Vivek A

Click here to view the github

Bio-Signal Analysis for Smoking

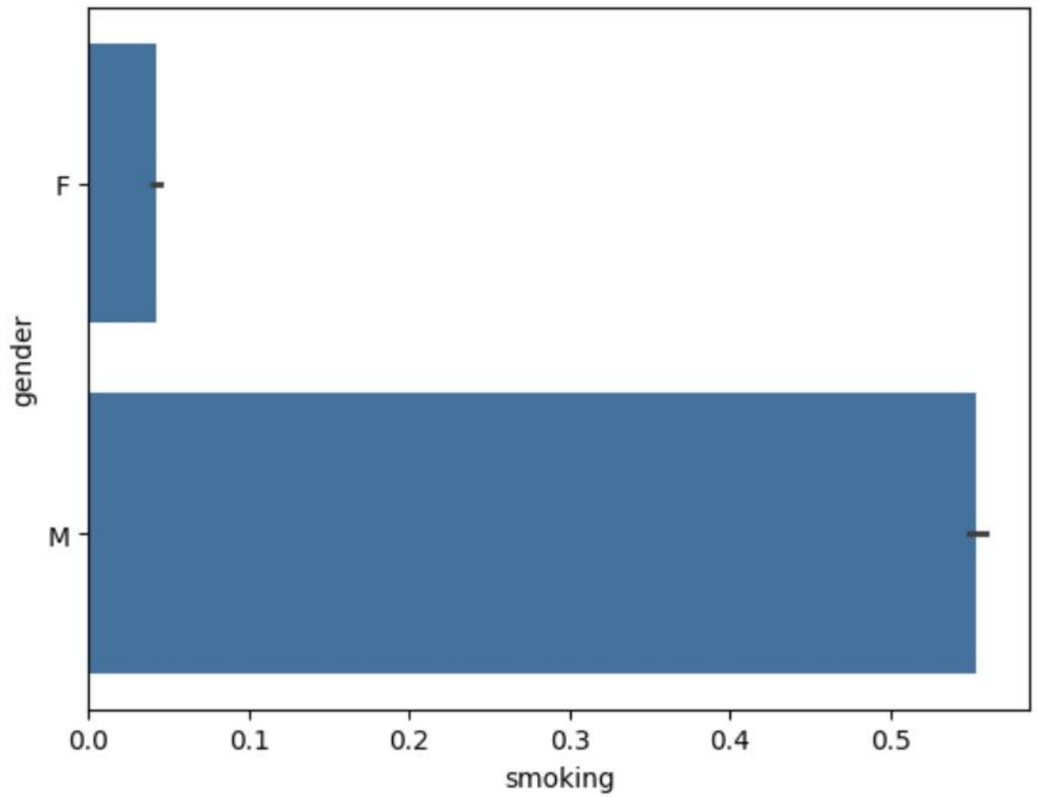
Data Science project

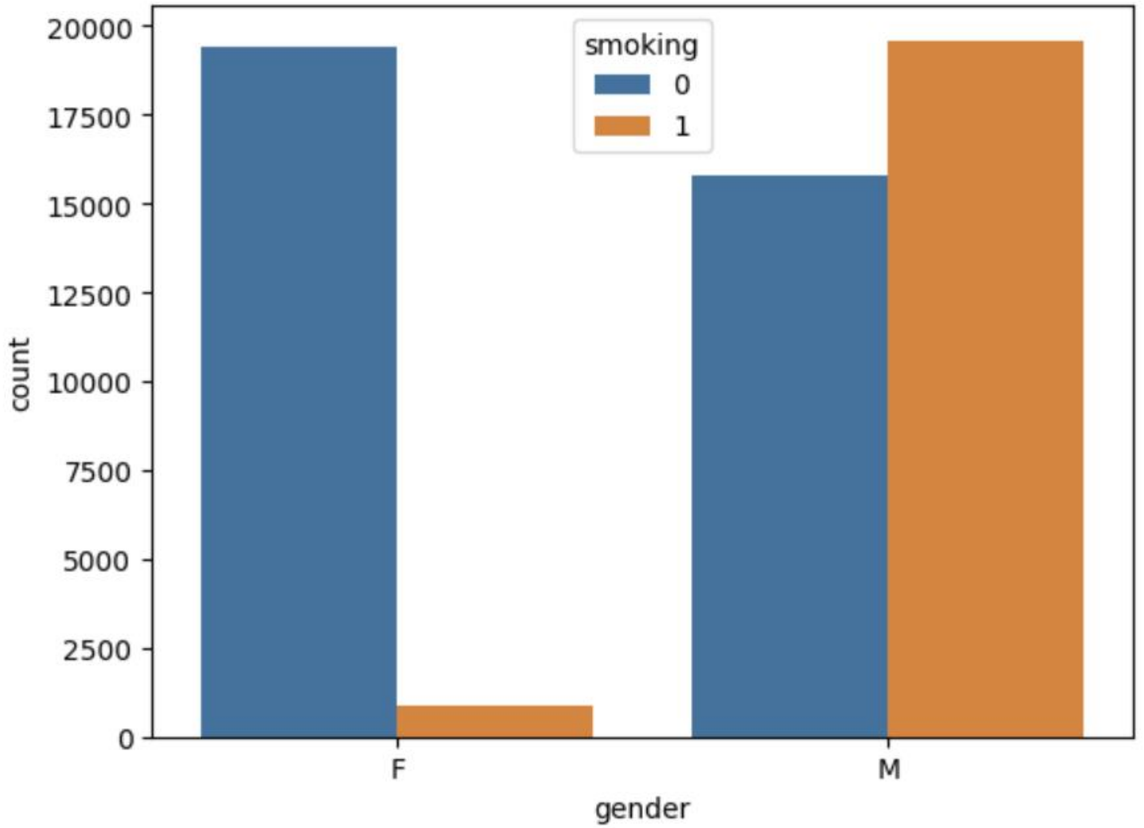
Problem statement

You are working as a data scientist in a global company. Over the years, the company has collected details and gathered a lot of information about individuals. The management wants to build an intelligent system from the data to determine the presence or absence of smoking in a person through bio-signals. Given a person's information, build a machine learning model that can classify the presence or absence of smoking

Insights from visualizations

Genderwise Smokers

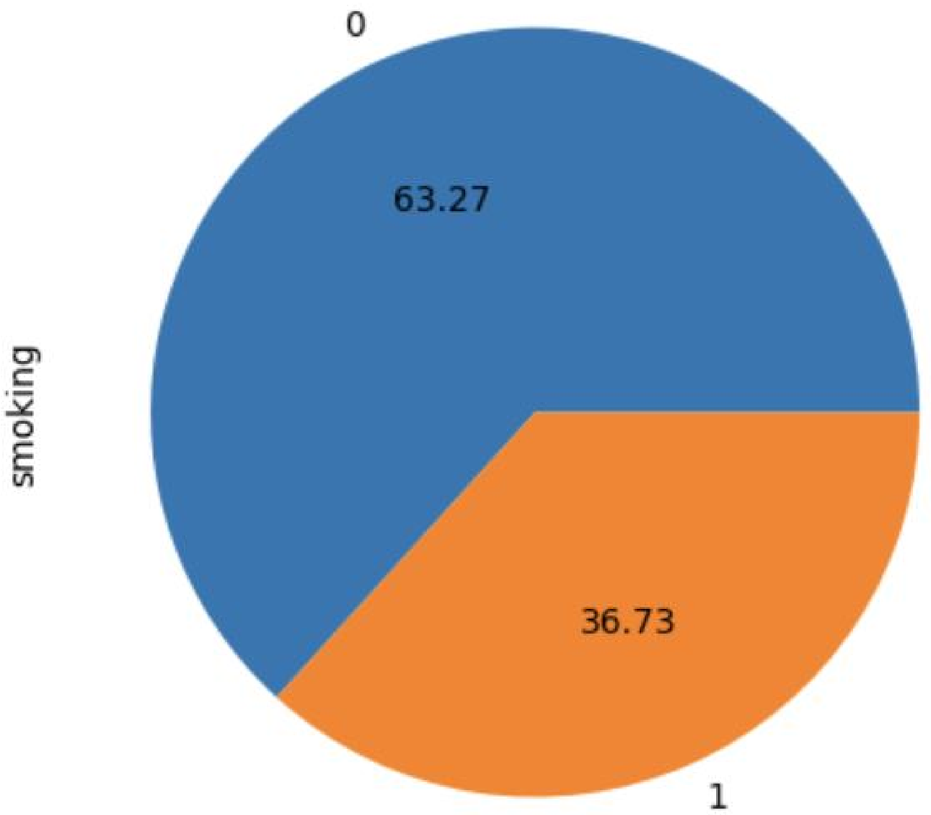




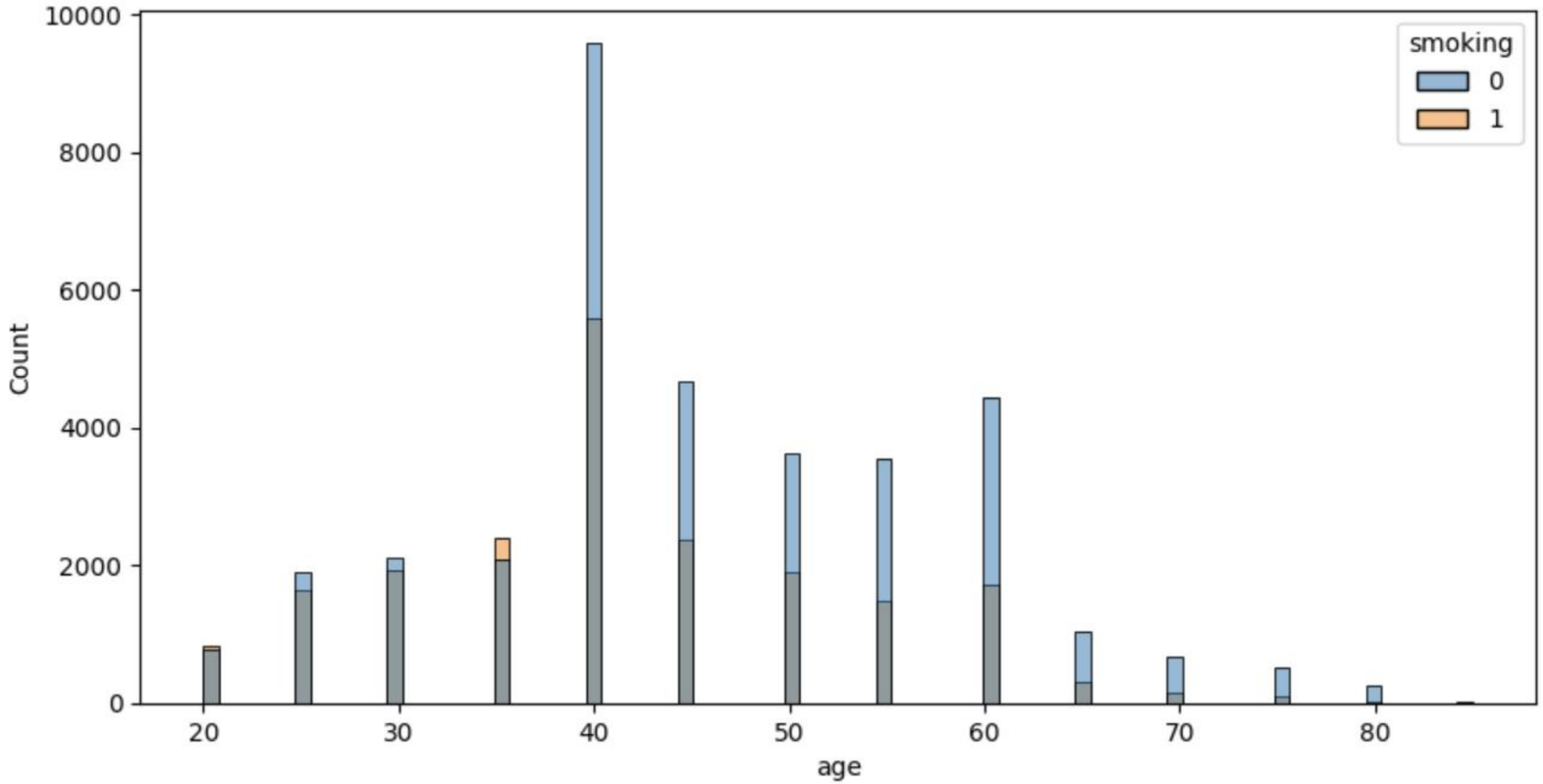
- We can clearly see from the below graph that most smokers are men

Total Smokers

- There are 36.73 percent of the people who are smoking ciggarette.



Smokers - Age



- Most number of smokers are having the age 40

Predictive Analysis

Logistic regression

Logistic Regression

```
x= df[['gender','height(cm)','Gtp','hemoglobin','triglyceride','age','weight(kg)','waist(cm)','serum creatinine','HDL','fasting blood sugar',
       'relaxation','systolic','ALT','LDL']]
y= df['smoking']
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,random_state=42)
from sklearn.preprocessing import StandardScaler
st=StandardScaler()
x_train=st.fit_transform(x_train)
x_test=st.transform(x_test)
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(x_train,y_train)
y_pred=lr.predict(x_test)
from sklearn.metrics import accuracy_score, classification_report
accuracy_score(y_test,y_pred)

print ('Acuracy score of Logistic Regression is ', accuracy_score(y_test,y_pred))
```

```
Acuracy score of Logistic Regression is  0.7351647365113565
```

- LogisticRegression have an accuracy score of 73.51 percentage

Decision Tree

Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
dt=DecisionTreeClassifier()
dt.fit(x_train,y_train)
y_pred=dt.predict(x_test)
print (classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.83      0.82      0.82      7027
           1       0.70      0.70      0.70      4112

    accuracy                           0.78     11139
   macro avg       0.76      0.76      0.76     11139
weighted avg       0.78      0.78      0.78     11139
```

- Decision Tree have an accuracy score of 78 percentage

Bagging Algorithm – Bagging Classifier

Bagging Algorithm – Bagging Classifier

```
from sklearn.ensemble import BaggingClassifier
bagging_clf=BaggingClassifier(estimator=DecisionTreeClassifier(),n_estimators=1000)
bagging_clf.fit(x_train,y_train).score(x_test,y_test)
y_pred=bagging_clf.predict(x_test)
print (classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.88      0.85      0.86      7027
           1       0.75      0.80      0.77      4112

    accuracy                           0.83     11139
   macro avg       0.81      0.82      0.82     11139
weighted avg       0.83      0.83      0.83     11139
```

- Bagging classifier have an accuracy score of 83 percentage

Bagging Algorithm – Extra Trees

Bagging Algorithm – Extra Trees

```
from sklearn.ensemble import ExtraTreesClassifier
et=ExtraTreesClassifier(n_estimators=1000, random_state=42)
et.fit(x_train,y_train)
y_pred=et.predict(x_test)
print (classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.89      0.84      0.86      7027
           1       0.75      0.82      0.78      4112

    accuracy                           0.83     11139
   macro avg       0.82      0.83      0.82     11139
weighted avg       0.84      0.83      0.83     11139
```

- Extra Trees have an accuracy score of 83 percentage

Bagging Algorithm – Random Forest

Bagging Algorithm – Random Forest ¶

```
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier(n_estimators=1000)
rf.fit(x_train,y_train)
y_pred=rf.predict(x_test)
print (classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.88      0.85      0.86      7027
           1       0.75      0.80      0.78      4112

    accuracy                           0.83     11139
   macro avg       0.82      0.82      0.82     11139
weighted avg       0.83      0.83      0.83     11139
```

- Random forest have an accuracy score of 83 percentage

Thanks

Thank you everyone who been with me with this journey.
Check the github repository for more info about the project