Problem statement:To predict how best the data fits

Data collection

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing,svm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [2]:

```
df=pd.read_csv(r"C:\Users\Gowthami\Downloads\insurance.csv")
df
```

Out[2]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

Data cleaning and preprocessing

Exploratory data analysis

In [3]:

```
df.head()
```

Out[3]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

In [4]:

```
df.tail()
```

Out[4]:

	age	sex	bmi	children	smoker	region	charges
1333	50	male	30.97	3	no	northwest	10600.5483
1334	18	female	31.92	0	no	northeast	2205.9808
1335	18	female	36.85	0	no	southeast	1629.8335
1336	21	female	25.80	0	no	southwest	2007.9450
1337	61	female	29.07	0	yes	northwest	29141.3603

In [5]:

```
df.shape
```

Out[5]:

```
(1338, 7)
```

In [6]:

```
df.describe
```

Out[6]:

```
<bound method NDFrame.describe of       age     sex     bmi  children smoker     region      charges
0      19  female  27.900         0    yes  southwest  16884.92400
1      18    male  33.770         1     no  southeast   1725.55230
2      28    male  33.000         3     no  southeast   4449.46200
3      33    male  22.705         0     no  northwest  21984.47061
4      32    male  28.880         0     no  northwest   3866.85520
...   ...     ...     ...       ...    ...        ...          ...
1333   50    male  30.970         3     no  northwest  10600.54830
1334   18  female  31.920         0     no  northeast   2205.98080
1335   18  female  36.850         0     no  southeast   1629.83350
1336   21  female  25.800         0     no  southwest   2007.94500
1337   61  female  29.070         0    yes  northwest  29141.36030

[1338 rows x 7 columns]>
```

In [7]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1338 non-null   int64  
 1   sex       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
```

In [8]:

```
df.isnull().any()
```

Out[8]:

```
age         False
sex         False
bmi         False
children    False
smoker      False
region      False
charges     False
dtype: bool
```

In [9]:

```
df.isna().sum()
```

Out[9]:

```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

In [10]:

```
df['region'].value_counts()
```

Out[10]:

```
region
southeast    364
southwest    325
northwest    325
northeast    324
Name: count, dtype: int64
```

In [11]:

```
convert={"sex":{"female":1,"male":0}}
df=df.replace(convert)
df
```

Out[11]:

	age	sex	bmi	children	smoker	region	charges
0	19	1	27.900	0	yes	southwest	16884.92400
1	18	0	33.770	1	no	southeast	1725.55230
2	28	0	33.000	3	no	southeast	4449.46200
3	33	0	22.705	0	no	northwest	21984.47061
4	32	0	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	0	30.970	3	no	northwest	10600.54830
1334	18	1	31.920	0	no	northeast	2205.98080
1335	18	1	36.850	0	no	southeast	1629.83350
1336	21	1	25.800	0	no	southwest	2007.94500
1337	61	1	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

In [12]:

```
convert={"smoker":{"yes":1,"no":0}}
df=df.replace(convert)
df
```

Out[12]:

	age	sex	bmi	children	smoker	region	charges
0	19	1	27.900	0	1	southwest	16884.92400
1	18	0	33.770	1	0	southeast	1725.55230
2	28	0	33.000	3	0	southeast	4449.46200
3	33	0	22.705	0	0	northwest	21984.47061
4	32	0	28.880	0	0	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	0	30.970	3	0	northwest	10600.54830
1334	18	1	31.920	0	0	northeast	2205.98080
1335	18	1	36.850	0	0	southeast	1629.83350
1336	21	1	25.800	0	0	southwest	2007.94500
1337	61	1	29.070	0	1	northwest	29141.36030

1338 rows × 7 columns

In [13]:

```
convert={"region":{"southwest":1,"southeast":2,"northeast":3,"northwest":4}}
df=df.replace(convert)
df
```

Out[13]:

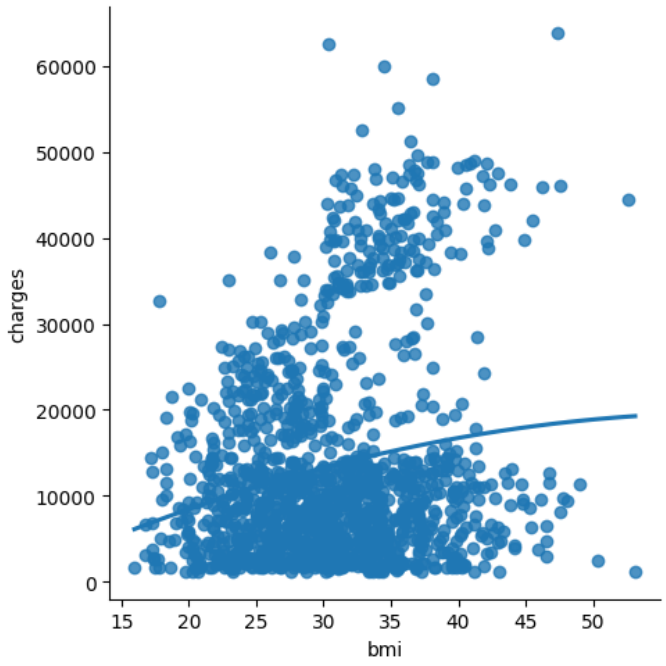
	age	sex	bmi	children	smoker	region	charges
0	19	1	27.900	0	1	1	16884.92400
1	18	0	33.770	1	0	2	1725.55230
2	28	0	33.000	3	0	2	4449.46200
3	33	0	22.705	0	0	4	21984.47061
4	32	0	28.880	0	0	4	3866.85520
...	...	...	...	...	...	...	...
1333	50	0	30.970	3	0	4	10600.54830
1334	18	1	31.920	0	0	3	2205.98080
1335	18	1	36.850	0	0	2	1629.83350
1336	21	1	25.800	0	0	1	2007.94500
1337	61	1	29.070	0	1	4	29141.36030

1338 rows × 7 columns

Data visualization

In [14]:

```
sns.lmplot(x='bmi',y='charges',order=2,data=df,ci=None)
plt.show()
```



In [15]:

```
x=np.array(df['bmi']).reshape(-1,1)
y=x=np.array(df['charges']).reshape(-1,1)
```

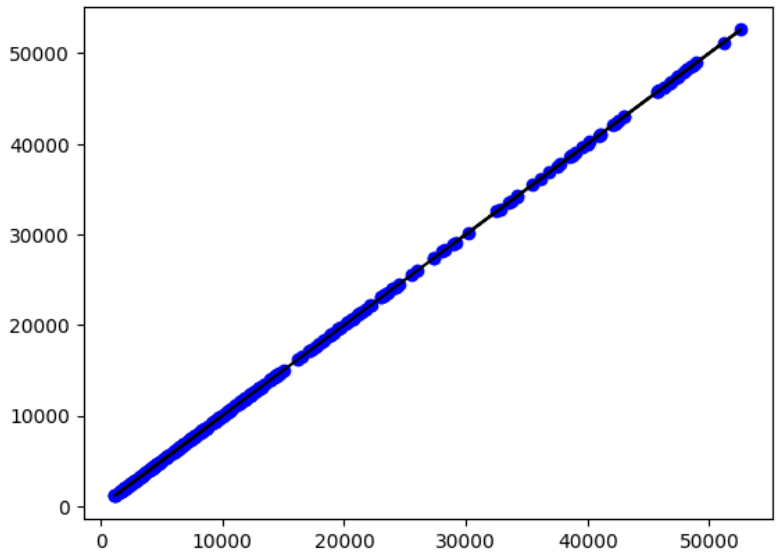
In [16]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=0)
lr=LinearRegression()
lr.fit(x_train,y_train)
print(lr.score(x_test,y_test))
```

```
1.0
```

In [17]:

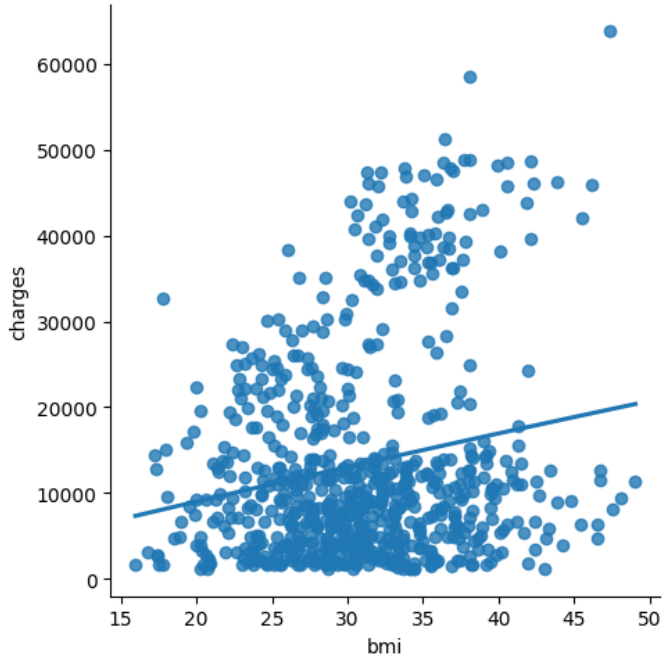
```
y_pred=lr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```



working with subset of data

In [18]:

```
df700=df[:][:700]
sns.lmplot(x='bmi',y='charges',order=2,ci=None,data=df700)
plt.show()
```



In [19]:

```
df700.fillna(method='ffill',inplace=True)
```

In [20]:

```
x=np.array(df700["bmi"]).reshape(-1,1)
y=np.array(df700['charges']).reshape(-1,1)
```

In [21]:

```
df700.dropna(inplace=True)
```

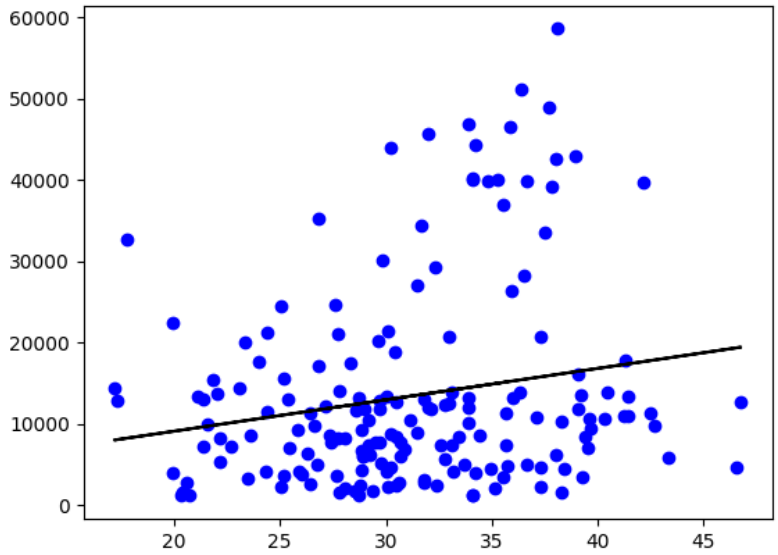
In [22]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
lr=LinearRegression()
lr.fit(x_train,y_train)
print(lr.score(x_test,y_test))
```

```
0.040628081724721654
```

In [23]:

```
y_pred=lr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```



Evaluation of model

In [24]:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
```

In [25]:

```
lr=LinearRegression()
lr.fit(x_train,y_train)
y_pred=lr.predict(x_test)
r2=r2_score(y_test,y_pred)
print(r2)
```

```
0.040628081724721654
```

Ridge Regression

In [26]:

```
from sklearn.linear_model import Lasso,Ridge
from sklearn.preprocessing import StandardScaler
```

In [27]:

```
plt.figure(figsize=(10,10))
sns.heatmap(df700.corr(),annot=True)
plt.show()
```

In [28]:

```
features=df.columns[0:1]
target=df.columns[-1]
```

In [29]:

```
x=df[features].values
y=df[target].values
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=1)
print("The dimension of X_train is {}".format(x_train.shape))
print("The dimension of X_test is {}".format(x_test.shape))
```

```
The dimension of X_train is (936, 1)
The dimension of X_test is (402, 1)
```

In [30]:

```
lr = LinearRegression()
#Fit model
lr.fit(x_train, y_train)
#predict
actual = y_test
train_score_lr = lr.score(x_train, y_train)
test_score_lr = lr.score(x_test, y_test)
print("\nLinear Regression Model:\n")
print("The train score for lr model is {}".format(train_score_lr))
print("The test score for lr model is {}".format(test_score_lr))
```

```
Linear Regression Model:

The train score for lr model is 0.0910963973805714
The test score for lr model is 0.08490473916580776
```

In [31]:

```
ridgeReg = Ridge(alpha=10)
ridgeReg.fit(x_train,y_train)
#train and test scorefor ridge regression
train_score_ridge = ridgeReg.score(x_train, y_train)
test_score_ridge = ridgeReg.score(x_test, y_test)
print("\nRidge Model:\n")
print("The train score for ridge model is {}".format(train_score_ridge))
print("The test score for ridge model is {}".format(train_score_ridge))
```

```
Ridge Model:

The train score for ridge model is 0.09109639711159634
The test score for ridge model is 0.09109639711159634
```

In [32]:

```
plt.figure(figsize=(10,10))
```

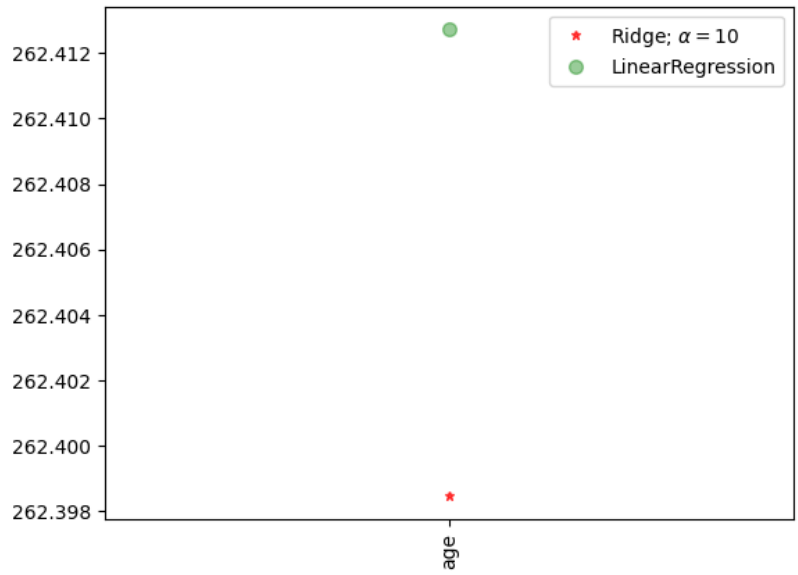
Out[32]:

```
<Figure size 1000x1000 with 0 Axes>
```

```
<Figure size 1000x1000 with 0 Axes>
```

In [33]:

```
lt.plot(features,ridgeReg.coef_,alpha=0.7,linestyle='none',marker="*",markersize=5,color='red',label=r'Ridge; $\alpha=10$',zorder=7)
lt.plot(features,lr.coef_,alpha=0.4,linestyle='none',marker="o",markersize=7,color='green',label='LinearRegression')
lt.xticks(rotation=90)
lt.legend()
lt.show()
```



Lasso Regression

In [34]:

```
lasso= Lasso(alpha=10)
lasso.fit(x_train,y_train)
#train and test scorefor ridge regression
train_score_ls = lasso.score(x_train, y_train)
test_score_ls= lasso.score(x_test, y_test)
print("\nRidge Model:\n")
print("The train score for lasso model is {}".format(train_score_ls))
print("The test score for lasso model is {}".format(test_score_ls))
```

```
Ridge Model:

The train score for lasso model is 0.09109639395809044
The test score for lasso model is 0.08490704421828055
```

In [35]:

```
plt.figure(figsize=(10,10))
```

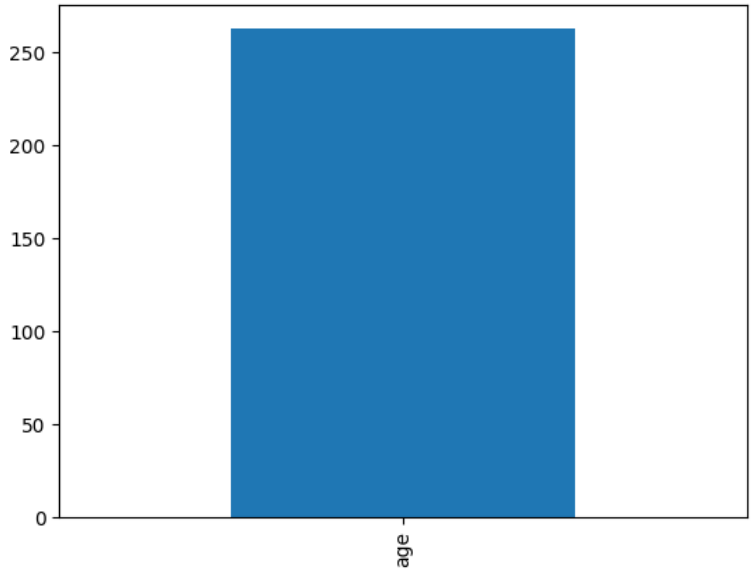
Out[35]:

```
<Figure size 1000x1000 with 0 Axes>
```

```
<Figure size 1000x1000 with 0 Axes>
```

In [36]:

```
pd.Series(lasso.coef_, features).sort_values(ascending = True).plot(kind = "bar")
plt.show()
```



In [37]:

```
from sklearn.linear_model import LassoCV
```

In [38]:

```
#using the linear cv model
from sklearn.linear_model import RidgeCV
#cross validation
ridge_cv=RidgeCV(alphas =[0.0001,0.001,0.01,0.1,1,10]).fit(x_train,y_train)
#score
print(ridge_cv.score(x_train,y_train))
print(ridge_cv.score(x_test,y_test))
```

```
0.09109639711159612
0.08490538609884613
```

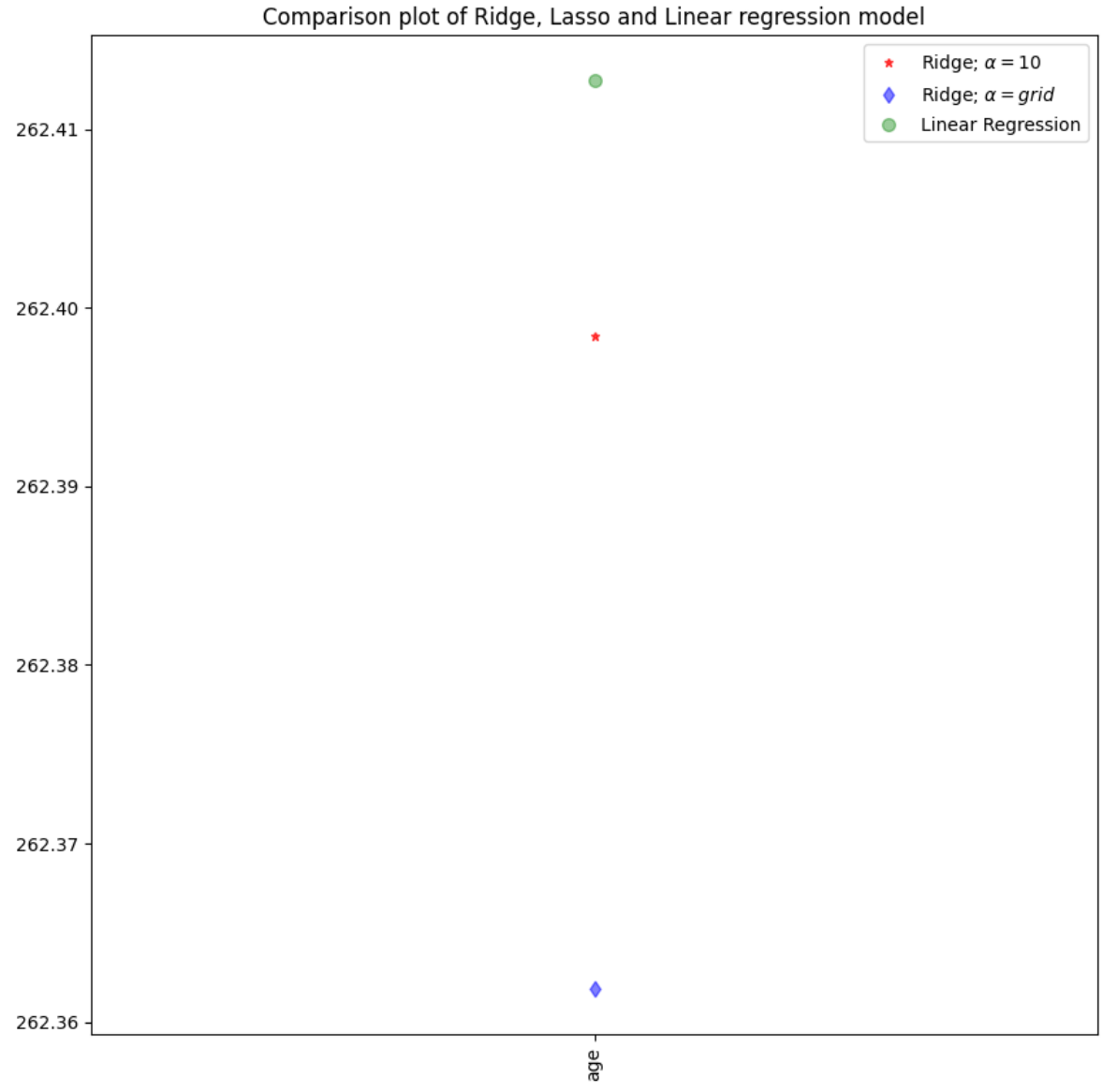
In [39]:

```
#using the linear cv model
from sklearn.linear_model import LassoCV
#cross validation
lasso_cv=LassoCV(alphas =[0.0001,0.001,0.01,0.1,1,10]).fit(x_train,y_train)
#score
print(lasso_cv.score(x_train,y_train))
print(lasso_cv.score(x_test,y_test))
```

```
0.09109639395809044
0.08490704421828055
```

In [40]:

```
lt.figure(figsize = (10, 10))
add plot for ridge regression
lt.plot(features,ridgeReg.coef_,alpha=0.7,linestyle='none',marker='*',markersize=5,color='red',label=r'Ridge; $\alpha=10$',zorder=7)
add plot for lasso regression
lt.plot(lasso_cv.coef_,alpha=0.5,linestyle='none',marker='d',markersize=6,color='blue',label=r'Ridge; $\alpha= grid$')
add plot for linear model
lt.plot(features,lr.coef_,alpha=0.4,linestyle='none',marker='o',markersize=7,color='green',label='Linear Regression')
rotate axis
lt.xticks(rotation = 90)
lt.legend()
lt.title("Comparison plot of Ridge, Lasso and Linear regression model")
lt.show()
```



ElasticNet Regression

In [41]:

```
from sklearn.linear_model import ElasticNet
```

In [42]:

```
el=ElasticNet()
el.fit(x_train,y_train)
print(el.coef_)
print(el.intercept_)
```

```
[261.74450967]
3115.0831774262424
```

In [43]:

```
y_pred_elastic=el.predict(x_train)
```

In [44]:

```
mean_squared_error=np.mean((y_pred_elastic-y_train)**2)
print(mean_squared_error)
```

```
135077142.70714515
```

In [45]:

```
el=ElasticNet()
el.fit(x_train,y_train)
print(el.score(x_train,y_train))
```

```
0.09109580670592365
```

Logistic Regression

In [46]:

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
```

In [47]:

```
df=pd.read_csv(r"C:\Users\Gowthami\Downloads\insurance.csv")
df
```

Out[47]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

In [48]:

```
df.shape
```

Out[48]:

```
(1338, 7)
```

In [49]:

```
pd.set_option('display.max_rows',10000000000)
pd.set_option('display.max_columns',10000000000)
pd.set_option('display.width',95)
```

In [50]:

```
print('This Dataset has %d rows and %d columns'%(df.shape))
```

```
This Dataset has 1338 rows and 7 columns
```

In [51]:

```
df.head()
```

Out[51]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

In [52]:

```
df.tail()
```

Out[52]:

	age	sex	bmi	children	smoker	region	charges
1333	50	male	30.97	3	no	northwest	10600.5483
1334	18	female	31.92	0	no	northeast	2205.9808
1335	18	female	36.85	0	no	southeast	1629.8335
1336	21	female	25.80	0	no	southwest	2007.9450
1337	61	female	29.07	0	yes	northwest	29141.3603

In [53]:

```
df.describe
```

Out[53]:

```
<bound method NDFrame.describe of       age     sex     bmi  children smoker     region       charges
0      19  female  27.900         0    yes  southwest  16884.924000
1      18    male  33.770         1     no  southeast   1725.552300
2      28    male  33.000         3     no  southeast   4449.462000
3      33    male  22.705         0     no  northwest  21984.470610
4      32    male  28.880         0     no  northwest   3866.855200
5      31  female  25.740         0     no  southeast   3756.621600
6      46  female  33.440         1     no  southeast   8240.589600
7      37  female  27.740         3     no  northwest   7281.505600
8      37    male  29.830         2     no  northeast   6406.410700
9      60  female  25.840         0     no  northwest  28923.136920
10     25    male  26.220         0     no  northeast   2721.320800
11     62  female  26.290         0    yes  southeast  27808.725100
12     23    male  34.400         0     no  southwest   1826.843000
13     56  female  39.820         0     no  southeast  11090.717800
14     27    male  42.130         0    yes  southeast  39611.757700
15     19    male  24.600         1     no  southwest   1837.237000
16     52  female  30.780         1     no  northeast  10797.336200
```

In [54]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1338 non-null   int64  
 1   sex       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
```

In [55]:

```
df.isnull().sum()
```

Out[55]:

```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

In [56]:

```
convert={"smoker":{"yes":1,"no":0}}
df=df.replace(convert)
df
```

Out[56]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	1	southwest	16884.924000
1	18	male	33.770	1	0	southeast	1725.552300
2	28	male	33.000	3	0	southeast	4449.462000
3	33	male	22.705	0	0	northwest	21984.470610
4	32	male	28.880	0	0	northwest	3866.855200
5	31	female	25.740	0	0	southeast	3756.621600
6	46	female	33.440	1	0	southeast	8240.589600
7	37	female	27.740	3	0	northwest	7281.505600
8	37	male	29.830	2	0	northeast	6406.410700
9	60	female	25.840	0	0	northwest	28923.136920

In [57]:

```
convert={"sex":{"female":1,"male":0}}
df=df.replace(convert)
df
```

Out[57]:

	age	sex	bmi	children	smoker	region	charges
0	19	1	27.900	0	1	southwest	16884.924000
1	18	0	33.770	1	0	southeast	1725.552300
2	28	0	33.000	3	0	southeast	4449.462000
3	33	0	22.705	0	0	northwest	21984.470610
4	32	0	28.880	0	0	northwest	3866.855200
5	31	1	25.740	0	0	southeast	3756.621600
6	46	1	33.440	1	0	southeast	8240.589600
7	37	1	27.740	3	0	northwest	7281.505600
8	37	0	29.830	2	0	northeast	6406.410700
9	60	1	25.840	0	0	northwest	28923.136920

In [58]:

```
convert={"region":{"southeast":1,"southwest":2,"northeast":3,"northwest":4}}
df=df.replace(convert)
df
```

136	19	0	34.100	0	0	2	1261.442000
137	22	0	25.175	0	0	4	2045.685250
138	54	1	31.900	3	0	1	27322.733860
139	22	1	36.000	0	0	2	2166.732000
140	34	0	22.420	2	0	3	27375.904780
141	26	0	32.490	1	0	3	3490.549100
142	34	0	25.300	2	1	1	18972.495000
143	29	0	29.735	2	0	4	18157.876000
144	30	0	28.690	3	1	4	20745.989100
145	29	1	38.830	3	0	1	5138.256700
146	46	0	30.495	3	1	4	40720.551050
147	51	1	37.730	1	0	1	9877.607700

In [59]:

```
features_matrix=df.iloc[:,0:4]
```

In [60]:

```
target_vector=df.iloc[:,-3]
```

In [61]:

```
print('The Feature Matrix has %d Rows and %d columns(s)'%(features_matrix.shape))
print('The Target Matrix has %d Rows and %d columns(s)'%(np.array(target_vector).reshape(-1,1).shape))
```

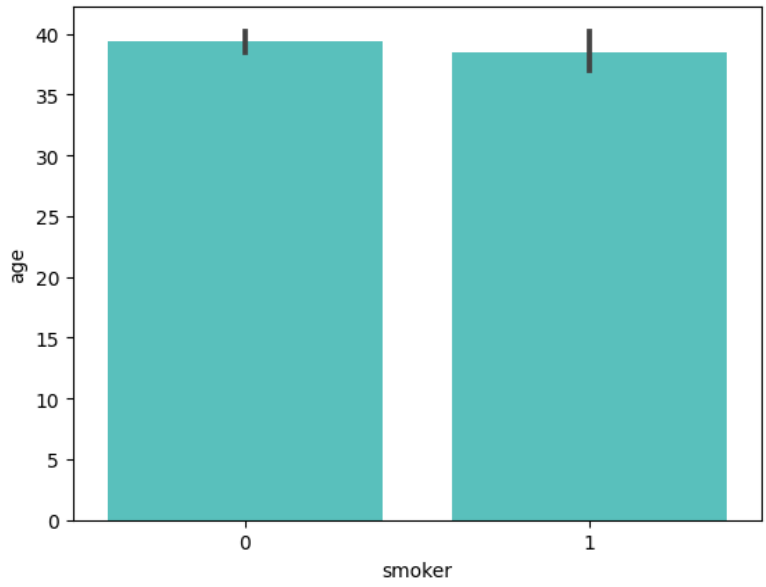
```
The Feature Matrix has 1338 Rows and 4 columns(s)
The Target Matrix has 1338 Rows and 1 columns(s)
```

In [62]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [63]:

```
sns.barplot(x='smoker', y='age', data=df, color="mediumturquoise")
plt.show()
```



In [64]:

```
features_matrix_standardized=StandardScaler().fit_transform(features_matrix)
```

In [65]:

```
algorithm=LogisticRegression(max_iter=10000)
```

In [66]:

```
Logistic_Regression_Model=algorithm.fit(features_matrix_standardized,target_vector)
```

In [67]:

```
observation=[[1,0,0.99539,-0.0588]]
```

In [68]:

```
predictions=Logistic_Regression_Model.predict(observation)
print('The model predicted the observation to belong to class %s'%(predictions))
```

```
The model predicted the observation to belong to class [0]
```

In [69]:

```
print('The algoritham was trained to predict one of the two classes:%s'%(algorithm.classes_))
```

```
The algoritham was trained to predict one of the two classes:[0 1]
```

In [70]:

```
l says the probability of the observation we passed belonging to class[0] Is %s" " "%(algorithm.predict_proba(observation)[0][0]))
l says the probability of the observation we passed belonging to class['1'] Is %s" " "%(algorithm.predict_proba(observation)[0][0]))
```

```
 The Model says the probability of the observation we passed belonging to class[0] Is 0.8057075871331396
 The Model says the probability of the observation we passed belonging to class['1'] Is 0.8057075871331396
```

In [71]:

```
x=np.array(df['age']).reshape(-1,1)
y=np.array(df['smoker']).reshape(-1,1)
```

In [72]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.05)
lo=LogisticRegression()
lo.fit(x_train,y_train)
print(lo.score(x_test,y_test))
```

```
0.7761194029850746

C:\Users\Gowthami\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\utils\validation.py:1143: DataConv
ersionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,
), for example using ravel().
  y = column_or_1d(y, warn=True)
```

Decision Tree

In [73]:

```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
```

In [74]:

```
df=pd.read_csv(r"C:\Users\Gowthami\Downloads\insurance.csv")
df
```

Out[74]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.924000
1	18	male	33.770	1	no	southeast	1725.552300
2	28	male	33.000	3	no	southeast	4449.462000
3	33	male	22.705	0	no	northwest	21984.470610
4	32	male	28.880	0	no	northwest	3866.855200
5	31	female	25.740	0	no	southeast	3756.621600
6	46	female	33.440	1	no	southeast	8240.589600
7	37	female	27.740	3	no	northwest	7281.505600
8	37	male	29.830	2	no	northeast	6406.410700
9	60	female	25.840	0	no	northwest	28923.136920

In [75]:

```
df.shape
```

Out[75]:

```
(1338, 7)
```

In [76]:

```
df.isnull().any()
```

Out[76]:

```
age         False
sex         False
bmi         False
children    False
smoker      False
region      False
charges     False
dtype: bool
```

In [77]:

```
df['region'].value_counts()
```

Out[77]:

```
region
southeast    364
southwest    325
northwest    325
northeast    324
Name: count, dtype: int64
```

In [78]:

```
convert={"sex":{"female":1,"male":0}}
df=df.replace(convert)
df
```

85	45	0	22.895	2	yes	northwest	21098.554050
86	57	1	31.160	0	yes	northwest	43578.939400
87	56	1	27.200	0	no	southwest	11073.176000
88	46	1	27.740	0	no	northwest	8026.666600
89	55	1	26.980	0	no	northwest	11082.577200
90	21	1	39.490	0	no	southeast	2026.974100
91	53	1	24.795	1	no	northwest	10942.132050
92	59	0	29.830	3	yes	northeast	30184.936700
93	35	0	34.770	2	no	northwest	5729.005300
94	64	1	31.300	2	yes	southwest	47291.055000
95	28	1	37.620	1	no	southeast	3766.883800
96	54	1	30.800	3	no	southwest	12105.320000
97	55	0	38.280	0	no	southeast	10226.284200

In [79]:

```
convert={"smoker":{"yes":1,"no":0}}
df=df.replace(convert)
df
```

Out[79]:

	age	sex	bmi	children	smoker	region	charges
0	19	1	27.900	0	1	southwest	16884.924000
1	18	0	33.770	1	0	southeast	1725.552300
2	28	0	33.000	3	0	southeast	4449.462000
3	33	0	22.705	0	0	northwest	21984.470610
4	32	0	28.880	0	0	northwest	3866.855200
5	31	1	25.740	0	0	southeast	3756.621600
6	46	1	33.440	1	0	southeast	8240.589600
7	37	1	27.740	3	0	northwest	7281.505600
8	37	0	29.830	2	0	northeast	6406.410700
9	60	1	25.840	0	0	northwest	28923.136920

In [80]:

```
x=["bmi","children"]
y=["Yes","No"]
all_inputs=df[x]
all_classes=df["sex"]
```

In [81]:

```
(x_train,x_test,y_train,y_test)=train_test_split(all_inputs,all_classes,test_size=0.03)
```

In [82]:

```
clf=DecisionTreeClassifier(random_state=0)
```

In [83]:

```
clf.fit(x_train,y_train)
```

Out[83]:

```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=0)
```

In [84]:

```
score=clf.score(x_test,y_test)
print(score)
```

```
0.3902439024390244
```

Random Forest

In [85]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt ,seaborn as sns
df=pd.read_csv(r"C:\Users\Gowthami\Downloads\insurance.csv")
df
```

1	18	male	33.770	1	no	southeast	1725.552300
2	28	male	33.000	3	no	southeast	4449.462000
3	33	male	22.705	0	no	northwest	21984.470610
4	32	male	28.880	0	no	northwest	3866.855200
5	31	female	25.740	0	no	southeast	3756.621600
6	46	female	33.440	1	no	southeast	8240.589600
7	37	female	27.740	3	no	northwest	7281.505600
8	37	male	29.830	2	no	northeast	6406.410700
9	60	female	25.840	0	no	northwest	28923.136920
10	25	male	26.220	0	no	northeast	2721.320800
11	62	female	26.290	0	yes	southeast	27808.725100
12	23	male	34.400	0	no	southwest	1826.843000
13	56	female	39.820	0	no	southeast	11090.717800

In [86]:

```
df.shape
```

Out[86]:

```
(1338, 7)
```

In [87]:

```
df['region'].value_counts()
```

Out[87]:

```
region
southeast    364
southwest    325
northwest    325
northeast    324
Name: count, dtype: int64
```

In [88]:

```
df['bmi'].value_counts()
```

Out[88]:

```
bmi
32.300    13
28.310     9
30.495     8
30.875     8
31.350     8
30.800     8
34.100     8
28.880     8
33.330     7
35.200     7
25.800     7
32.775     7
27.645     7
32.110     7
38.060     7
25.460     7
30.590     7
```

In [89]:

```
m={"sex":{"female":1,"male":0}}
df=df.replace(m)
print(df)
```

```
34     28    0  36.400         1    yes  southwest  51194.559140
35     19    0  20.425         0     no  northwest   1625.433750
36     62    1  32.965         3     no  northwest  15612.193350
37     26    0  20.800         0     no  southwest   2302.300000
38     35    0  36.670         1    yes  northeast  39774.276300
39     60    0  39.900         0    yes  southwest  48173.361000
40     24    1  26.600         0     no  northeast   3046.062000
41     31    1  36.630         2     no  southeast   4949.758700
42     41    0  21.780         1     no  southeast   6272.477200
43     37    1  30.800         2     no  southeast   6313.759000
44     38    0  37.050         1     no  northeast   6079.671500
45     55    0  37.300         0     no  southwest  20630.283510
46     18    1  38.665         2     no  northeast   3393.356350
47     28    1  34.770         0     no  northwest   3556.922300
48     60    1  24.530         0     no  southeast  12629.896700
49     36    0  35.200         1    yes  southeast  38709.176000
50     18    1  35.625         0     no  northeast   2211.130750
51     21    1  33.630         2     no  northwest   3579.828700
52     48    0  28.000         1    yes  southwest  23568.272000
53     36    0  34.430         0    yes  southeast  37742.575700
```

In [90]:

```
n={"smoker":{"yes":1,"no":0}}
df=df.replace(n)
print(df)
```

```
25     59    1  27.720         3       0  southeast  14001.133800
26     63    1  23.085         0       0  northeast  14451.835150
27     55    1  32.775         2       0  northwest  12268.632250
28     23    0  17.385         1       0  northwest   2775.192150
29     31    0  36.300         2       1  southwest  38711.000000
30     22    0  35.600         0       1  southwest  35585.576000
31     18    1  26.315         0       0  northeast   2198.189850
32     19    1  28.600         5       0  southwest   4687.797000
33     63    0  28.310         0       0  northwest  13770.097900
34     28    0  36.400         1       1  southwest  51194.559140
35     19    0  20.425         0       0  northwest   1625.433750
36     62    1  32.965         3       0  northwest  15612.193350
37     26    0  20.800         0       0  southwest   2302.300000
38     35    0  36.670         1       1  northeast  39774.276300
39     60    0  39.900         0       1  southwest  48173.361000
40     24    1  26.600         0       0  northeast   3046.062000
41     31    1  36.630         2       0  southeast   4949.758700
42     41    0  21.780         1       0  southeast   6272.477200
43     37    1  30.800         2       0  southeast   6313.759000
44     38    0  37.050         1       0  northeast   6079.671500
```

In [91]:

```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[91]:

RandomForestClassifier

RandomForestClassifier()

In [92]:

```
rf=RandomForestClassifier()
params={'max_depth':[2,3,5,20],
 'min_samples_leaf':[5,10,20,50,100,200],
 'n_estimators':[10,25,30,50,100,200]}
```

In [93]:

```
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rf,param_grid=params,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

Out[93]:

GridSearchCV
estimator: RandomForestClassifier
RandomForestClassifier

In [94]:

```
grid_search.best_score_
```

Out[94]:

```
0.5258279594437787
```

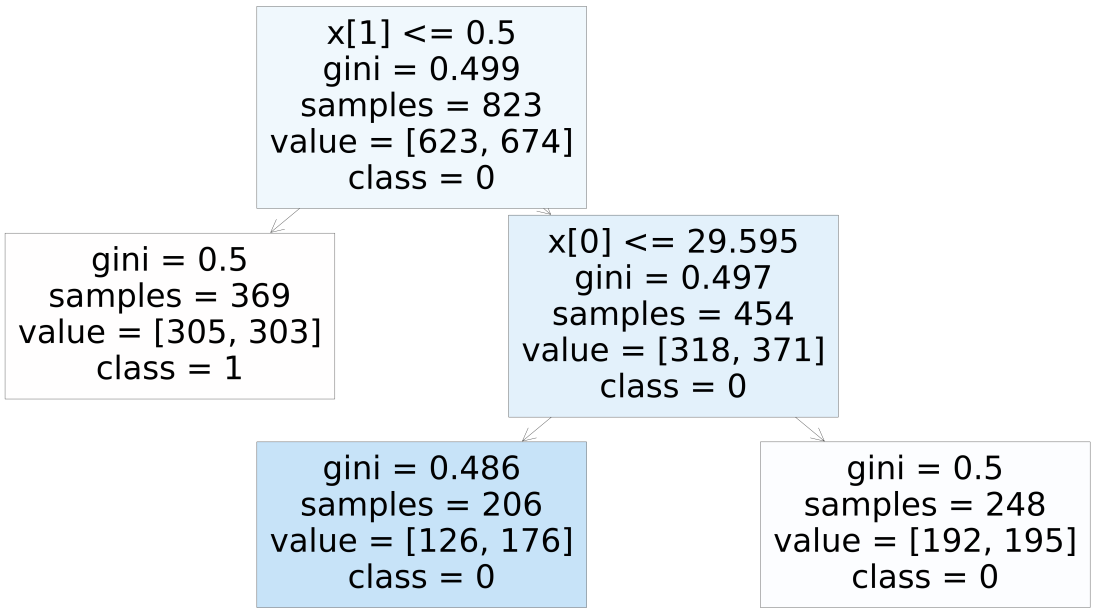
In [95]:

```
rf_best=grid_search.best_estimator_
print(rf_best)
```

```
RandomForestClassifier(max_depth=2, min_samples_leaf=200, n_estimators=10)
```

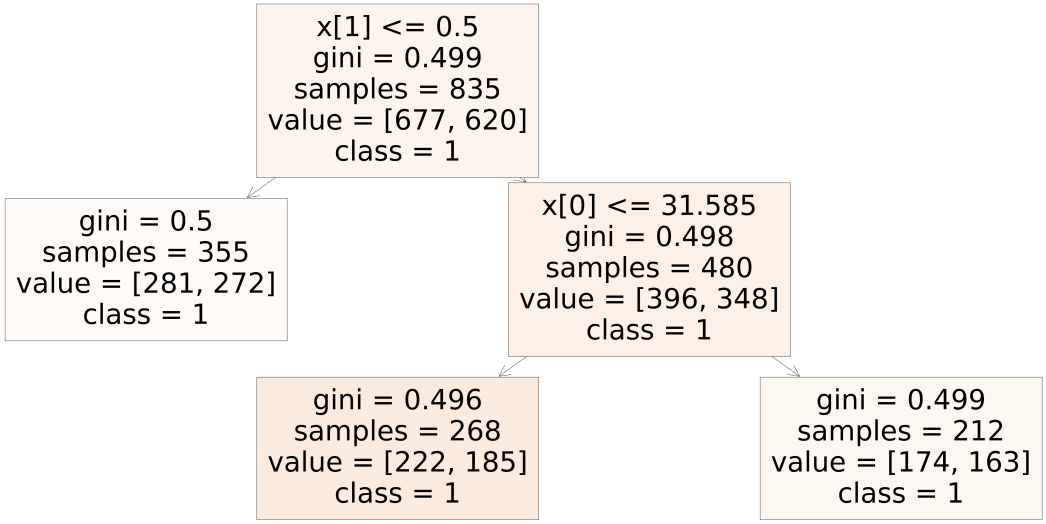
In [96]:

```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,40))
plot_tree(rf_best.estimators_[4],class_names=['1','0'],filled=True);
```



In [97]:

```
from sklearn.tree import plot_tree
plt.figure(figsize=(70,30))
plot_tree(rf_best.estimators_[6],class_names=["1","0"],filled=True);
```



In [98]:

```
rf_best.feature_importances_
```

Out[98]:

```
array([0.47680759, 0.52319241])
```

In [99]:

```
rf=RandomForestClassifier(random_state=0)
```

In [100]:

```
rf.fit(x_train,y_train)
```

Out[100]:

RandomForestClassifier

RandomForestClassifier(random_state=0)

In [101]:

```
score=rf.score(x_test,y_test)
print(score)
```

```
0.4146341463414634
```

In [ ]: