

```
In [ ]: ▶ # Importing the Library
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, mean_squared_error, log_loss
from tqdm import tqdm_notebook
import seaborn as sns
import imageio
import time
from IPython.display import HTML

#from sklearn.preprocessing import OneHotEncoder
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import make_blobs

In [ ]: ▶ #data = pd.read_csv('Final_Refined_OneHotEncoded.csv')
data = pd.read_csv('Final_Refined_Encoded_Normalysed.csv')

In [ ]: ▶ data.head(20)

In [ ]: ▶ # Saperating the Target Column
X, y = data.iloc[:, :-1], data.iloc[:, -1]

In [ ]: ▶ # Test-Train Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, rand

In [ ]: ▶ # Standard-Scalar
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

In [ ]: ▶ # Training Function
def train_model(X_train, y_train, X_test, y_test, classifier, **kwargs):
    model = classifier(**kwargs)
    model.fit(X_train, y_train)

    # check accuracy and print out the results
    fit_accuracy = model.score(X_train, y_train)
    test_accuracy = model.score(X_test, y_test)

    print(f"Train accuracy: {fit_accuracy:0.2%}")
    print(f"Test accuracy: {test_accuracy:0.2%}")

    return model
```

```
In [ ]: ▶ # Random Forests
        model = train_model(X_train, y_train, X_test, y_test, RandomForestClassifier,
                             pd.Series(model.feature_importances_, X.columns).sort_values(ascending=False))
```

```
In [ ]: ▶
```