Prediction of Insurance Claim with Machine Learning

What is Machine Learning?

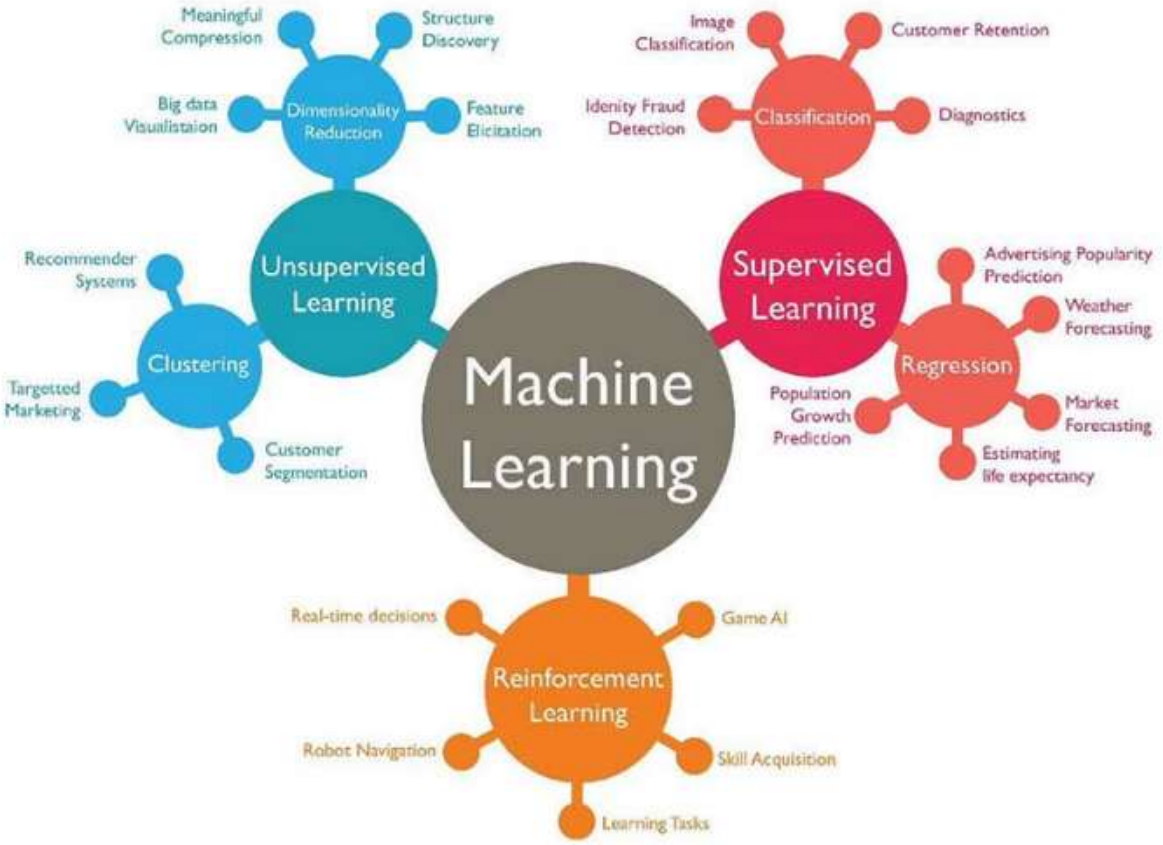
Arthur Samuel, an expert of artificial Intelligence and computer gaming, once stated about Machine Learning as “Field of study that gives computers the capability to learn without being explicitly programmed”.

In another very simple language Machine Learning is automating and improving the learning process of computers based on their experiences without being actually programmed i.e. without any human assistance. The process starts with feeding good quality data and then training the machines by building machine learning models using the data and different algorithms. The choice of algorithms depends on what type of data we have and kind of task we are trying to automate.

Type of Machine Learning?

It is classified into 2 types, which is as followed:

1. Supervised Learning:
2. Unsupervised Learning
3. Reinforcement Learning

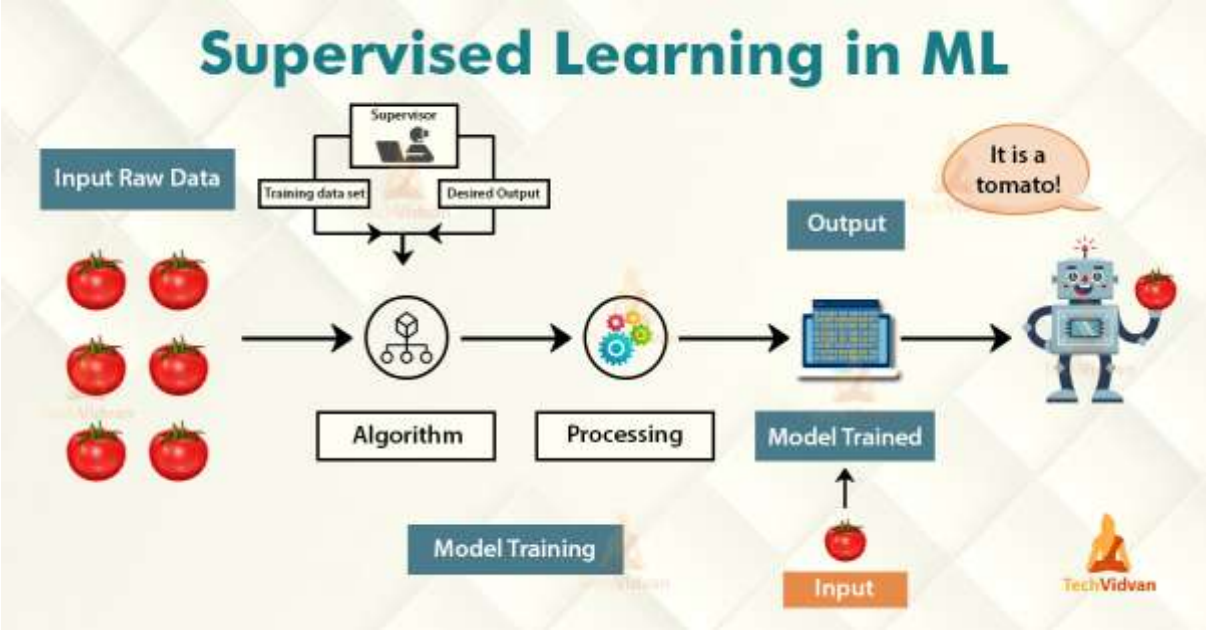


Overview of Supervised Learning Algorithm

In supervised Learning, Machine is trained by using Labelled data. Our dataset having both target and independent variable are called labelled. We can understand this more clearly by an example, as in class we have teacher to supervised likewise in this we have one training data to learn and understand the dataset and then as per learning of training data we try to predict the output of test data. This algorithm is very much useful to solve various real world problems.

Type of supervised Learning

1. **Regression**: In this we have problems which are continuous output data or numerical output. For example: age, weight, temperature, salary etc.
2. **Classification**: In this we have problems which are categorical output data or character output. For example: weather, class, colour etc.



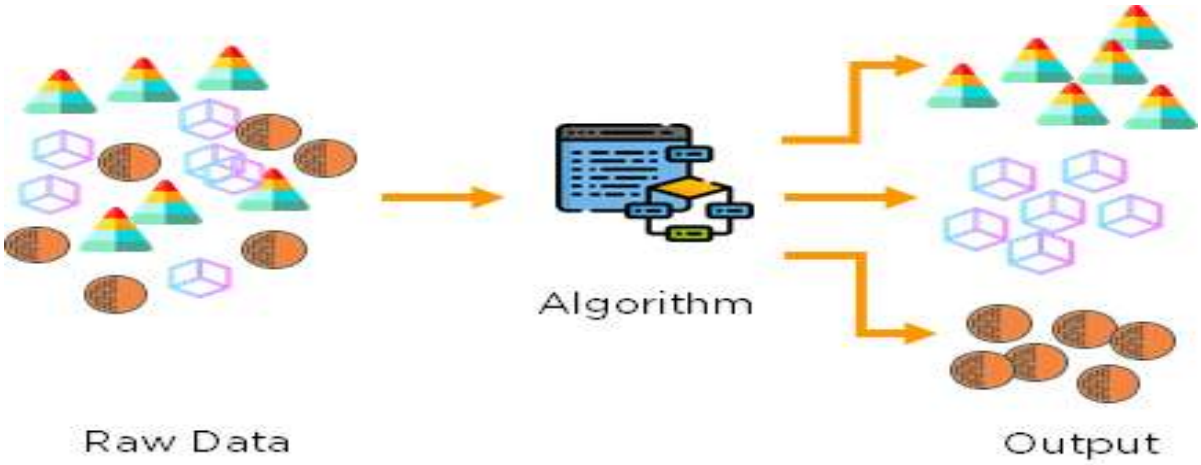
Overview of Unsupervised Learning Algorithm

Unsupervised learning does not have labels to work off of, resulting in the creation of hidden structures. Relationships between data points are perceived by the algorithm in an abstract manner, with no input required from human beings. The creation of these hidden structures is what makes unsupervised learning algorithms versatile. Instead of a defined and set problem statement, unsupervised learning algorithms can adapt to the data by dynamically changing hidden structures. This offers more post-deployment development than supervised learning algorithms.

Type of unsupervised Learning

1. **Clustering**: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behaviour.

2. **Association**: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.



Overview of reinforcement learning Algorithm

It features an algorithm that improves upon itself and learns from new situations using a trial-and-error method. Favourable outputs are encouraged or 'reinforced', and non-favourable outputs are discouraged or 'punished'. We can understand reinforcement learning more clearly by an example as finding the shortest route between two points on a map, the solution is not an absolute value. Instead, it takes on a score of effectiveness, expressed in a percentage value. The higher this percentage value is, the more reward is given to the algorithm. Thus, the program is trained to give the best possible solution for the best possible reward.

Introduction of our Problem

Insurance fraud is a huge problem in the industry. It's difficult to identify fraud claims. Machine Learning is in a unique position to help the Auto Insurance industry with this problem.

In this project, we are provided a dataset which has the details of the insurance policy along with the customer details. It also has the details of the accident on the basis of which the claims have been made.
In this example, we will be working with some auto insurance data to demonstrate how we can create a predictive model that predicts if an insurance claim is fraudulent or not.

Dataset:
https://github.com/dsrscientist/Data-Science-ML-Capstone-Projects/blob/master/Automobile_insurance_fraud.csv

First, we must import all the important libraries which we are going to use for pre-processing of data

```
#Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

After importing the libraries, we will load the dataset on which we have to work on (Load data from above link given as dataset)

```
#Importing Dataset
df= pd.read_csv('Automobile_insurance_fraud.csv')
```

```
#Checking Dataset
df.head()
```

From here the job of data scientist starts. Now we start working on dataset. We check the features present in our data and then we will look at their datatypes.

```
#Checking Dimension of Dataframe
df.ndim
```

```
#Checking Shape of Dataframe
df.shape
```

```
#Checking Datatypes of columns of Dataframe
df.dtypes
```

Then we can check different values total counts in column. And also check the null values or missing value.

```
#Checking Nullvalues in the Dataframe
df.isnull().sum()
```

```
#Checking different values total counts in the column
df['police_report_available'].value_counts()
```

```
#Checking different values total counts in the column
df['police_report_available'].value_counts()
```

```
#Replacing value in the column
df['police_report_available']=df['police_report_available'].replace('?','NO')
```

```
#Checking different values total counts in the column
df['fraud_reported'].value_counts()
```

```
#Checking different values total counts in the column
df['insured_relationship'].value_counts()
```

```
#Checking different values total counts in the column
df['insured_sex'].value_counts()
```

```
#Droping unnecessary columns
df=df.drop(['policy_bind_date','auto_year','_c39','incident_location','insured_sex','incident_city','incident_hour_of_t
```

After checking this much we can now plot histogram, bar graph and many more for exploratory data analysis. Plotting of Histogram graph

is known as univariate analysis and plotting of Bar Graph, Scatter Graph etc are called bivariate analysis.

EDA

```
# Histogram Plot
plt.hist(df['fraud_reported'])
```

```
#Scattered Plot
plt.figure(figsize=[22,12])
plt.scatter(df['age'],df['months_as_customer'])
```

```
#Bar Graph
plt.figure(figsize=[22,12])
plt.bar(df['auto_make'],df['age'])
```

If our dataset has string values and sometimes it is important to convert these string data into numeric values to pass the data from Machine Learning models. In this project, we have many columns having string values which need to convert to numeric values. This process of conversion is known as Label Encoding.

Label Encoding

Now we need to convert string values to numeric values so that our model can read the dataframe.

```
#Label Encoding
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
df['Partner']=le.fit_transform(df['Partner'])
df['Dependents']=le.fit_transform(df['Dependents'])
df['PhoneService']=le.fit_transform(df['PhoneService'])
df['MultipleLines']=le.fit_transform(df['MultipleLines'])
df['InternetService']=le.fit_transform(df['InternetService'])
df['OnlineSecurity']=le.fit_transform(df['OnlineSecurity'])
df['OnlineBackup']=le.fit_transform(df['OnlineBackup'])
df['DeviceProtection']=le.fit_transform(df['DeviceProtection'])
df['TechSupport']=le.fit_transform(df['TechSupport'])
df['StreamingTV']=le.fit_transform(df['StreamingTV'])
df['StreamingMovies']=le.fit_transform(df['StreamingMovies'])
df['Contract']=le.fit_transform(df['Contract'])
df['PaperlessBilling']=le.fit_transform(df['PaperlessBilling'])
df['PaymentMethod']=le.fit_transform(df['PaymentMethod'])
df['Churn']=le.fit_transform(df['Churn'])
```

Now we should check correlation between each column, and it is also called Multivariant Analysis. Correlation of any column with itself is always 1.

Correlation

```
#Checking Correlation
corr=df.corr()
plt.figure(figsize=[22,12])
sns.heatmap(corr,annot=True)
plt.show()
```

```
#Checking Correlation with respect to target column
corr['fraud_reported'].sort_values()
```

Now, we can check the outliers present in our dataset by plotting boxplot graphs of each column.

```
#Checking outliers with Boxplot Graph
df.iloc[:,0:14].boxplot(figsize=[20,8])
```

```
#Checking outliers with Boxplot Graph
df.iloc[:,14::].boxplot(figsize=[20,8])
```

```
#Improving Outliers
from scipy import stats
z=np.abs(stats.zscore(df))
df_new=(z<2).all(axis=1)
df=df[df_new]
```

After this much of analysis, we will split our dataset into independent variables X and target variable Y.

```
#X,Y Split
X=df.drop('fraud_reported',axis=1)
Y=df['fraud_reported']
```

Now we check the skewness of independent variables X. If skewness of each column is between -0.5 to 0.5 then we can take our data forward, otherwise we need to improve that by using any one method among power transformation, log transformation and square root transformation. In our case, we have used Power Transformation.

```
# Checking Skewness
X.skew()
```

```
#Improving Skewness
from sklearn.preprocessing import power_transform
pt=power_transform(X, method='yeo-johnson')
X=pd.DataFrame(pt, columns=X.columns)
X.skew()
```

Now we will import all the modules which we are going to use in models.

```
#Importing Libraries
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
from sklearn.metrics import r2_score
```

For getting maximum accuracy we will need to find the best random state. In our case best random state we get is 8.

```
#Finding Best Random State
maxAcc=0
maxRS=0
for i in range(1,100):
    X_train, X_test, Y_train, Y_test=train_test_split(X,Y,random_state=i, test_size=0.3)
    LR=LogisticRegression()
    LR.fit(X_train,Y_train)
    pred=LR.predict(X_test)
    acc=accuracy_score(Y_test,pred)
    if acc>maxAcc:
        maxAcc=acc
        maxRS=i
        print('Best accuracy is', maxAcc,'on Random_state', maxRS)
```

We will use train test split method to predict target values by calling different ML algorithms here we are calling only 4 algorithms, that is Logistic Regression, DecisionTree classifier, RandomForestClassifier and SVC.

Train-Test Split

```
#Train-Test Split
X_train, X_test, Y_train, Y_test=train_test_split(X,Y,random_state=8, test_size=0.3)
```

ML Algorithms

```
#Logistic Regression
lr=LogisticRegression()
lr.fit(X_train,Y_train)
predlr=lr.predict(X_test)
print('Accuracy Score :', accuracy_score(Y_test,predlr))
print('Confusion Matrix :', confusion_matrix(Y_test,predlr))
print('Classification Report :', classification_report(Y_test,predlr))
```

```
#Decision Tree Classifier
dtc=DecisionTreeClassifier()
dtc.fit(X_train,Y_train)
preddtc=dtc.predict(X_test)
print('Accuracy Score :', accuracy_score(Y_test,preddtc))
print('Confusion Matrix :', confusion_matrix(Y_test,preddtc))
print('Classification Report :', classification_report(Y_test,preddtc))
```

```
#Random Forest Classifier
rfc=RandomForestClassifier()
rfc.fit(X_train,Y_train)
predrfc=rfc.predict(X_test)
print('Accuracy Score :', accuracy_score(Y_test,predrfc))
print('Confusion Matrix :', confusion_matrix(Y_test,predrfc))
print('Classification Report :', classification_report(Y_test,predrfc))
```

```
#SVC Model
svc=SVC()
svc.fit(X_train,Y_train)
predsvc=svc.predict(X_test)
print('Accuracy Score :', accuracy_score(Y_test,predsvc))
print('Confusion Matrix :', confusion_matrix(Y_test,predsvc))
print('Classification Report :', classification_report(Y_test,predsvc))
```

After predicting the target values we have to check the overfitting, underfitting of actual data and the predicted data. For this we did the cross validation of all algorithm. And differentiate the accuracy of actual value and the accuracy found during checking cross validation. Algorithm have the minimum value will be expected to find algorithm for predict the test data.

Cross Validation Score

```
#Logistic Regression
from sklearn.model_selection import cross_val_score
scr1=cross_val_score(lr,X,Y,cv=5)
print('Cross Validation Score:', scr1.mean())
```

```
#Decision Tree Classifier
scr2=cross_val_score(dtc,X,Y,cv=5)
print('Cross Validation Score:', scr2.mean())
```

```
#Random Forest Classifier
scr3=cross_val_score(rfc,X,Y,cv=5)
print('Cross Validation Score:', scr3.mean())
```

```
#SVC
scr4=cross_val_score(svc,X,Y,cv=5)
print('Cross Validation Score:', scr4.mean())
```

We will save the best ML algorithm for future use.

```
#Saving Model
import joblib
joblib.dump(gcv.best_estimator_,'Automobile_Insurance_Fraud_Prediction.pkl')
```

Conclusion

From the exploratory Data analysis, we could generate insight from the data. How each of the features relates to the target. Also, it can be seen from the evaluation of 4 models that RandomForestClassifier performed better than other models.

Summary

In this blog, I have presented you a modern Data science problem with the basics concepts of Machine Learning and I hope this blog was helpful and would have motivated you enough to get interested in the topic.