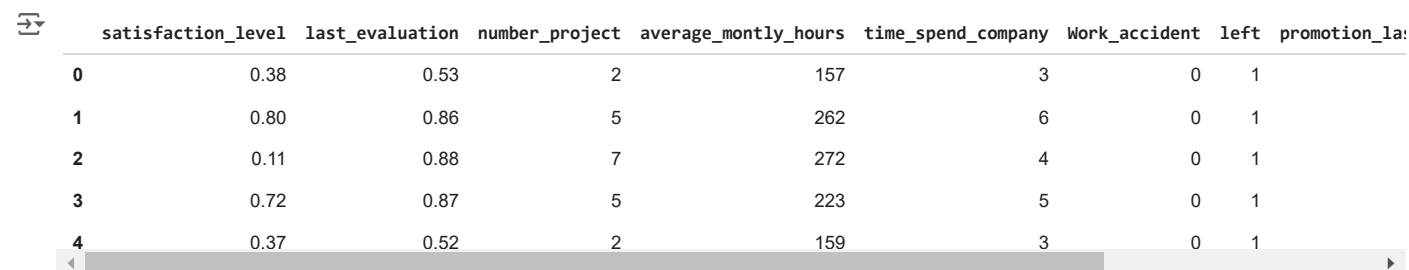


1) Import Libraries

```
import pandas as pd
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report
from sklearn.decomposition import PCA
```

2) Load the dataset

```
data = pd.read_csv('/content/HR_comma_sep.csv')
data.head()
```



| | satisfaction_level | last_evaluation | number_project | average_monthly_hours | time_spend_company | Work_accident | left | promotion_last_year |
|---|--------------------|-----------------|----------------|-----------------------|--------------------|---------------|------|---------------------|
| 0 | 0.38 | 0.53 | 2 | 157 | 3 | 0 | 1 | |
| 1 | 0.80 | 0.86 | 5 | 262 | 6 | 0 | 1 | |
| 2 | 0.11 | 0.88 | 7 | 272 | 4 | 0 | 1 | |
| 3 | 0.72 | 0.87 | 5 | 223 | 5 | 0 | 1 | |
| 4 | 0.37 | 0.52 | 2 | 159 | 3 | 0 | 1 | |

Next steps:

[Generate code with data](#)[View recommended plots](#)[New interactive sheet](#)

3) Preprocess the data

```
le_department = LabelEncoder()
le_salary = LabelEncoder()
data['Department'] = le_department.fit_transform(data['Department'])
data['salary'] = le_salary.fit_transform(data['salary'])
```

4) Separate features and target variable

```
X = data.drop('left', axis=1)
y = data['left']

# Scale the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Apply PCA to reduce the dataset dimensions while retaining 95% of variance
pca = PCA(n_components=0.95)
X_pca = pca.fit_transform(X_scaled)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, random_state=42)
X_train_pca, X_test_pca, y_train, y_test = train_test_split(X_pca, y, test_size=0.3, random_state=42)
```

5) Train and evaluate Naïve Bayes on the full dataset

```
nb_classifier = GaussianNB()
nb_classifier.fit(X_train, y_train)
y_pred_nb = nb_classifier.predict(X_test)

# Train and evaluate SVM on the full dataset
svm_classifier = SVC(kernel='linear', random_state=42)
svm_classifier.fit(X_train, y_train)
y_pred_svm = svm_classifier.predict(X_test)

# Train and evaluate Naïve Bayes on the PCA-reduced dataset
nb_classifier_pca = GaussianNB()
nb_classifier_pca.fit(X_train_pca, y_train)
y_pred_nb_pca = nb_classifier_pca.predict(X_test_pca)

# Train and evaluate SVM on the PCA-reduced dataset
svm_classifier_pca = SVC(kernel='linear', random_state=42)
svm_classifier_pca.fit(X_train_pca, y_train)
y_pred_svm_pca = svm_classifier_pca.predict(X_test_pca)
```

```
print("Naïve Bayes Accuracy (Full Dataset):", accuracy_score(y_test, y_pred_nb))  
print("SVM Accuracy (Full Dataset):", accuracy_score(y_test, y_pred_svm))  
print("\nNaïve Bayes Accuracy (PCA-Reduced Dataset):", accuracy_score(y_test, y_pred_nb_pca))  
print("SVM Accuracy (PCA-Reduced Dataset):", accuracy_score(y_test, y_pred_svm_pca))
```



```
Naïve Bayes Accuracy (Full Dataset): 0.7991111111111111  
SVM Accuracy (Full Dataset): 0.7708888888888888
```

```
Naïve Bayes Accuracy (PCA-Reduced Dataset): 0.8622222222222222  
SVM Accuracy (PCA-Reduced Dataset): 0.7708888888888888
```