```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import mean_squared_error, r2_score

data= pd.read_csv('/content/drive/MyDrive/files2/TSLA.csv')

xv_train = data[['High','Low']]
y_train = data['Volume']

RFC = RandomForestClassifier(random_state=0)
RFC.fit(xv_train,y_train)
```

```
RandomForestClassifier(random_state=0)
```

```
LR = LogisticRegression(max_iter=1000)
LR.fit(xv_train,y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/
_logistic.py:458: ConvergenceWarning: lbfgs failed to converge
(status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as
shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

```
LogisticRegression(max_iter=1000)
```

```
DTC = DecisionTreeClassifier()
DTC.fit(xv_train,y_train)
```

```
DecisionTreeClassifier()
```

```
k=3
knn = KNeighborsClassifier(n_neighbors=k)
knn.fit(xv_train,y_train)
```

```
KNeighborsClassifier(n_neighbors=3)
```

```
lr = LinearRegression()

lr.fit(xv_train,y_train)
```

```
LinearRegression()
```

```
# Assuming 'data' is already defined
# xv_train = data[['Height','Weight']]
# y_train = data['Index']

# Example input (replace with actual input)
high = float(input("Enter high (): "))
low = float(input("Enter low (): "))

# Create a DataFrame from user input, use 'Low' instead of 'low' for
consistency
user_data = pd.DataFrame({'High': [high], 'Low': [low]}) # Fixed typo
here

# Select only the relevant features for prediction
user_data_for_prediction = user_data[['High', 'Low']]  # Select the
same features used in training

# Get predictions from each model
rfc_pred = RFC.predict(user_data_for_prediction)  # Use the DataFrame
with correct features
lr_pred = LR.predict(user_data_for_prediction)
dtc_pred = DTC.predict(user_data_for_prediction)
knn_pred = knn.predict(user_data_for_prediction)
linear_pred = lr.predict(user_data_for_prediction)

# Create a DataFrame from the predictions
predictions_df = pd.DataFrame({
    'RandomForest': rfc_pred,
    'LogisticRegression': lr_pred,
    'DecisionTree': dtc_pred,
    'KNeighbors': knn_pred,
    'LinearRegression': linear_pred
})

# Merge predictions with input data
merged_output = pd.concat([user_data, predictions_df], axis=1)
print(merged_output)
```

```
Enter high (): 5
Enter low (): 3.5
   High  Low  RandomForest  LogisticRegression  DecisionTree
KNeighbors  \
0   5.0  3.5      93831500           150977500      93831500
25699000

   LinearRegression
0      3.173082e+07
```