LAB PRACTICE ASSIGNMENT – 1 (ML)

Name – Ankur Kaushal
BATCH – CSE4

Build predictive model with Logistic Regression to predict the IRIS flower category from the feature measures of petal and sepal.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import OneHotEncoder
df = pd.read_csv("iris.csv",
                 header=None,
                 names=["sepal_length", "sepal_width",
"petal_length", "petal_width", "species"])
X = df.drop("species", axis=1)
y = df["species"]
encoder = OneHotEncoder()
y = encoder.fit_transform(y.values.reshape(-1, 1))
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)
model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
acc = accuracy_score(y_test, y_pred)
print("Accuracy: ", acc)
```

Build predictive model with Logistic Regression to predict whether income of a person exceeds $50K/yr based on census data.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
df = pd.read_csv("adult.csv",
                 header=None,
                 names=["age", "workclass", "fnlwgt", "education",
"education-num", "marital-status", "occupation",
"relationship", "race", "sex", "capital-gain", "capital-loss",
"hours-per-week", "native-country", "class"])
df = pd.get_dummies(df, columns=["workclass", "education",
"marital-status", "occupation", "relationship", "race", "sex",
"native-country"])
X_train, X_test, y_train, y_test =
train_test_split(df.drop("class", axis=1), df["class"],
test_size=0.2)
model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
acc = accuracy_score(y_test, y_pred)
print("Accuracy: ", acc)
```

Build predictive models to predict the age of abalone from physical measurements.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error
df = pd.read_csv("abalone.csv",
                 header=None,
                 names=["sex", "length", "diameter", "height",
"whole_weight", "shucked_weight", "viscera_weight",
"shell_weight", "rings"])
X_train, X_test, y_train, y_test =
train_test_split(df.drop("rings", axis=1), df["rings"],
test_size=0.2)
model = RandomForestRegressor()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
mae = mean_absolute_error(y_test, y_pred)
print("Mean Absolute Error: ", mae)
```