In [28]:

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

In [29]:

```
data=pd.read_csv(r"C:\Users\karis\Documents\breast_cancer.csv")
data.head()
```

Out[29]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavit
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	

5 rows × 32 columns

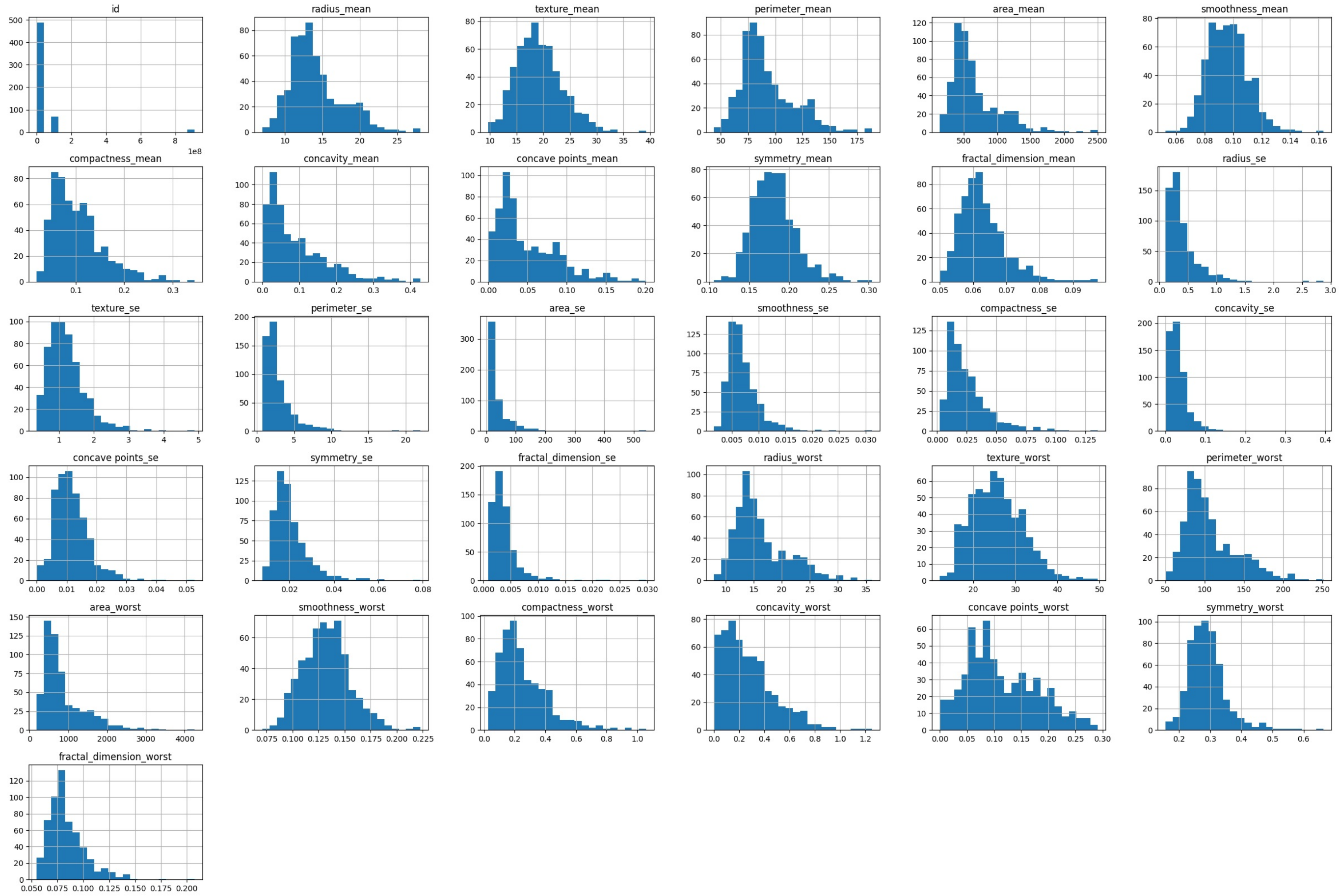
In [30]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 32 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   id                       569 non-null    int64
 1   diagnosis                569 non-null    object
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
dtypes: float64(30), int64(1), object(1)
memory usage: 142.4+ KB
```

In [31]:

```
data.hist(bins=22,figsize=(30,20))
plt.show()
```



In [32]:

```
data.describe()
```

Out[32]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.0
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.0
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.0
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.0
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.0
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.0
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.1
max	9.113205e+08	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.4

8 rows × 31 columns

In [33]:

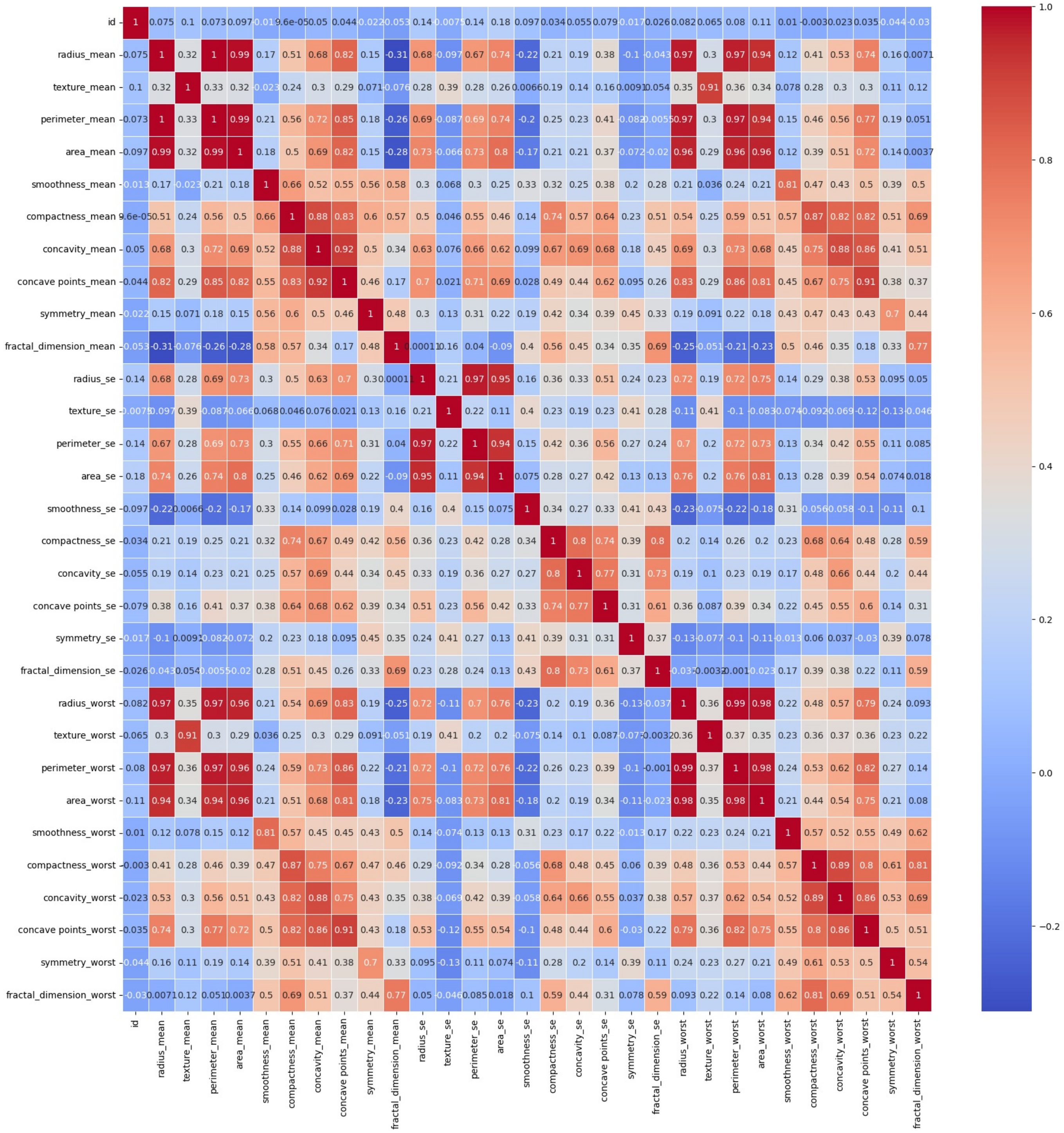
```
data.isnull().values.any()
```

Out[33]:

```
np.False_
```

In [34]:

```
numeric_data = data.select_dtypes(include=['float64', 'int64'])
correat = numeric_data.corr()
plt.figure(figsize=(20, 20))
sns.heatmap(correat, annot=True, cmap="coolwarm", linewidths=0.5)
plt.show()
```



In [35]:

```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
data['diagnosis'] = label_encoder.fit_transform(data['diagnosis'])
data
```

Out[35]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	conca
0	842302	1	17.99	10.38	122.80	1001.0	0.11840	0.27760	
1	842517	1	20.57	17.77	132.90	1326.0	0.08474	0.07864	
2	84300903	1	19.69	21.25	130.00	1203.0	0.10960	0.15990	
3	84348301	1	11.42	20.38	77.58	386.1	0.14250	0.28390	
4	84358402	1	20.29	14.34	135.10	1297.0	0.10030	0.13280	
...	...	...	...	...	...	...	...	...	
564	926424	1	21.56	22.39	142.00	1479.0	0.11100	0.11590	
565	926682	1	20.13	28.25	131.20	1261.0	0.09780	0.10340	
566	926954	1	16.60	28.08	108.30	858.1	0.08455	0.10230	
567	927241	1	20.60	29.33	140.10	1265.0	0.11780	0.27700	
568	92751	0	7.76	24.54	47.92	181.0	0.05263	0.04362	

569 rows × 32 columns

In [36]:
```
data.describe()
```

Out[36]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mear
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.00000
mean	3.037183e+07	0.372583	14.127292	19.289649	91.969033	654.889104	0.096360	0.10434
std	1.250206e+08	0.483918	3.524049	4.301036	24.298981	351.914129	0.014064	0.05281
min	8.670000e+03	0.000000	6.981000	9.710000	43.790000	143.500000	0.052630	0.01938
25%	8.692180e+05	0.000000	11.700000	16.170000	75.170000	420.300000	0.086370	0.06492
50%	9.060240e+05	0.000000	13.370000	18.840000	86.240000	551.100000	0.095870	0.09263
75%	8.813129e+06	1.000000	15.780000	21.800000	104.100000	782.700000	0.105300	0.13040
max	9.113205e+08	1.000000	28.110000	39.280000	188.500000	2501.000000	0.163400	0.34540

8 rows × 32 columns

In [37]:
```
data_Cancer = data[data["diagnosis"] == 1]
data_NonCancer = data[data["diagnosis"] == 0]
```

In [38]:
```
data_Cancer .describe()
```

Out[38]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean
count	2.120000e+02	212.0	212.000000	212.000000	212.000000	212.000000	212.000000	212.000000
mean	3.681805e+07	1.0	17.462830	21.604906	115.365377	978.376415	0.102898	0.145188
std	1.378965e+08	0.0	3.203971	3.779470	21.854653	367.937978	0.012608	0.053987
min	8.670000e+03	1.0	10.950000	10.380000	71.900000	361.600000	0.073710	0.046050
25%	8.613450e+05	1.0	15.075000	19.327500	98.745000	705.300000	0.094010	0.109600
50%	8.953665e+05	1.0	17.325000	21.460000	114.200000	932.000000	0.102200	0.132350
75%	8.911290e+06	1.0	19.590000	23.765000	129.925000	1203.750000	0.110925	0.172400
max	9.112962e+08	1.0	28.110000	39.280000	188.500000	2501.000000	0.144700	0.345400

8 rows × 32 columns

In [39]:
```
data_NonCancer.describe()
```

Out[39]:

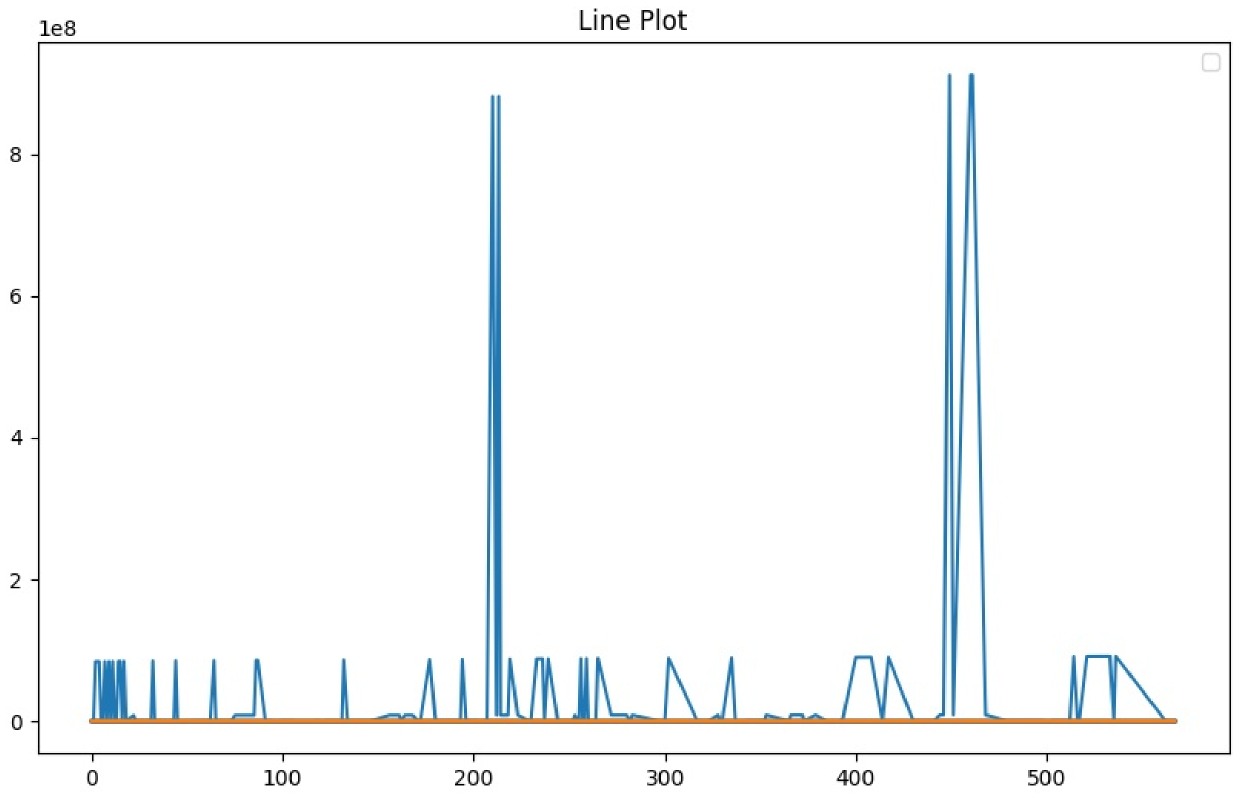
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean
count	3.570000e+02	357.0	357.000000	357.000000	357.000000	357.000000	357.000000	357.000000
mean	2.654382e+07	0.0	12.146524	17.914762	78.075406	462.790196	0.092478	0.080085
std	1.167397e+08	0.0	1.780512	3.995125	11.807438	134.287118	0.013446	0.033750
min	8.913000e+03	0.0	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380
25%	8.746620e+05	0.0	11.080000	15.150000	70.870000	378.200000	0.083060	0.055620
50%	9.089160e+05	0.0	12.200000	17.390000	78.180000	458.400000	0.090760	0.075290
75%	8.812816e+06	0.0	13.370000	19.760000	86.100000	551.100000	0.100700	0.097550
max	9.113205e+08	0.0	17.850000	33.810000	114.600000	992.100000	0.163400	0.223900

8 rows × 32 columns

In [40]:

```
columns = data_Cancer.columns
plt.figure(figsize=(10,6))
for column in columns:
    plt.plot(data_Cancer[column])
plt.title('Line Plot')
plt.legend()
plt.show()
```

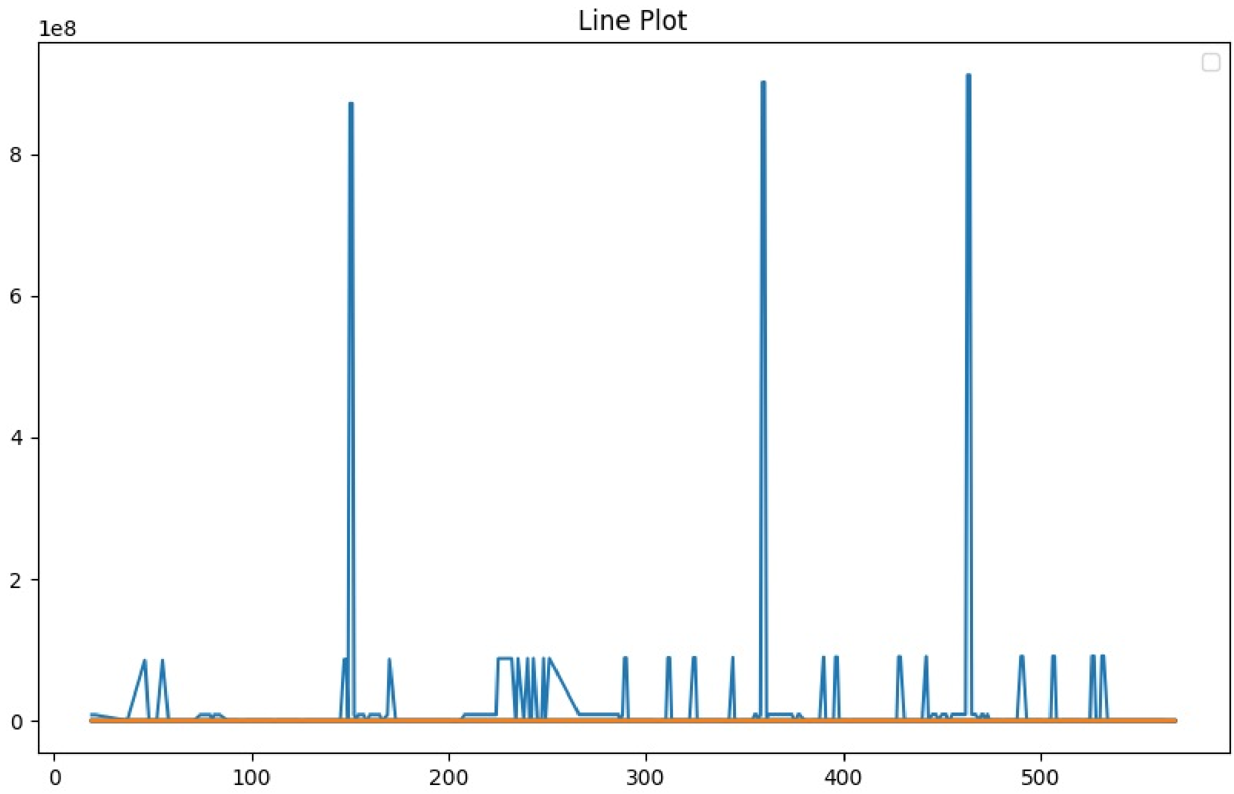
```
C:\Users\karis\AppData\Local\Temp\ipykernel_2568\1205333819.py:6: UserWarning: No artists with labels found to p
ut in legend.  Note that artists whose label start with an underscore are ignored when legend() is called with n
o argument.
  plt.legend()
```



In [41]:

```
columns = data_NonCancer.columns
plt.figure(figsize=(10,6))
for column in columns:
    plt.plot(data_NonCancer[column])
plt.title('Line Plot')
plt.legend()
plt.show()
```

```
C:\Users\karis\AppData\Local\Temp\ipykernel_2568\1270843764.py:6: UserWarning: No artists with labels found to p
ut in legend.  Note that artists whose label start with an underscore are ignored when legend() is called with n
o argument.
  plt.legend()
```



In [42]:
```
row_count = len(data_Cancer)
n = min(400, row_count)
cancer_sample = data_Cancer.sample(n=n)
```

In [43]:
```
new_dataset=pd.concat([cancer_sample,data_NonCancer],axis=0)
new_dataset.head()
```

Out[43]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	conc
196	875938	1	13.77	22.29	90.63	588.9	0.12000	0.1267	
70	859575	1	18.94	21.31	123.60	1130.0	0.09009	0.1029	
339	89812	1	23.51	24.27	155.10	1747.0	0.10690	0.1283	
449	911157302	1	21.10	20.52	138.10	1384.0	0.09684	0.1175	
330	896839	1	16.03	15.51	105.80	793.2	0.09491	0.1371	

5 rows × 32 columns

In [44]:
```
new_dataset.tail()
```

Out[44]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavit
558	925277	0	14.59	22.68	96.39	657.1	0.08473	0.13300	
559	925291	0	11.51	23.93	74.52	403.5	0.09261	0.10210	
560	925292	0	14.05	27.15	91.38	600.4	0.09929	0.11260	
561	925311	0	11.20	29.37	70.67	386.0	0.07449	0.03558	
568	92751	0	7.76	24.54	47.92	181.0	0.05263	0.04362	

5 rows × 32 columns

In [45]:
```
new_dataset.groupby("diagnosis").mean()
```

Out[45]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concav
diagnosis								
0	2.654382e+07	12.146524	17.914762	78.075406	462.790196	0.092478	0.080085	
1	3.681805e+07	17.462830	21.604906	115.365377	978.376415	0.102898	0.145188	

2 rows × 31 columns

In [46]:

```
X = new_dataset.drop(columns = "diagnosis",axis= 1)
Y= new_dataset["diagnosis"]
```

In [47]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)
```

In [48]:

```
print(X.shape,X_train.shape,X_test.shape)
```

```
(569, 31) (455, 31) (114, 31)
```

In [49]:

```
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split, RandomizedSearchCV
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score
rf = RandomForestClassifier(random_state=42, n_jobs=-1)
param_dist = {
    'n_estimators': [100, 200, 500],
    'max_features': ['sqrt', 'log2'],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'bootstrap': [True, False]
}
random_search = RandomizedSearchCV(estimator=rf,
                                   param_distributions=param_dist,
                                   n_iter=100,  # Number of parameter combinations to try
                                   scoring='roc_auc',
                                   cv=3,  # 3-fold cross-validation
                                   verbose=2,
                                   random_state=42,
                                   n_jobs=-1)
random_search.fit(X_train, Y_train)
best_rf = random_search.best_estimator_
Y_pred = best_rf.predict(X_test)
Y_prob = best_rf.predict_proba(X_test)[:, 1]
print("Classification Report:\n", classification_report(Y_test, Y_pred))
print("Confusion Matrix:\n", confusion_matrix(Y_test, Y_pred))
roc_auc = roc_auc_score(Y_test, Y_prob)
print(f"AUC-ROC Score: {roc_auc}")
print("Best Hyperparameters:\n", random_search.best_params_)
```

```
Fitting 3 folds for each of 100 candidates, totalling 300 fits
Classification Report:
               precision    recall  f1-score   support

           0       0.99      0.97      0.98        72
           1       0.95      0.98      0.96        42

    accuracy                           0.97       114
   macro avg       0.97      0.97      0.97       114
weighted avg       0.97      0.97      0.97       114

Confusion Matrix:
 [[70  2]
 [ 1 41]]
AUC-ROC Score: 0.998015873015873
Best Hyperparameters:
 {'n_estimators': 100, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_features': 'sqrt', 'max_depth': None,
'bootstrap': True}
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js