

Finding accuracy score for linear regression and multi linear regression

```
In [ ]: # Import libraries
import joblib
import numpy as np
import pandas as pd
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
```

Import Dataset

```
In [ ]: # Load data
df = pd.read_csv('../datasets/mldata2.csv')
df.head()
```

Out []:

	age	height	weight	gender	likeness
0	27	170.688	76.0	Male	Biryani
1	41	165.000	70.0	Male	Biryani
2	29	171.000	80.0	Male	Biryani
3	27	173.000	102.0	Male	Biryani
4	29	164.000	67.0	Male	Biryani

```
In [ ]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 245 entries, 0 to 244
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ---
 0    age        245 non-null    int64
 1   height     245 non-null    float64
 2    weight    245 non-null    float64
 3    gender     245 non-null    object
 4   likeness  245 non-null    object
dtypes: float64(2), int64(1), object(2)
memory usage: 9.7+ KB
```

```
In [ ]: # Convert 'gender' data type to int
df['gender'] = df['gender'].replace('Male', 1)
df['gender'] = df['gender'].replace('Female', 0)
```

```
In [ ]: # Check unique values in 'gender'
df['gender'].unique()
```

```
Out [ ]: array([1, 0], dtype=int64)
```

Simple Linear Regression

```
In [ ]: # Split data into input (X) and output (y)
X = df[['age', 'height', 'weight', 'gender']]
y = df['likeness']
```

```
In [ ]: X.head()
```

Out []:

	age	height	weight	gender
0	27	170.688	76.0	1
1	41	165.000	70.0	1
2	29	171.000	80.0	1
3	27	173.000	102.0	1
4	29	164.000	67.0	1

```
In [ ]: y.head()
```

```
Out [ ]: 0    Biryani
1    Biryani
2    Biryani
3    Biryani
4    Biryani
Name: likeness, dtype: object
```

```
In [ ]: # Split data into test and train (80/20)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

```
In [ ]: # Create a model
model = DecisionTreeClassifier()
# Fit the model
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Check score
score = accuracy_score(y_test, y_pred)
score
```

```
Out [ ]: 0.40816326530612246
```

```
In [ ]: # Save model (extension joblib)
joblib.dump(model, 'dt_foodie.joblib')
```

```
Out [ ]: ['dt_foodie.joblib']
```

```
In [ ]: # Load the model
loaded_model = joblib.load('dt_foodie.joblib')

# Make predictions on loaded model
loaded_pred = loaded_model.predict(X_test)

# Check loaded model score
loaded_score = accuracy_score(y_test, loaded_pred)
loaded_score
```

```
Out [ ]: 0.40816326530612246
```

Both scores are same, it means the model is saved properly.