

Portfolio assignment 16

30 min: Train a decision tree to predict one of the categorical columns of your own dataset.

- Split your dataset into a train (70%) and test (30%) set.
- Use the train set to fit a DecisionTreeClassifier. You are free to choose which columns you want to use as feature variables and you are also free to choose the max_depth of the tree.
- Use your decision tree model to make predictions for both the train and test set.
- Calculate the accuracy for both the train set predictions and test set predictions.
- Is the accuracy different? Did you expect this difference?
- Use the plot_tree function above to create a plot of the decision tree. Take a few minutes to analyse the decision tree. Do you understand the tree?



In [1]:

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
import pandas as pd
from sklearn import tree
import graphviz

def plot_tree_classification(model, features, class_names):
    # Generate plot data
    dot_data = tree.export_graphviz(model, out_file=None,
                                    feature_names=features,
                                    class_names=class_names,
                                    filled=True, rounded=True,
                                    special_characters=True)

    # Turn into graph using graphviz
    graph = graphviz.Source(dot_data)

    # Write out a pdf
    graph.render("decision_tree")

    # Display in the notebook
    return graph
```

In [2]:

```
pokemons = pd.read_csv('../Pokemon.csv')
pokemons.fillna(value= 0)
```

Out[2]:

	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Gener
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	
4	4	Charmander	Fire	0	309	39	52	43	60	50	65	
...	
795	719	Diancie	Rock	Fairy	600	50	100	150	100	150	50	
796	719	DiancieMega Diancie	Rock	Fairy	700	50	160	110	160	110	110	
797	720	HoopaHoopa Confined	Psychic	Ghost	600	80	110	60	150	130	70	
798	720	HoopaHoopa Unbound	Psychic	Dark	680	80	160	60	170	130	80	
799	721	Volcanion	Fire	Water	600	80	110	120	130	90	70	

800 rows × 13 columns



In [3]:

```
pokemons_train, pokemons_test = train_test_split(pokemons, test_size = 0.3, stratify=pokemons['Type 1'])
print(pokemons_train.shape, pokemons_test.shape)
```

(560, 13) (240, 13)

In [4]:

```
features= ['Attack']
dt_classification = DecisionTreeClassifier(max_depth = 3) # Increase max_depth to see effect
dt_classification.fit(pokemons_train[features], pokemons_train['Type 1'])
```

Out[4]:

DecisionTreeClassifier(max_depth=3)

In [5]:

```
plot_tree_classification(dt_classification, features, pokemons['Type 1'].unique())
```

Out[5]:

