In [1]:
```
import pandas
df = pandas.read_csv('SmokingDataSet.csv')
df.head()
```

Out[1]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	av
0	Male	67.0	0	1	Yes	Private	Urban	
1	Male	80.0	0	1	Yes	Private	Rural	
2	Female	49.0	0	0	Yes	Private	Urban	
3	Female	79.0	1	0	Yes	Self-employed	Rural	
4	Male	81.0	0	0	Yes	Private	Urban	

In [2]:
```
obj_list = list(df.select_dtypes(include='object'))
obj_list
```

Out[2]:
```
['gender', 'ever_married', 'work_type', 'Residence_type', 'smoking
_status']
```

In [3]:
```
from sklearn import preprocessing
for i in obj_list:
    Encoder = preprocessing.LabelEncoder()
    df[i]= Encoder.fit_transform(df[i])
```

In [4]:
```
x = df.drop(columns=['stroke'],axis=1)
y = df['stroke']
```

In [5]:
```
from imblearn.over_sampling import RandomOverSampler
over_sampler = RandomOverSampler(sampling_strategy='minority')
x,y = over_sampler.fit_resample(x,y)
```

In [6]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25
```

In [7]:
```
from sklearn.linear_model import LogisticRegression
logistic_model = LogisticRegression(max_iter=1000)
logistic_model.fit(x_train,y_train)
logistic_pred = logistic_model.predict(x_test)
```

In [8]:
```
from sklearn.tree import DecisionTreeClassifier
DT_model = DecisionTreeClassifier()
DT_model.fit(x_train,y_train)
tree_pred = DT_model.predict(x_test)
```

In [9]:
```
from sklearn import svm
model_svm = svm.SVC()
model_svm.fit(x_train,y_train)
svm_pred = model_svm.predict(x_test)
```

In [10]:
```
from sklearn.ensemble import RandomForestClassifier
RF_model = RandomForestClassifier(n_estimators=1000)
RF_model.fit(x_train,y_train)
y_pred = RF_model.predict(x_test)
```

In [11]:
```
from sklearn.ensemble import VotingClassifier
final_model = VotingClassifier(
    estimators=[('lr',logistic_model), ('dt',DT_model),('svm',model
```

In [12]:
```
from sklearn.metrics import confusion_matrix
final_model.fit(x_train,y_train)
final_pred = final_model.predict(x_test)
cm_log = confusion_matrix(y_test,final_pred)
cm_log
```

Out[12]:
```
array([[858, 326],
       [200, 983]])
```

In [15]:
```
from sklearn.metrics import precision_score,recall_score,roc_auc_sc
print("Precision Score for Logistic model",precision_score(y_test,l
print("Precision Score for Decision model",precision_score(y_test,t
print("Precision Score for SVM model",precision_score(y_test,svm_pr
print("Precision Score for Ensemble model",precision_score(y_test,f
print("Precision Score for RandomForest model",precision_score(y_te

print("\nRecall Score for Logistic model",recall_score(y_test,logis
print("Recall Score for Decision model",recall_score(y_test,tree_pr
print("Recall Score for SVM model",recall_score(y_test,svm_pred))
print("Recall Score for Ensemble model",recall_score(y_test,final_p
print("Recall Score for RandomForest model",recall_score(y_test,y_p


print("\nROC-AUC Score for logistic model",roc_auc_score(y_test,log
print("ROC-AUC Score for Decision model",roc_auc_score(y_test,tree_
print("ROC-AUC Score for SVM model",roc_auc_score(y_test,svm_pred))
print("ROC-AUC Score for Ensemble model",roc_auc_score(y_test,final
print("ROC-AUC Score for RandomForest model",roc_auc_score(y_test,y
```

```
Precision Score for Logistic model 0.7389312977099237
Precision Score for Decision model 0.9403815580286169
Precision Score for SVM model 0.7257448433919023
Precision Score for Ensemble model 0.7509549274255156
Precision Score for RandomForest model 0.9817427385892116

Recall Score for Logistic model 0.8182586644125106
Recall Score for Decision model 1.0
Recall Score for SVM model 0.8030431107354185
Recall Score for Ensemble model 0.830938292476754
Recall Score for RandomForest model 1.0

ROC-AUC Score for logistic model 0.7647036565305797
ROC-AUC Score for Decision model 0.9683277027027026
ROC-AUC Score for SVM model 0.7499168256379796
ROC-AUC Score for Ensemble model 0.7778002273194581
ROC-AUC Score for RandomForest model 0.9907094594594595
```

In [16]:
```
from prettytable import PrettyTable
Comparision_table = PrettyTable(["Model", "Precision Score", "Recal
Comparision_table.add_row(["Logistic Model","0.73", "0.81", "0.76"]
Comparision_table.add_row(["Decision Model","0.94", "1.0", "0.96"])
Comparision_table.add_row(["SVM Model","0.72", "0.80", "0.74"])
Comparision_table.add_row(["Ensemble Model","0.75", "0.83", "0.77"]
Comparision_table.add_row(["RandomForest Model","0.98", "1.0", "0.9

print(Comparision_table)
```

```
+--------------------+-----------------+-------------+-----------
----+
|       Model        | Precision Score | Recall Score | ROC-AUC Sc
ore |
+--------------------+-----------------+-------------+-----------
----+
|   Logistic Model   |       0.73      |     0.81     |      0.76
|
|   Decision Model   |       0.94      |     1.0      |      0.96
|
|     SVM Model      |       0.72      |     0.80     |      0.74
|
|   Ensemble Model   |       0.75      |     0.83     |      0.77
|
| RandomForest Model |       0.98      |     1.0      |      0.99
|
+--------------------+-----------------+-------------+-----------
----+
```