Problem statement:To predict How Best the data fits

1. Data collection

In [33]:

```
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 [34]:

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

Out[34]:

	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

2.Data cleaning and Preprocessing

#Exploratory data anlysis

In [35]:

```
df.head()
```

Out[35]:

	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 [36]:

```
df.tail()
```

Out[36]:

	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 [37]:

```
df.shape
```

Out[37]:

```
(1338, 7)
```

In [38]:

```
df.describe
```

Out[38]:

```
<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 [39]:

```
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 [40]:

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

Out[40]:

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

In [41]:

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

Out[41]:

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

In [42]:

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

Out[42]:

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

In [43]:

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

Out[43]:

	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

In [44]:

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

Out[44]:

	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

In [45]:

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

Out[45]:

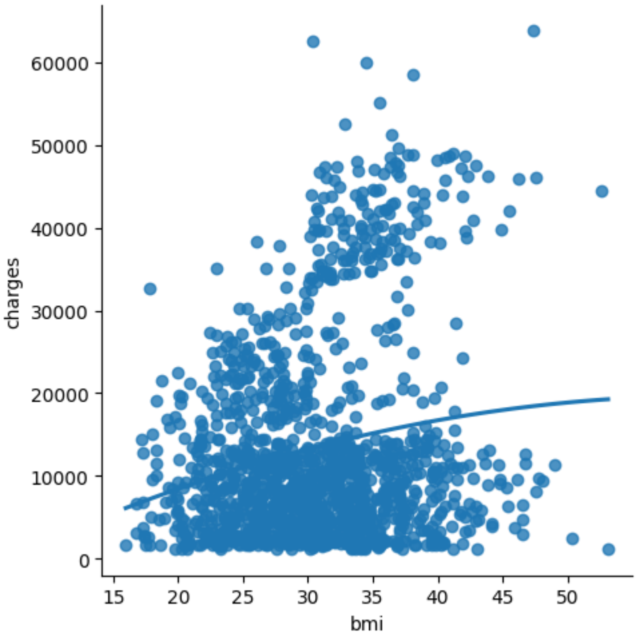
	age	sex	bmi	children	smoker	region	charges
0	19	1	27.900	0	1	1	16884.92400
1	18	0	33.770	1	0	2	1725.55230
2	28	0	33.000	3	0	2	4449.46200
3	33	0	22.705	0	0	3	21984.47061
4	32	0	28.880	0	0	3	3866.85520
...	...	...	...	...	...	...	...
1333	50	0	30.970	3	0	3	10600.54830
1334	18	1	31.920	0	0	4	2205.98080
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	3	29141.36030

1338 rows × 7 columns

3.Data Visualization

In [46]:

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



In [47]:

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

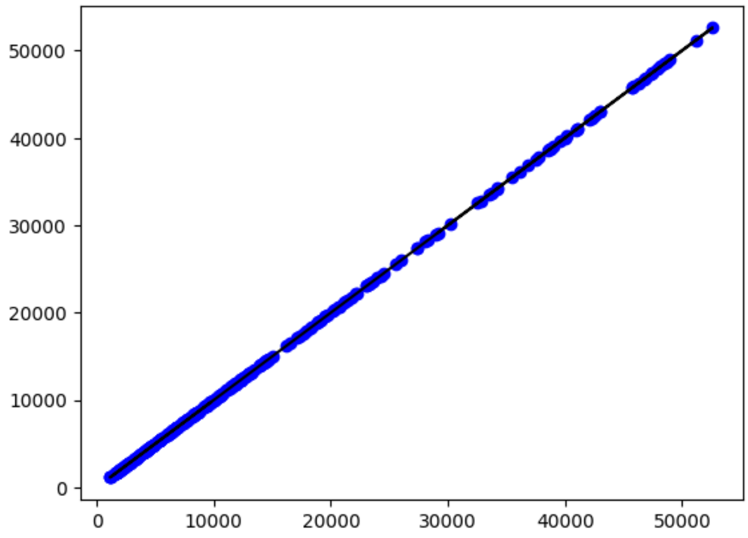
In [48]:

```
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 [49]:

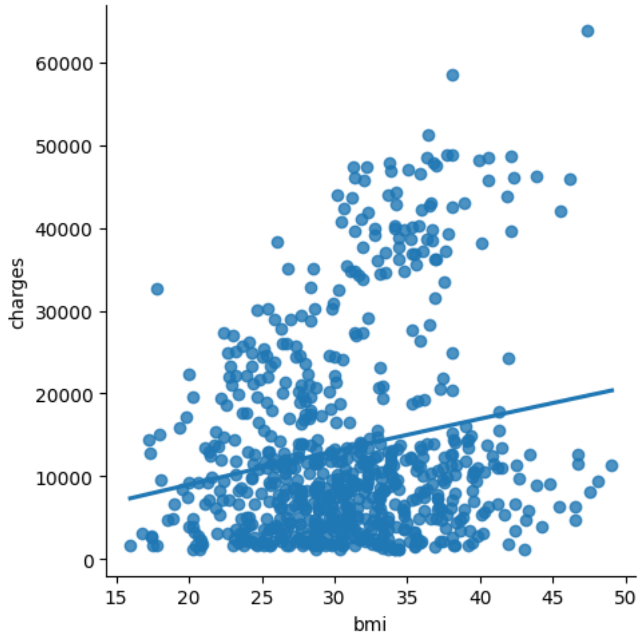
```
y_pred=lr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```



working with subset of data

In [50]:

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



In [51]:

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

In [52]:

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

In [53]:

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

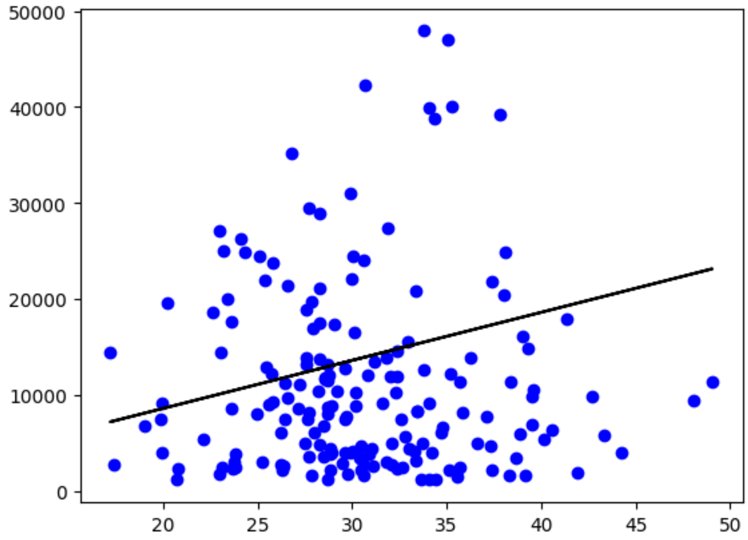
In [54]:

```
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))
```

```
-0.1630229146000015
```

In [55]:

```
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 [56]:

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

In [57]:

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

```
-0.1630229146000015
```

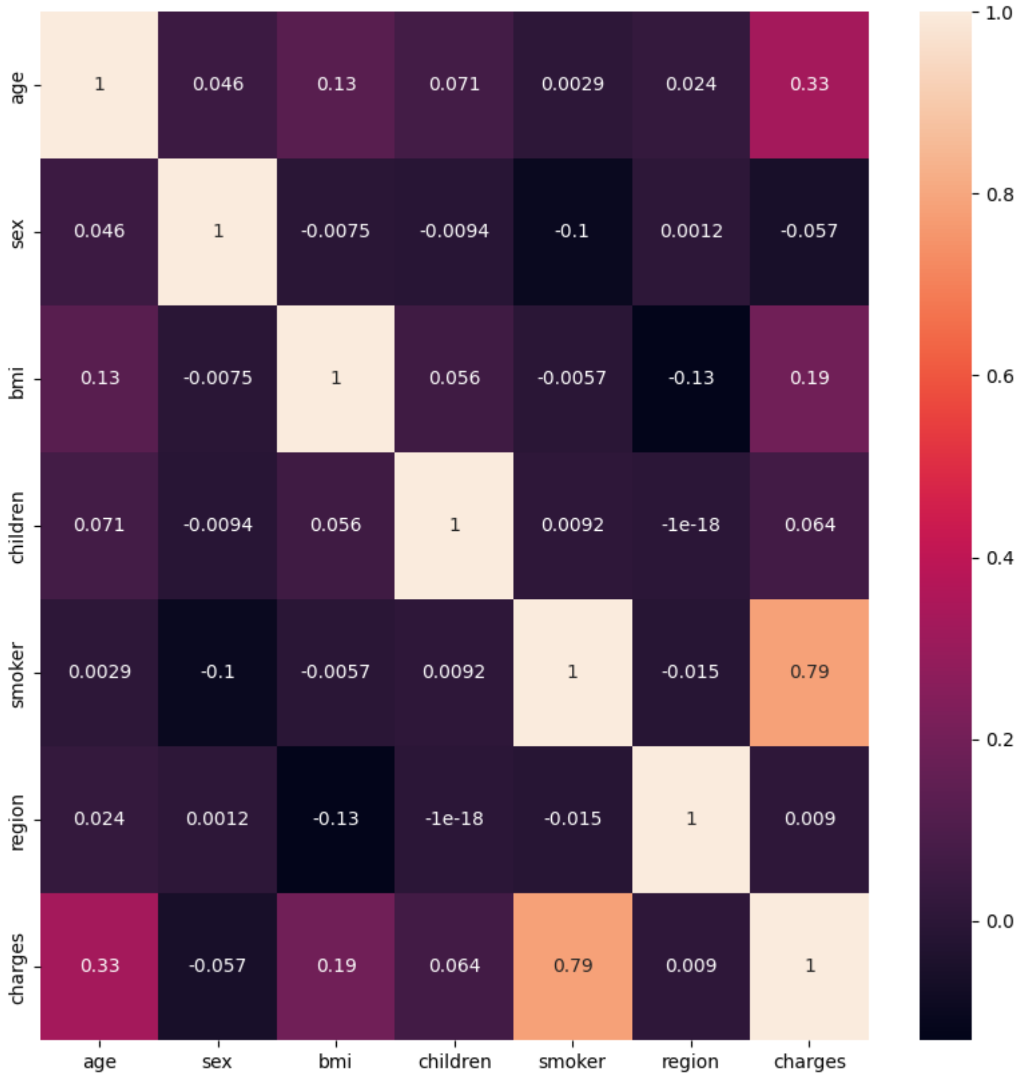
Ridge Regression

In [58]:

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

In [59]:

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



	age	sex	bmi	children	smoker	region	charges
age	1	0.046	0.13	0.071	0.0029	0.024	0.33
sex	0.046	1	-0.0075	-0.0094	-0.1	0.0012	-0.057
bmi	0.13	-0.0075	1	0.056	-0.0057	-0.13	0.19
children	0.071	-0.0094	0.056	1	0.0092	-1e-18	0.064
smoker	0.0029	-0.1	-0.0057	0.0092	1	-0.015	0.79
region	0.024	0.0012	-0.13	-1e-18	-0.015	1	0.009
charges	0.33	-0.057	0.19	0.064	0.79	0.009	1

In [60]:

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

In [61]:

```
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 [62]:

```
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 [63]:

```
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(test_score_ridge))
```

```

Ridge Model:

The train score for ridge model is 0.09109639711159634
The test score for ridge model is 0.08490538609860176
```

In [64]:

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

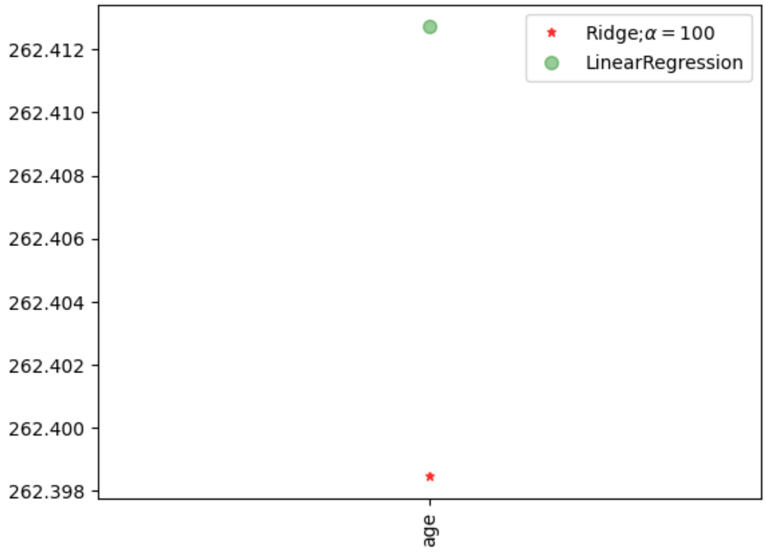
Out[64]:

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

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

In [66]:

```
 plt.plot(features,ridgeReg.coef_,alpha=0.7,linestyle='none',marker="*",markersize=5,color='red',label=r'Ridge;$\alpha=100$')
plt.plot(features,lr.coef_,alpha=0.4,linestyle='none',marker="o",markersize=7,color='green',label='LinearRegression')
plt.xticks(rotation=90)
plt.legend()
plt.show()
```



Lasso Regression

In [67]:

```
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 0.09109639395809055
The test score for lasso model is 0.08490704421828055
```

In [68]:

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

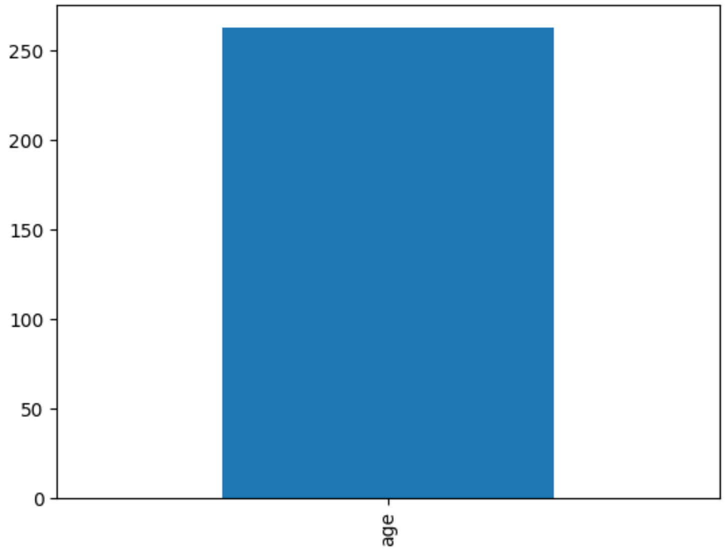
Out[68]:

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

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

In [69]:

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



In [70]:

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

In [71]:

```
#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
0.08490538609884779
```

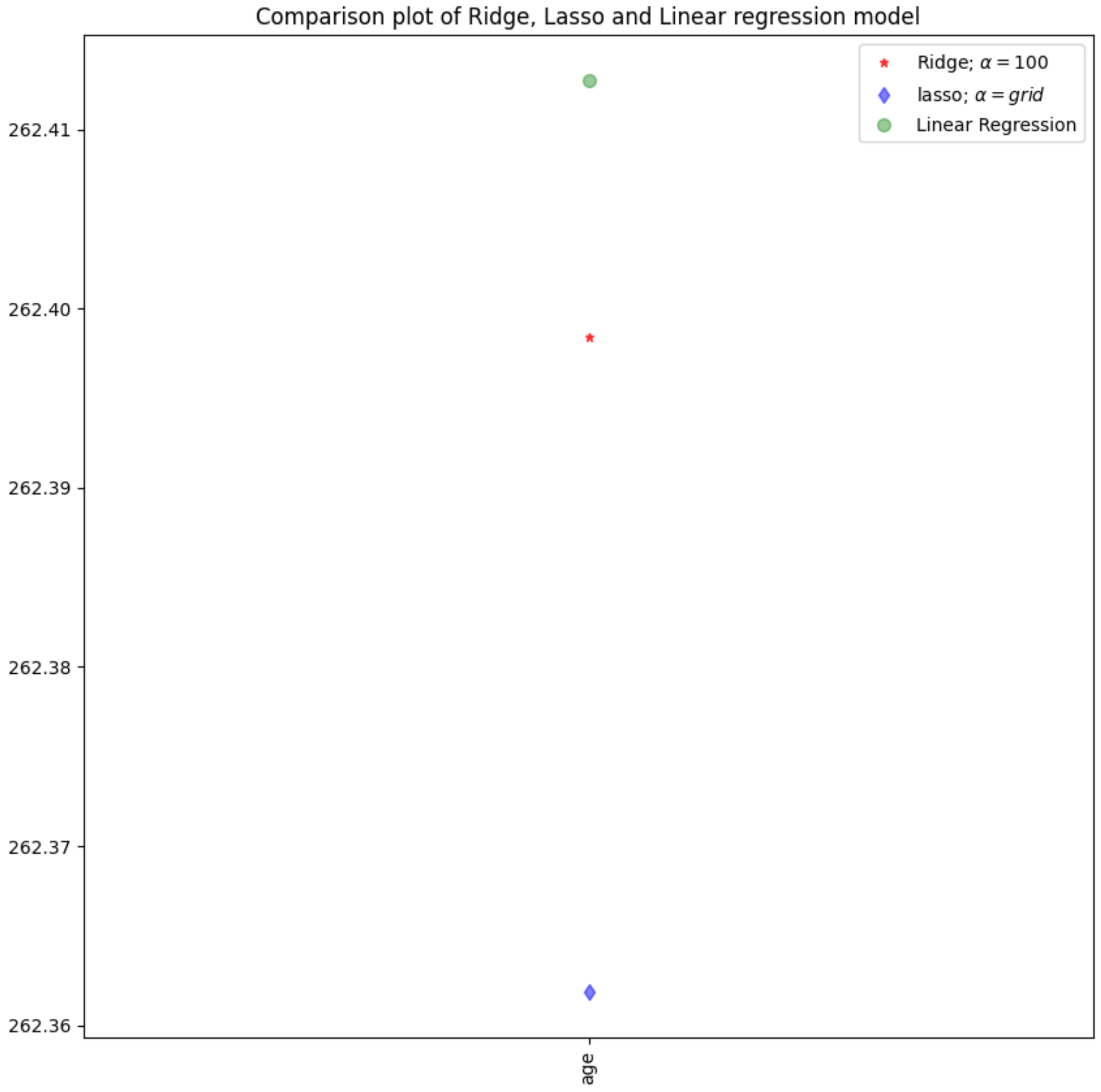
In [72]:

```
#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))
```

```
0.09109639395809055
0.08490704421828055
```

In [73]:

```
plt.figure(figsize = (10, 10))
#add plot for ridge regression
plt.plot(features,ridgeReg.coef_,alpha=0.7,linestyle='none',marker='*',markersize=5,color='red',label=r'Ridge; $\alpha=100$')
#add plot for lasso regression
plt.plot(lasso_cv.coef_,alpha=0.5,linestyle='none',marker='d',markersize=6,color='blue',label=r'lasso; $\alpha = grid$')
#add plot for linear model
plt.plot(features,lr.coef_,alpha=0.4,linestyle='none',marker='o',markersize=7,color='green',label='Linear Regression')
#rotate axis
plt.xticks(rotation = 90)
plt.legend()
plt.title("Comparison plot of Ridge, Lasso and Linear regression model")
plt.show()
```



ElasticNet Regression

In [74]:

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

In [75]:

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

```
[261.74450967]
3115.083177426244
```

In [76]:

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

In [77]:

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

```
135077142.70714515
```

In [78]:

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

```
0.09109580670592365
```

Logistic Regression

In [79]:

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

In [80]:

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

Out[80]:

	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 [81]:

```
df.shape
```

Out[81]:

```
(1338, 7)
```

In [82]:

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

In [83]:

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

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

In [84]:

```
df.head()
```

Out[84]:

	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 [85]:

```
df.describe
```

```
42      41    male  21.780         1     no  southeast   6272.477200
43      37  female  30.800         2     no  southeast   6313.759000
44      38    male  37.050         1     no  northeast   6079.671500
45      55    male  37.300         0     no  southwest  20630.283510
46      18  female  38.665         2     no  northeast   3393.356350
47      28  female  34.770         0     no  northwest   3556.922300
48      60  female  24.530         0     no  southeast  12629.896700
49      36    male  35.200         1    yes  southeast  38709.176000
50      18  female  35.625         0     no  northeast   2211.130750
51      21  female  33.630         2     no  northwest   3579.828700
52      48    male  28.000         1    yes  southwest  23568.272000
53      36    male  34.430         0    yes  southeast  37742.575700
54      40  female  28.690         3     no  northwest   8059.679100
55      58    male  36.955         2    yes  northwest  47496.494450
56      58  female  31.825         2     no  northeast  13607.368750
57      18    male  31.680         2    yes  southeast  34303.167200
58      53  female  22.880         1    yes  southeast  23244.790200
59      34  female  37.335         2     no  northwest   5989.523650
60      43    male  27.360         3     no  northeast   8606.217400
61      25    male  33.660         4     no  southeast   4504.662400
```

In [86]:

```
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 [87]:

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

Out[87]:

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

In [88]:

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

```
  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
 10  25    male  26.220  0  0  northeast   2721.320800
 11  62  female  26.290  0  1  southeast  27808.725100
 12  23    male  34.400  0  0  southwest   1826.843000
 13  56  female  39.820  0  0  southeast  11090.717800
 14  27    male  42.130  0  1  southeast  39611.757700
```

In [89]:

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

Out[89]:

```
    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 [90]:

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

```
  1  18  0  33.770  1  0  1   1725.552300
  2  28  0  33.000  3  0  1   4449.462000
  3  33  0  22.705  0  0  4  21984.470610
  4  32  0  28.880  0  0  4   3866.855200
  5  31  1  25.740  0  0  1   3756.621600
  6  46  1  33.440  1  0  1   8240.589600
  7  37  1  27.740  3  0  4   7281.505600
  8  37  0  29.830  2  0  3   6406.410700
  9  60  1  25.840  0  0  4  28923.136920
 10  25  0  26.220  0  0  3   2721.320800
 11  62  1  26.290  0  1  1  27808.725100
 12  23  0  34.400  0  0  2   1826.843000
 13  56  1  39.820  0  0  1  11090.717800
```

In [91]:

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

In [92]:

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

In [93]:

```
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))
```

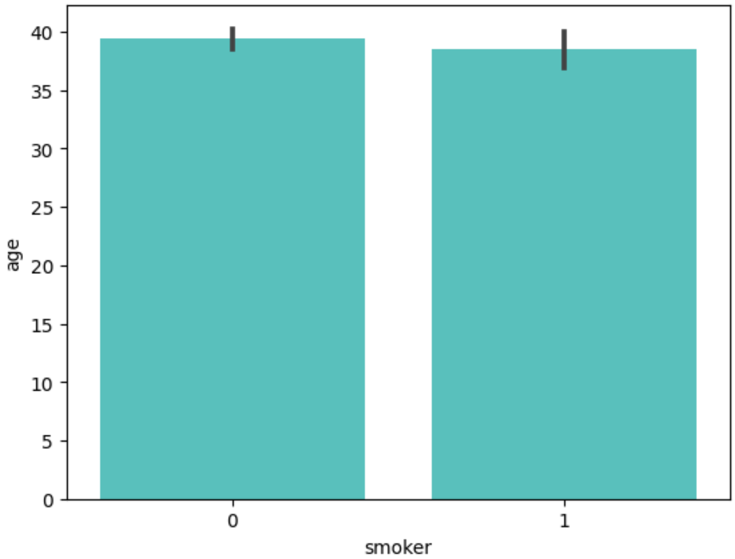
```
The Feature Matrix has 1338 Rows and 4 columns(s)
The Target Matrix has 1338 Rows and 1 columns(s)
```

In [94]:

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

In [95]:

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



In [96]:

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

In [97]:

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

In [98]:

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

In [99]:

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

In [101]:

```
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 [102]:

```
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 [104]:

```
"The Model says the probability of the observation we passed belonging to class[0] %s" " "%(algorithm.predict_proba(observation)[0][0]))
```

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

In [107]:

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

```
The Model says the probability of the observation we passed belonging to class['g'] Is 0.19429241286686041
```

In [108]:

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

In [109]:

```
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))
```

```
0.7611940298507462

C:\Users\jyothi reddy\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\utils\validation.py:1143: DataConve
rsionWarning: 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 [110]:

```
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 [111]:

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

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 [112]:

```
df.shape
```

Out[112]:

```
(1338, 7)
```

In [113]:

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

Out[113]:

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

In [114]:

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

Out[114]:

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

In [115]:

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

20	60	1	36.005	0	no	northeast	13228.846950
21	30	1	32.400	1	no	southwest	4149.736000
22	18	0	34.100	0	no	southeast	1137.011000
23	34	1	31.920	1	yes	northeast	37701.876800
24	37	0	28.025	2	no	northwest	6203.901750
25	59	1	27.720	3	no	southeast	14001.133800
26	63	1	23.085	0	no	northeast	14451.835150
27	55	1	32.775	2	no	northwest	12268.632250
28	23	0	17.385	1	no	northwest	2775.192150
29	31	0	36.300	2	yes	southwest	38711.000000
30	22	0	35.600	0	yes	southwest	35585.576000
31	18	1	26.315	0	no	northeast	2198.189850
32	19	1	28.600	5	no	southwest	4687.797000

In [116]:

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

19	30	0	35.300	0	1	southwest	36837.467000
20	60	1	36.005	0	0	northeast	13228.846950
21	30	1	32.400	1	0	southwest	4149.736000
22	18	0	34.100	0	0	southeast	1137.011000
23	34	1	31.920	1	1	northeast	37701.876800
24	37	0	28.025	2	0	northwest	6203.901750
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

In [117]:

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

In [118]:

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

In [119]:

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

In [120]:

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

Out[120]:

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

In [121]:

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

```
0.5853658536585366
```

Random Forest

In [122]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt ,seaborn as sns
```

In [123]:

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

```
34   28    male  36.400   1   yes  southwest  51194.559140
35   19    male  20.425   0    no  northwest   1625.433750
36   62  female  32.965   3    no  northwest  15612.193350
37   26    male  20.800   0    no  southwest   2302.300000
38   35    male  36.670   1   yes  northeast  39774.276300
39   60    male  39.900   0   yes  southwest  48173.361000
40   24  female  26.600   0    no  northeast   3046.062000
41   31  female  36.630   2    no  southeast   4949.758700
42   41    male  21.780   1    no  southeast   6272.477200
43   37  female  30.800   2    no  southeast   6313.759000
44   38    male  37.050   1    no  northeast   6079.671500
45   55    male  37.300   0    no  southwest  20630.283510
46   18  female  38.665   2    no  northeast   3393.356350
```

In [124]:

```
df.shape
```

Out[124]:

```
(1338, 7)
```

In [125]:

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

Out[125]:

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

In [126]:

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

```
25.800     7
32.775     7
27.645     7
32.110     7
38.060     7
25.460     7
30.590     7
27.360     7
24.320     7
34.800     7
27.500     6
19.950     6
29.920     6
30.115     6
26.600     6
30.200     6
35.530     6
33.630     6
28.595     6
37.100     6
```

In [127]:

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

```
      age  sex     bmi  children smoker     region       charges
0      19    1  27.900         0    yes  southwest  16884.924000
1      18    0  33.770         1     no  southeast   1725.552300
2      28    0  33.000         3     no  southeast   4449.462000
3      33    0  22.705         0     no  northwest  21984.470610
4      32    0  28.880         0     no  northwest   3866.855200
5      31    1  25.740         0     no  southeast   3756.621600
6      46    1  33.440         1     no  southeast   8240.589600
7      37    1  27.740         3     no  northwest   7281.505600
8      37    0  29.830         2     no  northeast   6406.410700
9      60    1  25.840         0     no  northwest  28923.136920
10     25    0  26.220         0     no  northeast   2721.320800
11     62    1  26.290         0    yes  southeast  27808.725100
12     23    0  34.400         0     no  southwest   1826.843000
13     56    1  39.820         0     no  southeast  11090.717800
14     27    0  42.130         0    yes  southeast  39611.757700
15     19    0  24.600         1     no  southwest   1837.237000
16     52    1  30.780         1     no  northeast  10797.336200
17     23    0  23.845         0     no  northeast   2395.171550
```

In [128]:

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

```
      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
10     25    0  26.220         0       0  northeast   2721.320800
11     62    1  26.290         0       1  southeast  27808.725100
12     23    0  34.400         0       0  southwest   1826.843000
13     56    1  39.820         0       0  southeast  11090.717800
14     27    0  42.130         0       1  southeast  39611.757700
15     19    0  24.600         1       0  southwest   1837.237000
16     52    1  30.780         1       0  northeast  10797.336200
17     23    0  23.845         0       0  northeast   2395.171550
```

In [129]:

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

Out[129]:

```
▾ RandomForestClassifier
RandomForestClassifier()
```

In [130]:

```
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 [132]:

```
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[132]:

```
▸            GridSearchCV
▸ estimator: RandomForestClassifier
      ▸ RandomForestClassifier
```

In [133]:

```
grid_search.best_score_
```

Out[133]:

```
0.5096254446536932
```

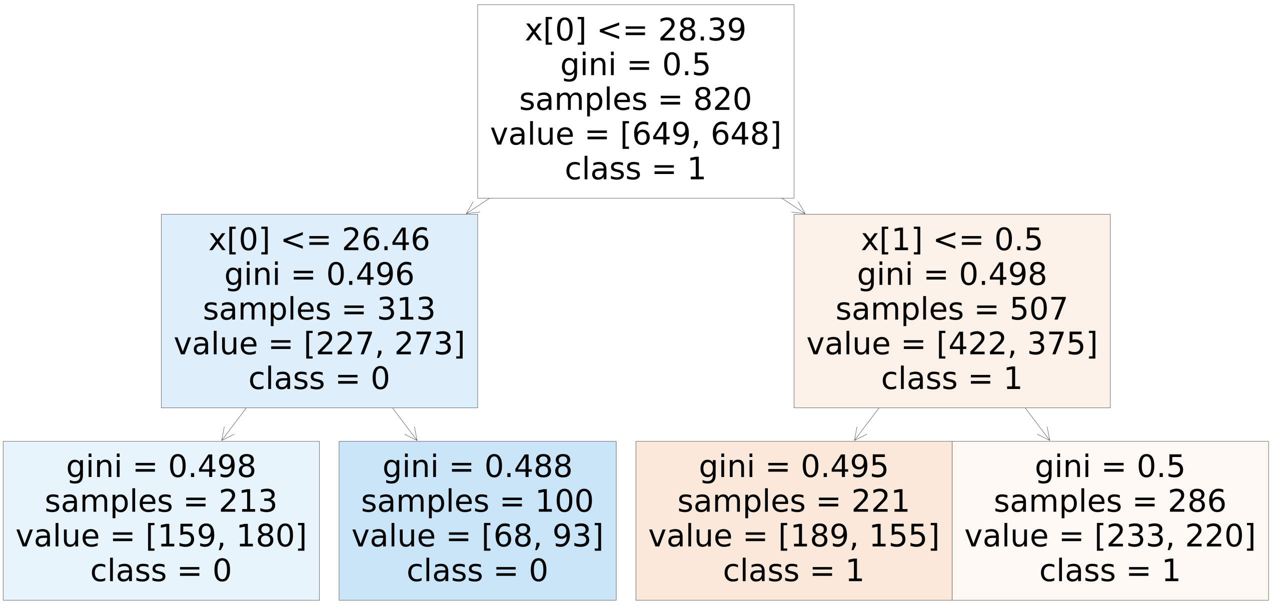
In [134]:

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

```
RandomForestClassifier(max_depth=2, min_samples_leaf=100, n_estimators=10)
```

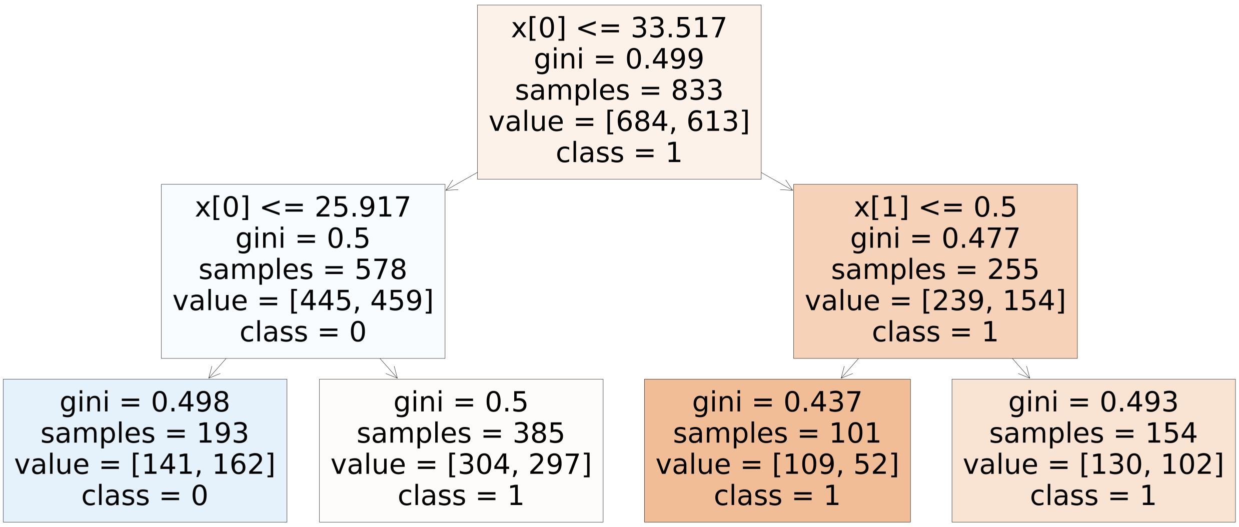
In [135]:

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



In [136]:

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



In [137]:

```
rf_best.feature_importances_
```

Out[137]:

```
array([0.8019108, 0.1980892])
```

In [140]:

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

In [141]:

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

Out[141]:

```
▾        RandomForestClassifier
RandomForestClassifier(random_state=0)
```

In [142]:

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

```
0.5609756097560976
```

In [ ]: