REPORT

Abstract:

Employee attrition is a challenge for nearly every organization. Statistics show that about 3 million Americans quit their job each month. There are many important factors such as Monthly Salary, Job Satisfaction, Work-Life Balance that plays a major role in attrition. Though many business owners do not want to face the idea that their own leadership styles may cause their own attrition issues. Here various machine learning algorithms can be applied to predict different factors and their effects on employees leading to attrition.

Introduction:

Reduction in staff and employees in a company through normal means too, such as retirement and resignation, the loss of customers or clients to old age or growing out of the company's target demographic can lead to attrition in business. Study shows that 87% of HR leaders consider improved retention a critical or high priority over the next five years. Different machine learning algorithms will help us understand the current scenario and improve it in the future.

Code of Documentation:

Various machine learning algorithms are applied on the data set. They are as follows:

- **Logistic Regression**

LOGISTIC REGRESSION

```
from sklearn.linear_model import LogisticRegression
log_reg = LogisticRegression()
log_reg.fit(X_train,Y_train)
train_predict = log_reg.predict(X_train)
test_predict = log_reg.predict(X_test)
y_prob = log_reg.predict(train)
y_pred = np.where(y_prob > 0.5, 1, 0)
train_test_error(train_predict , test_predict)
```

```
87.38916256157636 is the train accuracy
88.41158841158841 is the test accuracy
```

- **SGD**

STOCHASTIC GRADIENT DESCENT

```
from sklearn.linear_model import SGDClassifier
sgd = SGDClassifier(loss="hinge", penalty="l2")
sgd.fit(X_train,Y_train)
train_predict = sgd.predict(X_train)
test_predict = sgd.predict(X_test)
train_test_error(train_predict , test_predict)
```

```
74.48275862068967 is the train accuracy
77.12287712287711 is the test accuracy
```

- **Perceptron**

PERCEPTRON

```
from sklearn.linear_model import Perceptron
per = Perceptron(fit_intercept=False, n_iter=10, shuffle=False).fit(X_train,Y_train)
train_predict = per.predict(X_train)
test_predict = per.predict(X_test)
train_test_error(train_predict , test_predict)
```

```
85.66502463054188 is the train accuracy
86.51348651348651 is the test accuracy
```

- **SVM**

SUPPORT VECTOR MACHINE

```
from sklearn import svm
SVM = svm.SVC(probability=True)
SVM.fit(X_train,Y_train)
train_predict = SVM.predict(X_train)
test_predict = SVM.predict(X_test)
train_test_error(train_predict , test_predict)
```

```
100.0 is the train accuracy
92.10789210789211 is the test accuracy
```

- **KNN**

K-NEAREST NEIGHBORS

```
from sklearn import neighbors
n_neighbors = 15
knn = neighbors.KNeighborsClassifier(n_neighbors, weights='distance')
knn.fit(X_train,Y_train)
train_predict = knn.predict(X_train)
test_predict = knn.predict(X_test)
train_test_error(train_predict , test_predict)
```

```
100.0 is the train accuracy
91.20879120879121 is the test accuracy
```

- **Naïve Bayes**

NAIVE BAYES

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train,Y_train)
train_predict = gnb.predict(X_train)
test_predict = gnb.predict(X_test)
train_test_error(train_predict , test_predict)
```

```
85.86206896551725 is the train accuracy
87.41258741258741 is the test accuracy
```

- **Decision Tree**

DECISION TREE

```
from sklearn import tree
dec = tree.DecisionTreeClassifier()
dec.fit(X_train,Y_train)
train_predict = dec.predict(X_train)
test_predict = dec.predict(X_test)
train_test_error(train_predict , test_predict)
```

```
100.0 is the train accuracy
89.41058941058941 is the test accuracy
```

- **K-Means**

K-MEANS CLUSTERING

```
from sklearn.cluster import KMeans
kms = KMeans(n_clusters=2, random_state=1)
kms.fit(X_train,Y_train)
train_predict = kms.predict(X_train)
test_predict = kms.predict(X_test)
train_test_error(train_predict,test_predict)
```

```
58.423645320197046 is the train accuracy
61.33866133866134 is the test accuracy
```

Discussion:

Different algorithms gave different results on the basis of attrition value in the data set. A training data was passed in each of the algorithm to check the accuracy. The difference between the train accuracy and test accuracy varied from algorithm to algorithm.

Large differences were seen in the test accuracy of algorithms like SVM, KNN and Decision tree compared to that of train accuracy.

Whereas algorithms like SGD, Perceptron, Naïve Bayes and K-means clustering showed minute differences between the train and the test accuracies.

Results: All the above performed algorithms were compared to find out which gives the best accuracy.

	Test Accuracy	Train Accuracy
Logistic Regression	88.411588	87.389163
SGD	77.122877	74.482759
Perceptron	86.513487	85.665025
SVM	92.107892	100.000000
KNN	91.208791	100.000000
GaussianNB	87.412587	85.862069
Decision Tree	89.410589	100.000000
K Means Clustering	61.338661	58.423645

This shows that Logistic Regression gives the highest test accuracy when the binary outcome of Attrition, whether Yes or No is to be predicted. It is the best fit when we want to predict the probability of occurrence of an event by fitting data into a logit function.

References:

https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset

https://blog.bonus.ly/10-surprising-employee-retention-statistics-you-need-to-know