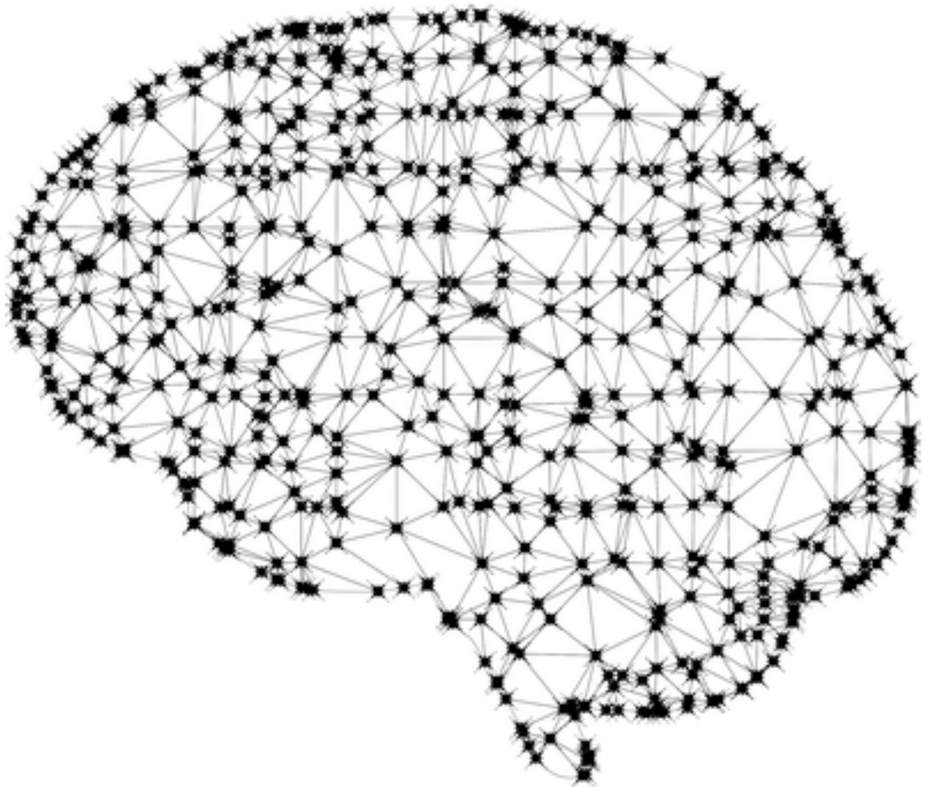
WEB SCRAPING JOB POSTINGS

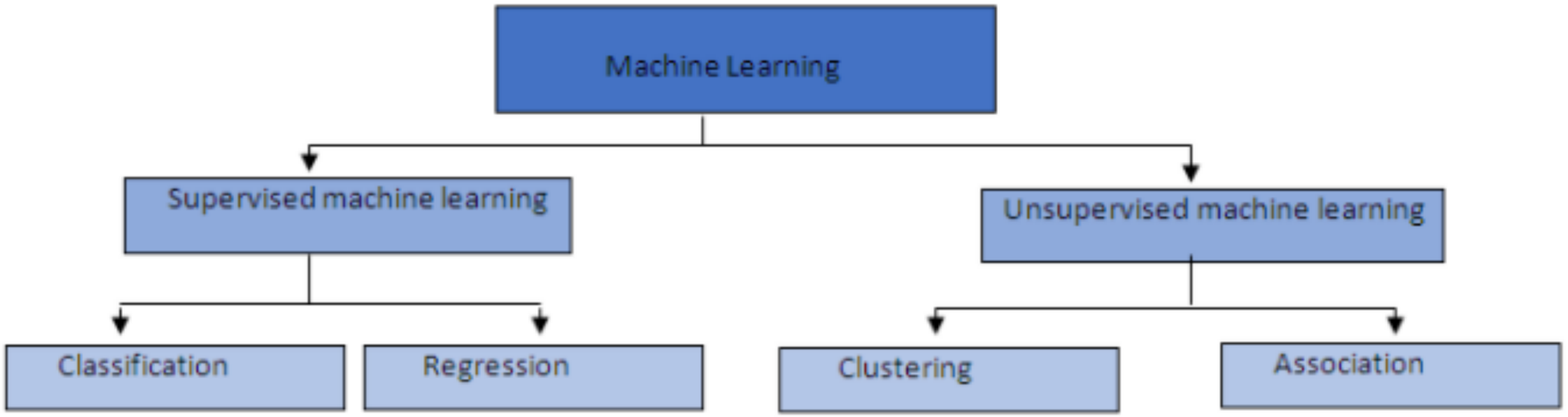
Regression Versus Classification Machine Learning: What's the Difference?



The difference between regression machine learning algorithms and classification machine learning algorithms confuse me

and I say that “understanding whether the machine learning task is a regression or classification problem is key for selecting the right algorithm to use.”

Here is a chart that shows the different groupings of machine learning:



Unfortunately, there is where the similarity between regression versus classification machine learning ends.

The main difference between them is that the output variable in regression is numerical (or continuous) while that for classification is categorical (or discrete).

Regression in machine learning

In machine learning, regression algorithms attempt to estimate the mapping function (f) from the input variables (x) to numerical or continuous output variables (y)

In this case, y is a real value, which can be an integer or a floating point value.

Here is an example of a regression problem in Project 4:

```
#define X and y
#y is a real value, which can be an integer or a floating point value

X = df_data[['title','classification', 'location']]
y = df_data['estimated_salary']
```

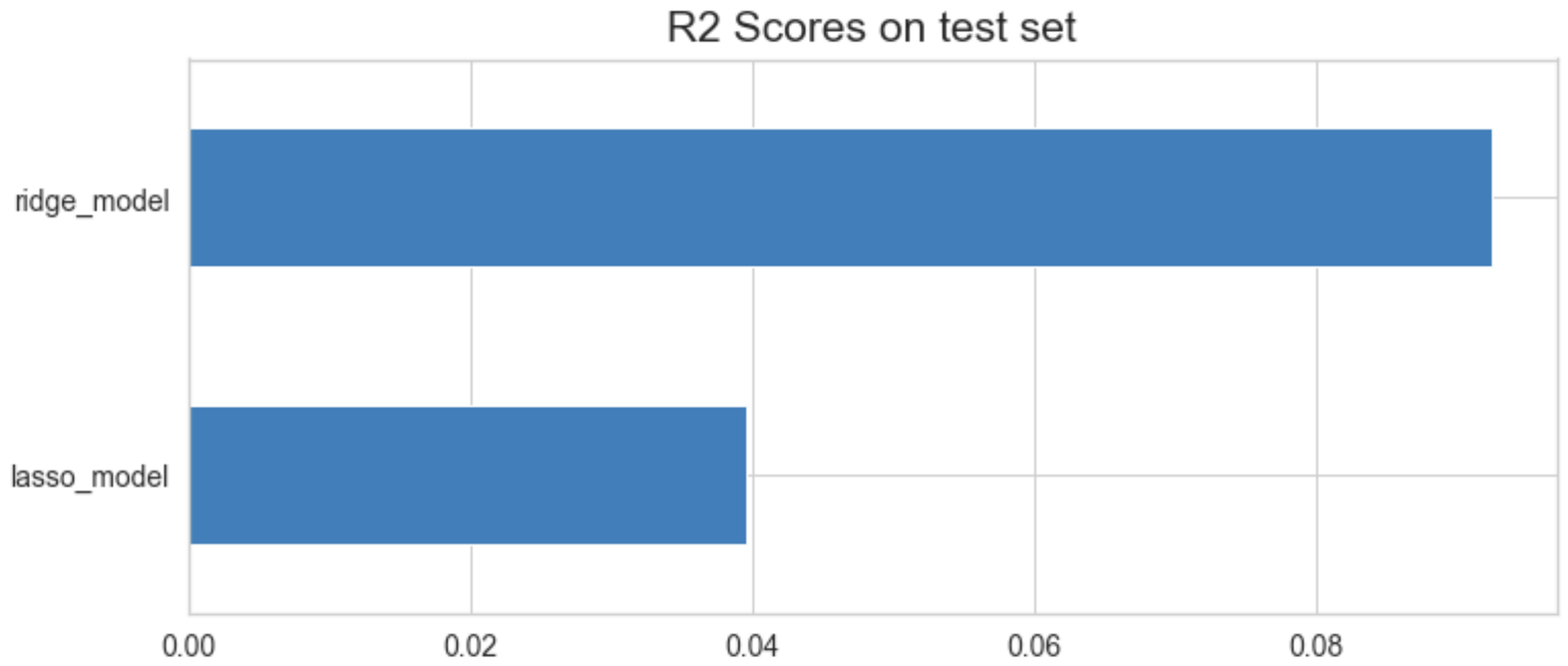
```
X_dummy = pd.get_dummies(X, drop_first=True)
X_dummy.shape
```

```
(885, 782)
```

```
#we frame this as a regression problem first to predict the average_salary
#spilt to train test split
X_train, X_test, y_train, y_test = train_test_split(X_dummy, y, test_size=0.3, random_state=42)
```

The Ridge and Lasso Scores

	test_mean_score	test_std_score
ridge_model	0.092427	0.086914
lasso_model	0.039605	0.168630



Classification in machine learning

On the other hand, classification algorithms attempt to estimate the mapping function (f) from the input variables (x) to discrete or categorical output variables (y).

In this case, y is a category that the mapping function predicts.

Here is an example of a classification problem in Project 4:

```
median_salary = df_data['estimated_salary'].median()
```

```
#y is a category that the mapping function predicts
#create lables from salary

df_data['salary_label'] = df_data['estimated_salary'].apply(lambda x: 1 if x > median_salary else 0)
```

```
# define X and y
#y is a category that the mapping function predicts

X_sal = df_data[['title','classification', 'location']]
y_sal = df_data['salary_label']
```

```
X_sal_dummy = pd.get_dummies(X_sal, drop_first=True)
print (X_sal_dummy.shape)
```

```
(885, 782)
```

Ensamble Methodes

Baseline Accuracy

```
max(df_data.salary_label.value_counts(normalize=True))*100
```

```
53.333333333333336
```

Data Preparation

```
X_train, X_test, y_train, y_test = train_test_split(X_sal_dummy, y_sal, test_size=0.33, random_state=42, stratify = y
```

```
y_test.mean()
```

```
0.46757679180887374
```

```
y_train.mean()
```

```
0.46621621621621623
```

Decision Tree Model

```
dt = DecisionTreeClassifier()
```

```
cross_val_score(dt, X_train, y_train, cv=10)
```

```
array([0.58333333, 0.68333333, 0.61666667, 0.5       , 0.63333333,
       0.56666667, 0.68965517, 0.77586207, 0.60344828, 0.65517241])
```

```
dt.fit(X_train, y_train)
```

```
dt.score(X_train, y_train)
```

```
0.9915540540540541
```

```
dt.score(X_test, y_test)
```

```
0.6109215017064846
```

Bootstrap with Pandas

```
X_sample = X_train.sample(replace=True, n=X_train.shape[0], random_state=42)
y_sample = y_train[X_sample.index]
```

```
bt_tree = DecisionTreeClassifier()
bt_tree.fit(X_sample, y_sample)
bt_tree.score(X_test, y_test)
```

```
0.6621160409556314
```

Bagging Classifier

```
bag = BaggingClassifier(n_estimators=1000)

bag.fit(X_train, y_train)
bag.score(X_test, y_test)
```

```
0.6791808873720137
```

Random ForestClassification

```
# Run RandomForestClassifier
from sklearn.metrics import confusion_matrix, recall_score, accuracy_score

rfc = RandomForestClassifier(random_state=1000)
rfc.fit(X_train, y_train)
rfc_predict = rfc.predict(X_test)
accuracy_score(y_test, rfc_predict)
```

```
/opt/anaconda3/lib/python3.7/site-packages/sklearn/ensemble/forest.py:245: FutureWarning: The default value of n_esti
mators will change from 10 in version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

```
0.6313993174061433
```

Thank 😊 You