges

Comparing the charges for Smokers and Non-Smokers

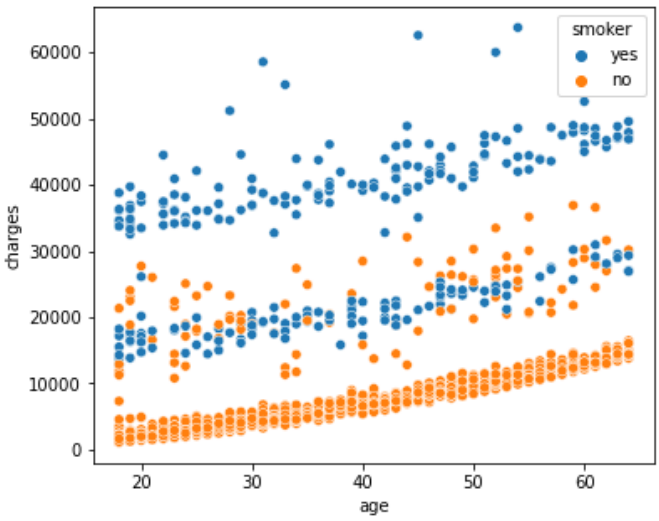
```
#%% Comparision of charges with respect to smoker and non smoker with gender
g = sns.catplot(x="sex", y="charges", hue="smoker", data=df, kind="bar",height=4, aspect=.7)
'''
```



There is a significant difference in medical expenses between smokers and non-smokers. Though the female smokers expenses is less compared to male smokers.
This inturn indicates the strong correlation between the charges incured by smokers and non smoker.Note that the charges for most customers are below 10,000$

Visualization of relationship charges and age also used different colour smoker

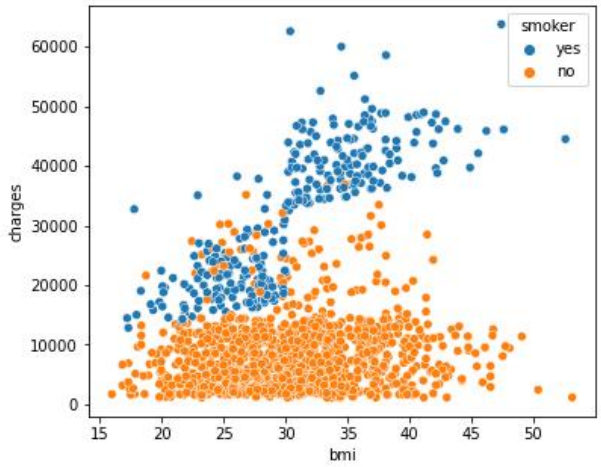
```
#%%
sns.scatterplot(data=df, x="age", y="charges", hue="smoker")
```



There is significant variation at every age, and it's clear that age alone cannot be used to accurately determine medical charges.
We also observe three clusters the first one shows non-smokers who have lower medical charges.The second shows mix of both smokers and non smokers which have a bit of high medical charges The third one show completly smokers who have higher medical charges . So the assumption would be that people who are non smokers and have less health isssues have lesser charges than the smokers and non smoker with healthe issues.

Visualization of charges in accordance with BMI

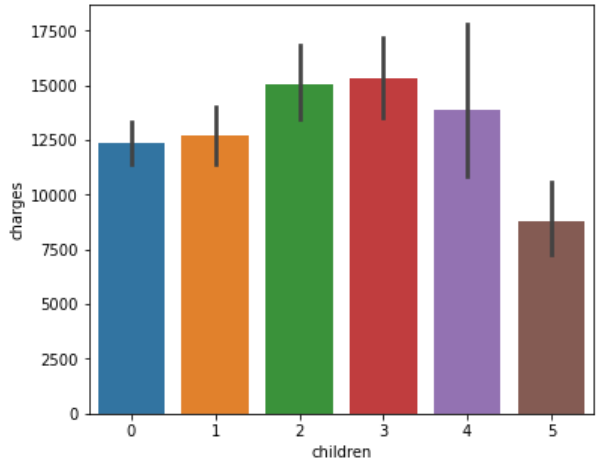
```
98  #%% viaualization of  charges and BMI
99  sns.scatterplot(data=df, x="bmi", y="charges",hue="smoker" )
```

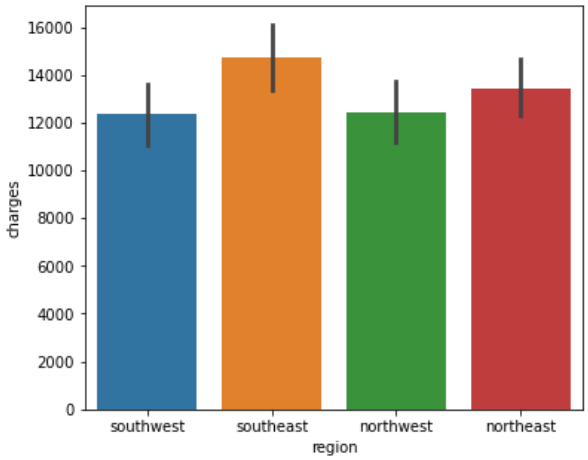


It seems that the smokers with lower BMI have less medical expenses compared to smokers with higher BMI

Comparing the charges w.r.t Number of Children and Region

```
4   #%%Comparing the charges with number of children
5   ax = sns.barplot(x="children", y="charges", data=df)
106 #%%
107 #Comparing the charges with region
108 ax = sns.barplot(x="region", y="charges", data=df)
```

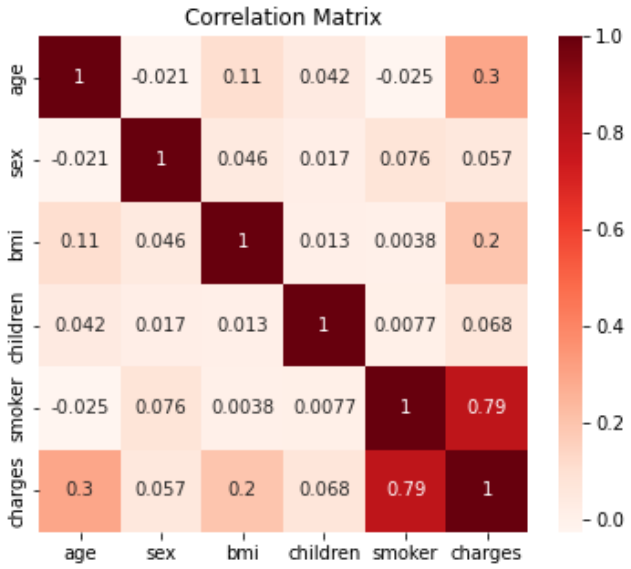




There isn't much difference in the charges with respect to number of children unless the number of childeren is greater than 4
Also region does not have much difference on the charges
The southeast region have slightly higher charges compared to other regions

Correlation Matrix

```
#%%
#Correlation Matrix
df.corr()
sns.heatmap(df.corr(), cmap='Reds', annot=True)
plt.title('Correlation Matrix');
```



The correlation Martix clear show the strong correlation between age and smoker

With the above Anslysis we can say that the values in some columns are more closely related to charges compared to others .Lets move on to creating a Regression model. For this we need to convert all columns to numerical

Converting Categorical column to numeric using Label encoder

```
#%% Converting categorical column 'sex' and 'smoker' to numerical using Label Encoder

cols = ['sex', 'smoker']
df[cols] = df[cols].apply(LabelEncoder().fit_transform)
df.head()
df[cols]
df.columns
```

The Gender and Smoker column had two categories hence used Label Encoder from scikitlearn . It converts categorical text data into model-understandable numerical data.

Converting Region categorical column to numeric using One Hot Encoder

```
#%% converting 'region' categorical column to numeric using One Hot Encoder

enc = preprocessing.OneHotEncoder()
enc.fit(df[['region']])
one_hot = enc.transform(df[['region']]).toarray()
one_hot
df[['northeast', 'northwest', 'southeast', 'southwest']] = one_hot
df.head()
df.shape
df.columns
num_df=df.drop(labels=['region'],axis=1)
num_df.columns

```

For categorical variables where no ordinal relationship exists, the integer encoding may not be enough, at best, or misleading to the model at worst. In this case, a one-hot encoding can be applied which converts this into binary matrix for each data instance .

Spliting the data into training and testing

```
#%%
#Splitting the data into train and  test then scaling the train data.

inputs = ['age', 'bmi', 'children', 'smoker', 'sex', 'northeast', 'northwest', 'southeast', 's
target =['charges']

X,y=num_df[inputs],num_df[target]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20,random_state=1)

scaler = MinMaxScaler()

X_train_scaled = scaler.fit_transform(X_train)
print(X_train_scaled)
X_test_scaled=scaler.transform(X_test)

```

The entire data set is now divided into training (80%) and test set (20%) .TheTraining set is used directly for learning. Finally, the performance is determined with the test set of the model on completely new data. The data must be normalized in order to be able to be trained with a neural network. For that will uses minmax scaler from scikit-learn. This is only fitted to the training data and then transformed to test data.

Creating Keras Sequential Model

```
#%%
#creating keras squential model
model = Sequential()
model.add(Dense(16, input_dim=9,  activation='relu'))

model.add(Dense(16, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(1, activation='linear'))
model.summary()

#Compliling the model for learning process
model.compile(loss='mse', optimizer='adam', metrics=['mse','mae'])
callback=EarlyStopping(monitor="mse",patience=10)
history = model.fit(X_train_scaled, y_train, epochs=1000,  batch_size=100,callbacks=[callback]

```

Now its time to creat a Model. A function is defined that creates a neural network as a model.
Its a simple Sequential Model from keras .Its has two hidden layers with one input and out layer respectively. The input layer has 16 nodes and has 9 input attributes and uses relu as an activation function. The network uses good practices such as the rectifier activation function for the hidden layers too with 16 and 32 nodes respectively. The output layer has one node with no activation function hence the argument as linear as we are dealing with regression problem. The efficient ADAM optimization algorithm is used as it achieves good convergence and a mean squared error loss function is optimized. A metric mean sqaure error and mean absolute error is used in to judge the performance of the model. Too many epochs can lead to overfitting of the training dataset, hence early stopping is used. Early stopping is a method that allows to specify an arbitrarily large number of training epochs and stop training once the model performance and stops improving on the test dataset. To discover the training epoch on which training was stopped, the “verbose” argument can be set to 1. The first sign of no improvement may not be the best time to stop training. This is because the model may get slightly worse before getting much better. We can account for this by adding a delay to the trigger in terms of the number of epochs on which we would like to see no improvement. This is done by setting the “patience” argument.Also monitor is set on the loss .Lastly the batch size is set as 100 .

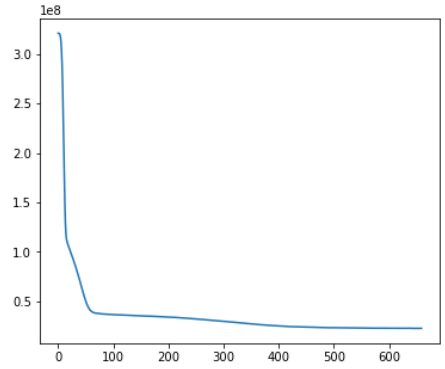
```
Model: "sequential_16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_64 (Dense)             (None, 16)                160
_________________________________________________________________
dense_65 (Dense)             (None, 16)                272
_________________________________________________________________
dense_66 (Dense)             (None, 32)                544
_________________________________________________________________
dense_67 (Dense)             (None, 1)                 33
=================================================================
Total params: 1,009
Trainable params: 1,009
Non-trainable params: 0
```

```
Epoch 656/2000
34/34 [==============================] - 0s 664us/step - loss: 22874710.0000 - mse: 22874710.0000 -
mae: 2859.1831
Epoch 657/2000
34/34 [==============================] - 0s 695us/step - loss: 22920408.0000 - mse: 22920408.0000 -
mae: 2811.6064
Epoch 658/2000
34/34 [==============================] - 0s 702us/step - loss: 22900258.0000 - mse: 22900258.0000 -
mae: 2836.5945
Epoch 659/2000
34/34 [==============================] - 0s 704us/step - loss: 22904338.0000 - mse: 22904338.0000 -
mae: 2881.1873
Epoch 660/2000
34/34 [==============================] - 0s 664us/step - loss: 22871824.0000 - mse: 22871824.0000 -
mae: 2873.4854
```

We see above here the it stopped at 660 epochs

Ploting the loss

```
184  #%%Plotting the loss
185  plt.plot(history.history["mse"])
186  plt.show()
```




```
87  #%%
88  #Calculating the Error
89
90  y_pred=model.predict(X_test_scaled)
91  y_pred
92  error=np.sqrt((y_pred-y_test)**2)
93  error.mean()
```

```
In [117]: error.mean()
Out[117]:
charges    2711.680906
dtype: float64
```

The error we got with this model is 2711$

2.7 Ausblick

From the regression analysis, we find that region and gender do not bring significant difference on charges. Age, BMI, number of children and smoking are the ones that drive the charges. Smoking seems to have the most influence on the medical charges.