

# *Project Title:*

Analysis and Predicting iPhone purchases based on Gender,age,and salary

Organization name:Apple

**Name:K.Pranavi**

**Roll No:20X01A0584**

**Year: IV**

Project Guide Name:

Mr. Lokesh B Sir

# Problem Statement

In the era of smartphones, understanding the factors that influence the decision to purchase an iPhone is crucial for businesses and marketers. This project aims to create a predictive model that determines the likelihood of an individual purchasing an iPhone based on their gender, age, and salary. The dataset consists of individuals' demographic information (gender and age) and financial status (salary), along with a binary variable indicating whether they purchased an iPhone (1 for "yes" and 0 for "no").

Overall, this project seeks to develop a predictive model that can assist businesses in identifying potential iPhone buyers based on their demographic and financial information, allowing for more targeted marketing strategies and product offerings.

## Project Title:

"Predicting iPhone Purchases based on Gender, Age, and Salary"

## 💡 Decision Tree Classifier Algorithm 💡

### ▾ Import the library's and load the dataset

```
import pandas as pd
dataset=pd.read_csv("Iphone Buyer.csv")
X=dataset.iloc[:, :-1].values
y=dataset.iloc[:, 3].values
```

### ▾ Convert Gender to number by using Label Encoding

```
from sklearn.preprocessing import LabelEncoder
labelEncoder_gender=LabelEncoder()
X[:,0]=labelEncoder_gender.fit_transform(X[:,0])
```

### ▾ Split Data

```
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=0)
```

### ▾ Fit the classifier

```
from sklearn.tree import DecisionTreeClassifier
classifier=DecisionTreeClassifier(criterion="entropy",random_state=0)
classifier.fit(X_train,y_train)
```

```
DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', random_state=0)
```

### ▾ Predict

```
y_pred=classifier.predict(X_test)
y_pred

array([0, 0, 0, 1])
```

### ▾ Evaluate the model performance

```
from sklearn import metrics
cm=metrics.confusion_matrix(y_test,y_pred)
print(cm)
accuracy=metrics.accuracy_score(y_test,y_pred)
print("Accuracy score",accuracy)
precision=metrics.precision_score(y_test,y_pred)
print("Precision score:",precision)
recall=metrics.recall_score(y_test,y_pred)
print("Recall score",recall)
```

```
[[3 0]
 [0 1]]
Accuracy score 1.0
Precision score: 1.0
Recall score 1.0
```

▼ ✨ Result :

- ▼ According to my model,Female has bought highest Iphone as compared to Males

---

✓ 0s completed at 12:17 PM

