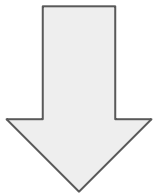
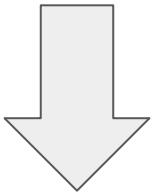
# Random Forest

```
def __init__(self, max_depth=None, min_samples_split=2):
        self.max_depth = max_depth
        self.min_samples_split = min_samples_split
        self.tree = None
```



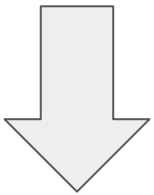
```
def __init__(self, max_depth=None, min_samples_split=2, n_features=None):
        self.max_depth = max_depth
        self.min_samples_split = min_samples_split
        self.n_features = n_features
        self.tree = None
```

```python
def _best_split(self, X, y):
    best_gain = -1
    split_idx, split_thresh = None, None
    n_features = X.shape[1]
    for feature_idx in range(n_features):
        ...
```



```python
def _best_split(self, X, y, feat_idxs):
    best_gain = -1
    split_idx, split_thresh = None, None
    for feature_idx in feat_idxs:
        ...
```

```
def _grow_tree(self, X, y, depth=0):
    ...
    best_feature, best_thresh = self._best_split(X, y)
    ...
```



```
def _grow_tree(self, X, y, depth=0):
    ...
    feat_idxs = np.random.choice(n_features, self.n_features, replace=False)
    best_feature, best_thresh = self._best_split(X, y, feat_idxs)
    ...
```