DAT340 - Assignment 1

March 29, 2022

Group PA 47 - Author: Stefano Ribes, ribes@chalmers.se

This Notebook can be viewed online at this link: https://colab.research.google.com/drive/1K5IDJ8wIE_jblm-tRKh2e3QpovpEBptB?usp=sharing

1 Programming Assignment 1

1.1 Task 0: Setup

[1]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[2]:
```
import os

ASSIGNMENT_ID = 'assignment_1'

data_dir = os.path.join(os.path.abspath(''), 'drive', 'MyDrive')
data_dir = os.path.join(data_dir, 'Colab Notebooks', 'dat340', ASSIGNMENT_ID)
data_dir = os.path.join(data_dir, 'data')
data_dir
```

[2]:
```
'/content/drive/MyDrive/Colab Notebooks/dat340/assignment_1/data'
```

[69]:
```
from IPython.display import set_matplotlib_formats
set_matplotlib_formats('pdf', 'svg')
```

1.2 Task 1: A classification example: fetal heart condition diagnosis

1.2.1 Step 1. Reading the data

[3]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
```

```
# Read the CSV file.
filename = os.path.join(data_dir, 'CTG.csv')
data = pd.read_csv(filename, skiprows=1)

# Select the relevant numerical columns.
selected_cols = ['LB', 'AC', 'FM', 'UC', 'DL', 'DS', 'DP', 'ASTV', 'MSTV',␣
 ↪'ALTV',
                 'MLTV', 'Width', 'Min', 'Max', 'Nmax', 'Nzeros', 'Mode',␣
 ↪'Mean',
                 'Median', 'Variance', 'Tendency', 'NSP']
data = data[selected_cols].dropna()

# Shuffle the dataset.
data_shuffled = data.sample(frac=1.0, random_state=0)

# Split into input part X and output part Y.
X = data_shuffled.drop('NSP', axis=1)

# Map the diagnosis code to a human-readable label.
def to_label(y):
    return [None, 'normal', 'suspect', 'pathologic'][(int(y))]

Y = data_shuffled['NSP'].apply(to_label)

# Partition the data into training and test sets.
Xtrain, Xtest, Ytrain, Ytest = train_test_split(X, Y, test_size=0.2,␣
 ↪random_state=0)
X.head()
```

[3]:

	LB	AC	FM	UC	DL	DS	DP	ASTV	MSTV	ALTV	…	Width	\
658	130.0	1.0	0.0	3.0	0.0	0.0	0.0	24.0	1.2	12.0	…	35.0	
1734	134.0	9.0	1.0	8.0	5.0	0.0	0.0	59.0	1.2	0.0	…	109.0	
1226	125.0	1.0	0.0	4.0	0.0	0.0	0.0	43.0	0.7	31.0	…	21.0	
1808	143.0	0.0	0.0	1.0	0.0	0.0	0.0	69.0	0.3	6.0	…	27.0	
825	152.0	0.0	0.0	4.0	0.0	0.0	0.0	62.0	0.4	59.0	…	25.0	

	Min	Max	Nmax	Nzeros	Mode	Mean	Median	Variance	Tendency
658	120.0	155.0	1.0	0.0	134.0	133.0	135.0	1.0	0.0
1734	80.0	189.0	6.0	0.0	150.0	146.0	150.0	33.0	0.0
1226	120.0	141.0	0.0	0.0	131.0	130.0	132.0	1.0	0.0
1808	132.0	159.0	1.0	0.0	145.0	144.0	146.0	1.0	0.0
825	136.0	161.0	0.0	0.0	159.0	156.0	158.0	1.0	1.0

[5 rows x 21 columns]

1.2.2 Step 2. Training the baseline classifier

[4]:
```
from sklearn.dummy import DummyClassifier

clf = DummyClassifier(strategy='most_frequent')
clf
```

[4]:
```
DummyClassifier(strategy='most_frequent')
```

[5]:
```
from sklearn.model_selection import cross_val_score

baseline_scores = cross_val_score(clf, Xtrain, Ytrain)
baseline_scores
```

[5]:
```
array([0.78235294, 0.78235294, 0.77941176, 0.77941176, 0.77941176])
```

> The result is a NumPy array that contains the accuracies on the different folds in the cross-validation. Aggregate these scores so that you get a single score that you can use to compare different classifiers.

In order to obtain a single score, I simply average the results of the 5 folds returned from the cross-validation.

[16]:
```
from sklearn.metrics import accuracy_score

def aggregate_scores(scores):
    return scores.mean()

baseline_score = aggregate_scores(baseline_scores)
baseline_score
```

[16]:
```
0.7805882352941176
```

1.2.3 Step 3. Trying out some different classifiers

First I import the tested classifiers.

[ ]:
```
# Tree-based classifiers
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
# Linear classifiers
from sklearn.linear_model import Perceptron
from sklearn.linear_model import LogisticRegression
from sklearn.svm import LinearSVC
# Neural network classifier (will take longer time to train)
from sklearn.neural_network import MLPClassifier
```

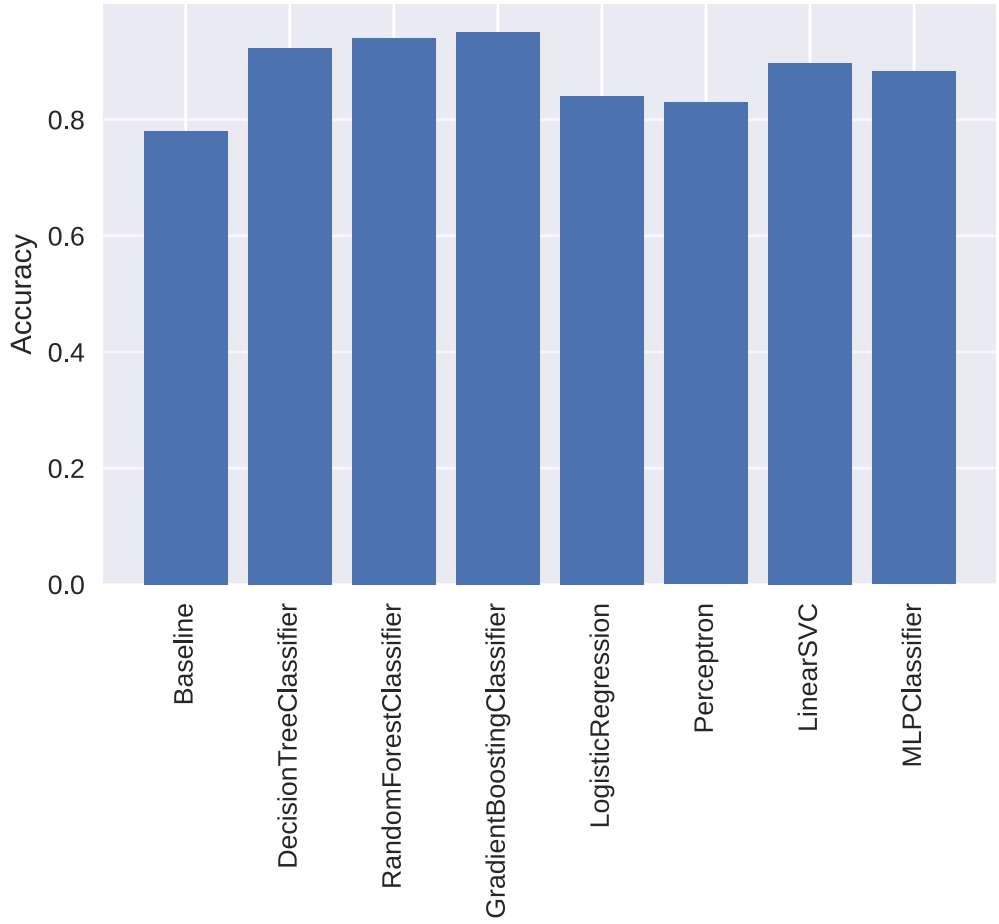
Since I don't have much experience with some of the proposed classifiers, I left most of the parameters at their default value.

[ ]:
```
models = {
    'Baseline' : DummyClassifier(strategy='most_frequent'),
    'DecisionTreeClassifier' : DecisionTreeClassifier(max_depth=None), # None:␣
 ↪till necessary
    'RandomForestClassifier' : RandomForestClassifier(n_estimators=256), # 100
    'GradientBoostingClassifier' : GradientBoostingClassifier(n_estimators=128,␣
 ↪max_leaf_nodes=8),
    'LogisticRegression' : LogisticRegression(solver='saga', max_iter=2000,␣
 ↪tol=0.1, C=5.5),
    'Perceptron' : Perceptron(early_stopping=True, tol=0.01),
    'LinearSVC' : LinearSVC(dual=False, max_iter=2000, tol=0.0001, C=2.),
    'MLPClassifier' : MLPClassifier(hidden_layer_sizes=(128, 64, 64, ),␣
 ↪learning_rate_init=0.01, batch_size=128, max_iter=500),
}

scores = {}

for model_type in models.keys():
    model = models[model_type]
    score = aggregate_scores(cross_val_score(model, Xtrain, Ytrain))
    scores[model_type] = score
    print(f'INFO. Model {model_type} aggregated score: {score:.4f}')

best_model_type = max(scores, key=scores.get)
print('=' * 80)
print(f'INFO. Best Model is {best_model_type} with score:␣
 ↪{scores[best_model_type]:.4f}')
```

```
INFO. Model Baseline aggregated score: 0.7806
INFO. Model DecisionTreeClassifier aggregated score: 0.9235
INFO. Model RandomForestClassifier aggregated score: 0.9406
INFO. Model GradientBoostingClassifier aggregated score: 0.9506
INFO. Model LogisticRegression aggregated score: 0.8406
INFO. Model Perceptron aggregated score: 0.8294
INFO. Model LinearSVC aggregated score: 0.8971
INFO. Model MLPClassifier aggregated score: 0.8829
================================================================================
INFO. Best Model is GradientBoostingClassifier with score: 0.9506
```

[ ]:
```
import matplotlib.pyplot as plt

linspace = [x for x in range(len(models.keys()))]
plt.bar(linspace, scores.values())
plt.xticks(linspace, [f'{m}' for m in models.keys()], rotation=90)
```

```
plt.grid(which='both', axis='y', alpha=0.7, zorder=1)
plt.ylabel('Accuracy')
plt.show()
```



Based on the cross-validation score, the highest score is achieved by the GradientBoostingClassifier model. Because of that, I proceed in training it on the full dataset and then evaluate on the test data.

1.2.4 Step 4. Final evaluation

[ ]:
```
models[best_model_type].fit(Xtrain, Ytrain)
Yguess = models[best_model_type].predict(Xtest)
acc = accuracy_score(Ytest, Yguess)
print(f'INFO. Trained {best_model_type} reached accuracy: {acc:.4f}')
```

```
INFO. Trained GradientBoostingClassifier reached accuracy: 0.9272
```

For the report. In your submitted report, please include a description of the classifier you selected and report its accuracy. (At this point, we are of course not asking you to describe internal workings of various machine learning models that we will cover in detail at later points during the course, but you are of course free to read about them if you're interested.)

Gradient Boosting Classifier The GradientBoostingClassifier is an advanced version of the AdaBoosting model. Like a random forest classifier, it's also ensemble-based, meaning that the classification is done by several "weak" learners, i.e. typically small decision trees.

The main idea, compared to random forests, is not just to create n decision trees and utilize their average score. Rather, each tree is built "from the previous one error". In practice, GradientBoostingClassifier generates n decision trees for which their result minimizes a certain loss function.

In classification task, GradientBoostingClassifier needs to perform a mapping from continuos-values outputs from the weak learners to class probability.

1.3 Task 2: Decision trees for classification

1.3.1 Setup

[7]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

%config InlineBackend.figure_format = 'svg'
plt.style.use('seaborn')
%matplotlib inline

# # Load Dataset
# filename = os.path.join(data_dir, 'iris.csv')
# data = pd.read_csv(filename)
# data_shuffled = data.sample(frac=1.0, random_state=0)
# X = data_shuffled.drop('species', axis=1)
# Y = data_shuffled['species']

# data_shuffled.head()

# Xtrain, Xtest, Ytrain, Ytest = train_test_split(X, Y, test_size=0.4,␣
 ↪random_state=0)
# clf = DecisionTreeClassifier()
# clf.fit(Xtrain, Ytrain);
```

Leaf

[8]:
```
class DecisionTreeLeaf:
    def __init__(self, value):
        self.value = value

    # This method computes the prediction for this leaf node. This will just␣
↪return a constant value.
    def predict(self, x):
        return self.value

    # Utility function to draw a tree visually using graphviz.
    def draw_tree(self, graph, node_counter, names):
        node_id = str(node_counter)
        val_str = f'{self.value:.4g}' if isinstance(self.value, float) else␣
↪str(self.value)
        graph.node(node_id, val_str, style='filled')
        return node_counter+1, node_id

    def __eq__(self, other):
        if isinstance(other, DecisionTreeLeaf):
            return self.value == other.value
        else:
            return False

class DecisionTreeBranch:
    def __init__(self, feature, threshold, low_subtree, high_subtree):
        self.feature = feature
        self.threshold = threshold
        self.low_subtree = low_subtree
        self.high_subtree = high_subtree

    # For a branch node, we compute the prediction by first considering the␣
↪feature, and then
    # calling the upper or lower subtree, depending on whether the feature is␣
↪or isn't greater
    # than the threshold.
    def predict(self, x):
        if x[self.feature] <= self.threshold:
            return self.low_subtree.predict(x)
        else:
            return self.high_subtree.predict(x)

    # Utility function to draw a tree visually using graphviz.
    def draw_tree(self, graph, node_counter, names):
        node_counter, low_id = self.low_subtree.draw_tree(graph, node_counter,␣
↪names)
        node_counter, high_id = self.high_subtree.draw_tree(graph,␣
↪node_counter, names)
```

```
        node_id = str(node_counter)
        fname = f'F{self.feature}' if names is None else names[self.feature]
        lbl = f'{fname} > {self.threshold:.4g}?'
        graph.node(node_id, lbl, shape='box', fillcolor='yellow',␣
↪style='filled, rounded')
        graph.edge(node_id, low_id, 'False')
        graph.edge(node_id, high_id, 'True')
        return node_counter+1, node_id
```

Decision Tree

[9]:
```
from graphviz import Digraph
from sklearn.base import BaseEstimator, ClassifierMixin
from abc import ABC, abstractmethod

class DecisionTree(ABC, BaseEstimator):

    def __init__(self, max_depth):
        super().__init__()
        self.max_depth = max_depth

    # As usual in scikit-learn, the training method is called *fit*. We first␣
↪process the dataset so that
    # we're sure that it's represented as a NumPy matrix. Then we call the␣
↪recursive tree-building method
    # called make_tree (see below).
    def fit(self, X, Y):
        if isinstance(X, pd.DataFrame):
            self.names = X.columns
            X = X.to_numpy()
        elif isinstance(X, list):
            self.names = None
            X = np.array(X)
        else:
            self.names = None
        Y = np.array(Y)
        self.root = self.make_tree(X, Y, self.max_depth)

    def draw_tree(self):
        graph = Digraph()
        self.root.draw_tree(graph, 0, self.names)
        return graph

    # By scikit-learn convention, the method *predict* computes the␣
↪classification or regression output
    # for a set of instances.
```

```
    # To implement it, we call a separate method that carries out the␣
↪prediction for one instance.
    def predict(self, X):
        if isinstance(X, pd.DataFrame):
            X = X.to_numpy()
        return [self.predict_one(x) for x in X]

    # Predicting the output for one instance.
    def predict_one(self, x):
        return self.root.predict(x)

    # This is the recursive training
    def make_tree(self, X, Y, max_depth):

        # We start by computing the default value that will be used if we'll␣
↪return a leaf node.
        # For classifiers, this will be the most common value in Y.
        default_value = self.get_default_value(Y)

        # First the two base cases in the recursion: is the training set␣
↪completely
        # homogeneous, or have we reached the maximum depth? Then we need to␣
↪return a leaf.

        # If we have reached the maximum depth, return a leaf with the majority␣
↪value.
        if max_depth == 0:
            return DecisionTreeLeaf(default_value)

        # If all the instances in the remaining training set have the same␣
↪output value,
        # return a leaf with this value.
        if self.is_homogeneous(Y):
            return DecisionTreeLeaf(default_value)

        # Select the "most useful" feature and split threshold. To rank the␣
↪"usefulness" of features,
        # we use one of the classification or regression criteria.
        # For each feature, we call best_split (defined in a subclass). We then␣
↪maximize over the features.
        n_features = X.shape[1]
        _, best_feature, best_threshold = max(self.best_split(X, Y, feature)␣
↪for feature in range(n_features))

        if best_feature is None:
            return DecisionTreeLeaf(default_value)
```

```
        # Split the training set into subgroups, based on whether the selected␣
↪feature is greater than
        # the threshold or not
        X_low, X_high, Y_low, Y_high = self.split_by_feature(X, Y,␣
↪best_feature, best_threshold)

        # Build the subtrees using a recursive call. Each subtree is associated
        # with a value of the feature.
        low_subtree = self.make_tree(X_low, Y_low, max_depth-1)
        high_subtree = self.make_tree(X_high, Y_high, max_depth-1)

        if low_subtree == high_subtree:
            return low_subtree

        # Return a decision tree branch containing the result.
        return DecisionTreeBranch(best_feature, best_threshold, low_subtree,␣
↪high_subtree)

    # Utility method that splits the data into the "upper" and "lower" part,␣
↪based on a feature
    # and a threshold.
    def split_by_feature(self, X, Y, feature, threshold):
        low = X[:,feature] <= threshold
        high = ~low
        return X[low], X[high], Y[low], Y[high]

    # The following three methods need to be implemented by the classification␣
↪and regression subclasses.

    @abstractmethod
    def get_default_value(self, Y):
        pass

    @abstractmethod
    def is_homogeneous(self, Y):
        pass

    @abstractmethod
    def best_split(self, X, Y, feature):
        pass
```

Tree Classifier

[10]:
```
from collections import Counter
```

```
class TreeClassifier(DecisionTree, ClassifierMixin):

    def __init__(self, max_depth=10, criterion='maj_sum'):
        super().__init__(max_depth)
        self.criterion = criterion

    def fit(self, X, Y):
        # For decision tree classifiers, there are some different ways to␣
↪measure
        # the homogeneity of subsets.
        if self.criterion == 'maj_sum':
            self.criterion_function = majority_sum_scorer
        elif self.criterion == 'info_gain':
            self.criterion_function = info_gain_scorer
        elif self.criterion == 'gini':
            self.criterion_function = gini_scorer
        else:
            raise Exception(f'Unknown criterion: {self.criterion}')
        super().fit(X, Y)
        self.classes_ = sorted(set(Y))

    # Select a default value that is going to be used if we decide to make a␣
↪leaf.
    # We will select the most common value.
    def get_default_value(self, Y):
        self.class_distribution = Counter(Y)
        return self.class_distribution.most_common(1)[0][0]

    # Checks whether a set of output values is homogeneous. In the␣
↪classification case,
    # this means that all output values are identical.
    # We assume that we called get_default_value just before, so that we can␣
↪access
    # the class_distribution attribute. If the class distribution contains just␣
↪one item,
    # this means that the set is homogeneous.
    def is_homogeneous(self, Y):
        return len(self.class_distribution) == 1

    # Finds the best splitting point for a given feature. We'll keep frequency␣
↪tables (Counters)
    # for the upper and lower parts, and then compute the impurity criterion␣
↪using these tables.
    # In the end, we return a triple consisting of
    # - the best score we found, according to the criterion we're using
    # - the id of the feature
```

```
    # - the threshold for the best split
    def best_split(self, X, Y, feature):

        # Create a list of input-output pairs, where we have sorted
        # in ascending order by the input feature we're considering.
        sorted_indices = np.argsort(X[:, feature])
        X_sorted = list(X[sorted_indices, feature])
        Y_sorted = list(Y[sorted_indices])

        n = len(Y)

        # The frequency tables corresponding to the parts *before and including*
        # and *after* the current element.
        low_distr = Counter()
        high_distr = Counter(Y)

        # Keep track of the best result we've seen so far.
        max_score = -np.inf
        max_i = None

        # Go through all the positions (excluding the last position).
        for i in range(0, n-1):

            # Input and output at the current position.
            x_i = X_sorted[i]
            y_i = Y_sorted[i]

            # Update the frequency tables.
            low_distr[y_i] += 1
            high_distr[y_i] -= 1

            # If the input is equal to the input at the next position, we will
            # not consider a split here.
            #x_next = XY[i+1][0]
            x_next = X_sorted[i+1]
            if x_i == x_next:
                continue

            # Compute the homogeneity criterion for a split at this position.
            score = self.criterion_function(i+1, low_distr, n-i-1, high_distr)

            # If this is the best split, remember it.
            if score > max_score:
                max_score = score
                max_i = i
```

```
        # If we didn't find any split (meaning that all inputs are identical),␣
↪return
        # a dummy value.
        if max_i is None:
            return -np.inf, None, None

        # Otherwise, return the best split we found and its score.
        split_point = 0.5*(X_sorted[max_i] + X_sorted[max_i+1])
        return max_score, feature, split_point
```

[11]:
```
def majority_sum_scorer(n_low, low_distr, n_high, high_distr):
    maj_sum_low = low_distr.most_common(1)[0][1]
    maj_sum_high = high_distr.most_common(1)[0][1]
    return maj_sum_low + maj_sum_high

def entropy(distr):
    n = sum(distr.values())
    ps = [n_i/n for n_i in distr.values()]
    return -sum(p*np.log2(p) if p > 0 else 0 for p in ps)

def info_gain_scorer(n_low, low_distr, n_high, high_distr):
    return -(n_low*entropy(low_distr)+n_high*entropy(high_distr))/(n_low+n_high)

def gini_impurity(distr):
    n = sum(distr.values())
    ps = [n_i/n for n_i in distr.values()]
    return 1-sum(p**2 for p in ps)

def gini_scorer(n_low, low_distr, n_high, high_distr):
    return -(n_low*gini_impurity(low_distr)+n_high*gini_impurity(high_distr))/
↪(n_low+n_high)
```

1.3.2 Evaluation

For the report. In your submitted report, please mention what value of max_depth you selected and what accuracy you got.

[12]:
```
model = TreeClassifier(max_depth=10, criterion='maj_sum')
model_score = aggregate_scores(cross_val_score(model, Xtrain, Ytrain))
print(model_score)
```

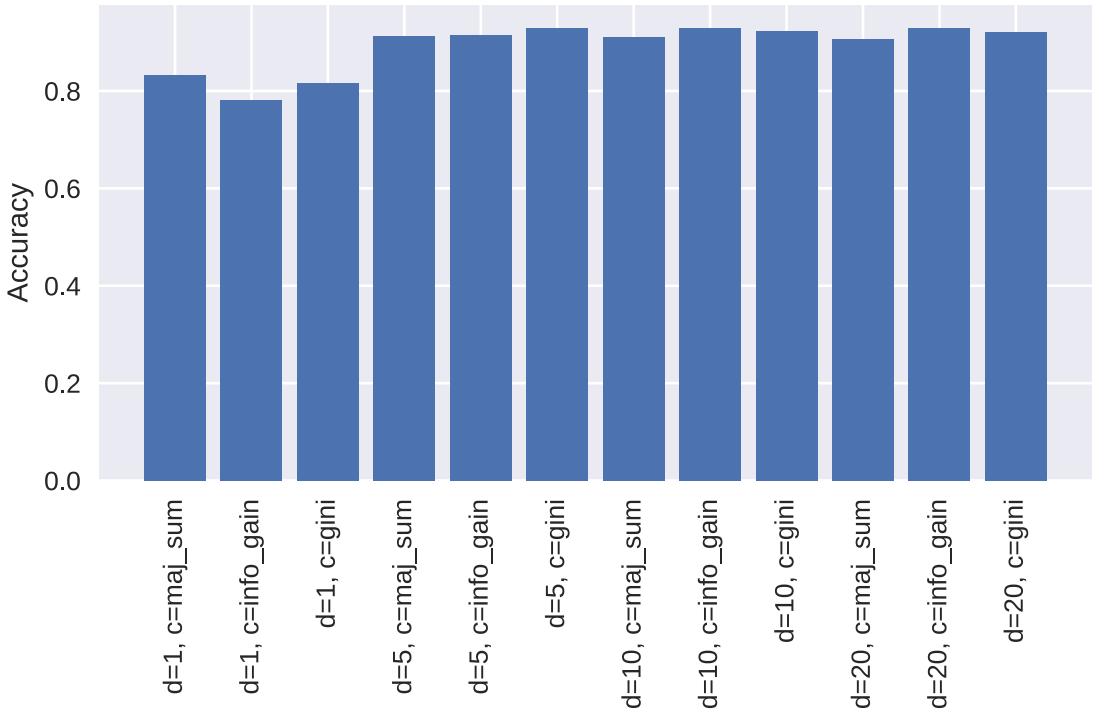
```
0.9099999999999999
```

[13]:
```
import itertools

depths = [1, 5, 10, 20]
```

```
criterions = ['maj_sum', 'info_gain', 'gini']
best_score = -1
scores = {}

for (d, c) in itertools.product(depths, criterions):
    model = TreeClassifier(max_depth=d, criterion=c)
    model_score = aggregate_scores(cross_val_score(model, Xtrain, Ytrain))
    scores[f'd={d}, c={c}'] = model_score
    print(f'INFO. Tree with depth {d} with criterion {c} achieved: {model_score:
 ↪.4f}')
    if model_score > best_score:
        best_model = model
        best_depth = d
        best_criterion = c
        best_score = model_score
print(f'INFO. Tree with depth {best_depth} with criterion {best_criterion}␣
 ↪achieved best score: {best_score}')
```

```
INFO. Tree with depth 1 with criterion maj_sum achieved: 0.8318
INFO. Tree with depth 1 with criterion info_gain achieved: 0.7806
INFO. Tree with depth 1 with criterion gini achieved: 0.8165
INFO. Tree with depth 5 with criterion maj_sum achieved: 0.9124
INFO. Tree with depth 5 with criterion info_gain achieved: 0.9159
INFO. Tree with depth 5 with criterion gini achieved: 0.9300
INFO. Tree with depth 10 with criterion maj_sum achieved: 0.9100
INFO. Tree with depth 10 with criterion info_gain achieved: 0.9294
INFO. Tree with depth 10 with criterion gini achieved: 0.9229
INFO. Tree with depth 20 with criterion maj_sum achieved: 0.9065
INFO. Tree with depth 20 with criterion info_gain achieved: 0.9300
INFO. Tree with depth 20 with criterion gini achieved: 0.9206
INFO. Tree with depth 5 with criterion gini achieved best score: 0.93
```

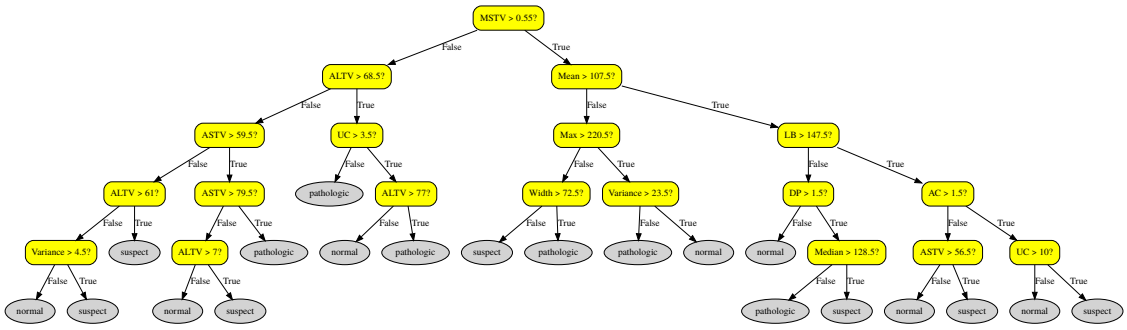
[14]:
```
fig, ax = plt.subplots()
linspace = [x for x in range(len(scores.keys()))]
plt.bar(linspace, scores.values())
plt.xticks(linspace, [f'{m}' for m in scores.keys()], rotation=90)
plt.grid(which='both', axis='y', zorder=1)
plt.ylabel('Accuracy')
plt.tight_layout()
plt.show()
```



[17]:
```
best_model.fit(Xtrain, Ytrain)
acc = accuracy_score(Ytest, best_model.predict(Xtest))
print(f'INFO. Trained TreeClassifier model with max_depth={best_depth} reached␣
 ↪test accuracy of: {acc:.4f}')
```

```
INFO. Trained TreeClassifier model with max_depth=5 reached test accuracy of:
0.9061
```

[ ]:
```
best_model.draw_tree()
```

[ ]:


1.4 Task 3: A regression example: predicting apartment prices

1.4.1 Setup

[18]:
```
# Read the CSV file using Pandas.
filename = os.path.join(data_dir, 'sberbank.csv')
alldata = pd.read_csv(filename)

# Convert the timestamp string to an integer representing the year.
def get_year(timestamp):
    return int(timestamp[:4])
alldata['year'] = alldata.timestamp.apply(get_year)

# Select the 9 input columns and the output column.
selected_columns = ['price_doc', 'year', 'full_sq', 'life_sq', 'floor',␣
 ↪'num_room', 'kitch_sq', 'full_all']
alldata = alldata[selected_columns]
alldata = alldata.dropna()

# Shuffle.
alldata_shuffled = alldata.sample(frac=1.0, random_state=0)

# Separate the input and output columns.
X = alldata_shuffled.drop('price_doc', axis=1)
# For the output, we'll use the log of the sales price.
Y = alldata_shuffled['price_doc'].apply(np.log)

# Split into training and test sets.
Xtrain, Xtest, Ytrain, Ytest = train_test_split(X, Y, test_size=0.2,␣
 ↪random_state=0)
Xtrain.head()
# Ytrain.head()
```

[18]:

	year	full_sq	life_sq	floor	num_room	kitch_sq	full_all
11042	2013	74	74.0	2.0	2.0	1.0	122862
11309	2013	60	1.0	23.0	2.0	1.0	156377
12432	2013	31	1.0	22.0	1.0	1.0	13890
9674	2013	37	20.0	12.0	1.0	10.0	156377
13000	2013	40	40.0	15.0	1.0	1.0	21155

1.4.2 Baseline

[19]:
```
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn.tree import DecisionTreeRegressor
```

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.neural_network import MLPRegressor
```

[20]:
```
from sklearn.dummy import DummyRegressor
from sklearn.model_selection import cross_validate

m1 = DummyRegressor()
cross_validate(m1, Xtrain, Ytrain,␣
↪scoring='neg_mean_squared_error')['test_score']
```

[20]:
```
array([-0.38697391, -0.39964814, -0.37373174, -0.38247357, -0.31537297])
```

1.4.3 Evaluation

Since we are using the negative MSE, we should still search for the model that "maximizes the score".

[ ]:
```
models = {
    'Baseline' : DummyRegressor(),
    'LinearRegression': LinearRegression(),
    'Ridge': Ridge(),
    'Lasso': Lasso(),
    'DecisionTreeRegressor': DecisionTreeRegressor(max_depth=None),
    'RandomForestRegressor': RandomForestRegressor(n_estimators=256),
    'GradientBoostingRegressor': GradientBoostingRegressor(n_estimators=128,␣
↪max_leaf_nodes=8),
    'MLPRegressor': MLPRegressor((128, 128, 128,), learning_rate_init=0.1,␣
↪batch_size=128, max_iter=500),
}

scores = {}

for model_type in models.keys():
    model = models[model_type]
    score = cross_validate(model, Xtrain, Ytrain,
                           scoring='neg_mean_squared_error')['test_score']
    score = aggregate_scores(score)
    scores[model_type] = score
    print(f'INFO. Model {model_type} aggregated score: {score:.4f}')

best_model_type = max(scores, key=scores.get)
print('=' * 80)
print(f'INFO. Best Model is {best_model_type} with score:␣
↪{scores[best_model_type]:.4f}')
```

```
INFO. Model Baseline aggregated score: -0.3716
INFO. Model LinearRegression aggregated score: -0.3432
INFO. Model Ridge aggregated score: -0.3431
INFO. Model Lasso aggregated score: -0.2960
INFO. Model DecisionTreeRegressor aggregated score: -0.6048
INFO. Model RandomForestRegressor aggregated score: -0.3097
INFO. Model GradientBoostingRegressor aggregated score: -0.2808

/usr/local/lib/python3.7/dist-
packages/sklearn/neural_network/_multilayer_perceptron.py:696:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (500) reached and
the optimization hasn't converged yet.
  ConvergenceWarning,

INFO. Model MLPRegressor aggregated score: -23.3665
================================================================================
INFO. Best Model is GradientBoostingRegressor with score: -0.2808
```

[ ]:
```
from sklearn.metrics import mean_squared_error

models[best_model_type].fit(Xtrain, Ytrain)
neg_mse = mean_squared_error(Ytest, models[best_model_type].predict(Xtest))
print(f'INFO. Trained {best_model_type} reached MSE of: {neg_mse:.4f}')
```

```
INFO. Trained GradientBoostingRegressor reached MSE of: 0.2969
```

> For the report. In your submitted report, please include a description of the regression model you selected and report its evaluation score.

A GradientBoostingRegressor model is very similar to a GradientBoostingClassifier model (explained above). It's output doesn't need to be mapped to a probability distribution. Instead, its output is an accumulated continuos sum of the outputs of its weak learners, i.e. decision trees.

1.5 Task 4: Decision trees for regression (Optional)

1.5.1 Step 1. Implementing the Regression Model

[59]:
```
from sklearn.base import RegressorMixin

class TreeRegressor(DecisionTree, RegressorMixin):

    def __init__(self, max_depth=10, criterion='maj_sum'):
        super().__init__(max_depth)
        self.criterion = criterion # Unused

    def fit(self, X, Y):
        # Reference: https://scikit-learn.org/stable/modules/tree.html
```

```
        # # 'squared_error', 'friedman_mse', 'absolute_error', 'poisson'},␣
↪default='squared_error'
        # def squared_error_scorer(n_low, low_distr, n_high, high_distr):
        #     low_mean = np.mean(low_distr.values())
        #     high_mean = np.mean(high_distr.values())
        #     low_loss = np.sum((low_distr.values() - low_mean)**2) / n_low
        #     high_loss = np.sum((high_distr.values() - high_mean)**2) / n_high
        #     return (n_low * low_loss + n_high * high_loss) / (n_low + n_high)

        # # if self.criterion == 'squared_error':
        # #     self.criterion_function = squared_error_scorer
        # self.criterion_function = squared_error_scorer

        super().fit(X, Y)

    # Select a default value that is going to be used if we decide to make a␣
↪leaf.
    # We will select the mean value.
    def get_default_value(self, Y):
        # ============================================================================
        # TODO: To be changed
        # ============================================================================
        # self.distribution = Counter(Y)
        # return self.distribution.most_common(1)[0][0]
        self.distribution = Y
        return np.mean(self.distribution)

    # Checks whether a set of output values is homogeneous. In the␣
↪classification case,
    # this means that all output values are identical.
    # We assume that we called get_default_value just before, so that we can␣
↪access
    # the distribution attribute. If the class distribution contains just one␣
↪item,
    # this means that the set is homogeneous.
    def is_homogeneous(self, Y):
        # ============================================================================
        # TODO: To be changed
        # ============================================================================
        # return len(self.distribution) == 1
        variance_threshold = 1.0
        return np.var(self.distribution) < variance_threshold

    # Finds the best splitting point for a given feature. We'll keep frequency␣
↪tables (Counters)
```

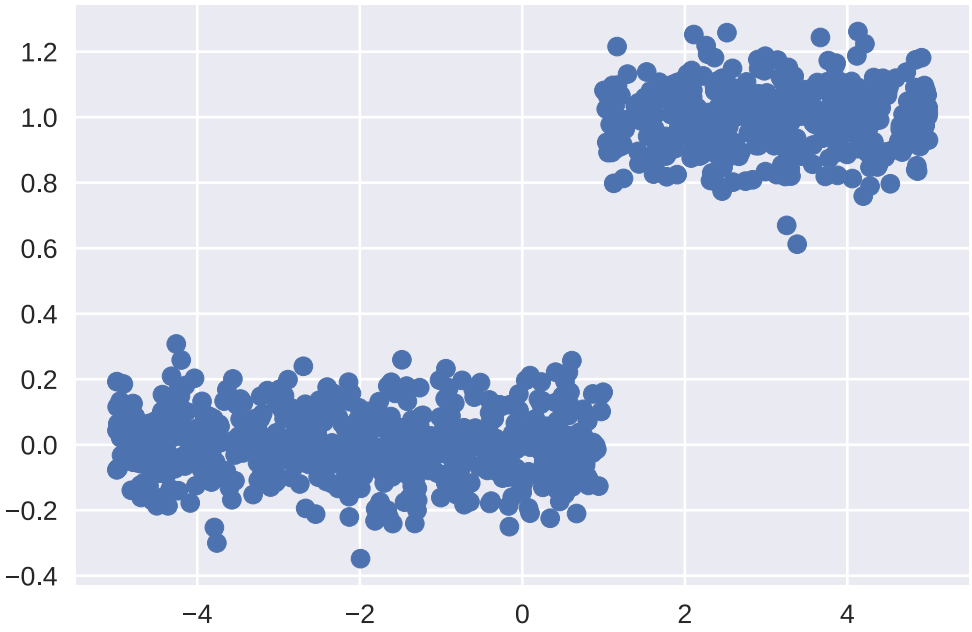
```
    # for the upper and lower parts, and then compute the impurity criterion␣
↪using these tables.
    # In the end, we return a triple consisting of
    # - the best score we found, according to the criterion we're using
    # - the id of the feature
    # - the threshold for the best split
    def best_split(self, X, Y, feature):
        # ================================================================
        # TODO: To be changed
        # ================================================================
        # Create a list of input-output pairs, where we have sorted
        # in ascending order by the input feature we're considering.
        sorted_indices = np.argsort(X[:, feature])
        X_sorted = np.array(list(X[sorted_indices, feature]))
        Y_sorted = np.array(list(Y[sorted_indices]))

        n = len(Y)

        # Keep track of the best result we've seen so far.
        max_score = -np.inf
        max_i = None

        partial_low_squared_sum = np.sum(Y_sorted[0]**2)
        partial_low_sum_squared = np.sum(Y_sorted[0])

        partial_high_squared_sum = np.sum(Y_sorted[1:]**2)
        partial_high_sum_squared = np.sum(Y_sorted[1:])

        var_tot = np.var(Y_sorted)

        for i in range(0, n-1):
            # Input and output at the current position.
            x_i = X_sorted[i]
            y_i = Y_sorted[i]

            partial_low_squared_sum += y_i**2
            partial_low_sum_squared += y_i

            partial_high_squared_sum -= y_i**2
            partial_high_sum_squared -= y_i

            n_low = i + 1
            n_high = n - n_low
            low_var = partial_low_squared_sum / n_low -␣
↪partial_low_sum_squared**2 / (n_low**2)
            high_var = partial_high_squared_sum / n_high -␣
↪partial_high_sum_squared**2 / (n_high**2)
```

```
            # If the input is equal to the input at the next position, we will
            # not consider a split here.
            x_next = X_sorted[i+1]
            if x_i == x_next:
                continue

            # Compute the homogeneity criterion for a split at this position, i.
↪ e. variance reduction
            score = tot_var - (n_high * high_var + n_low * low_var) / n
            print(score)
            # If this is the best split, remember it.
            if score > max_score:
                max_score = score
                max_i = i

        # If we didn't find any split (meaning that all inputs are identical),␣
↪return
        # a dummy value.
        if max_i is None:
            return -np.inf, None, None

        # Otherwise, return the best split we found and its score.
        split_point = 0.5*(X_sorted[max_i] + X_sorted[max_i+1])
        return max_score, feature, split_point
```

1.5.2 Step 2. Sanity Check

The following function will generate a small number of training examples for a simple regression task with one input variable.

[60]:
```
def make_some_data(n):
    x = np.random.uniform(-5, 5, size=n)
    Y = (x > 1) + 0.1 * np.random.normal(size=n)
    X = x.reshape(n, 1) # X needs to be a 2-dimensional matrix
    return X, Y

X, Y = make_some_data(1000)
plt.scatter(X, Y)
plt.show()
```



For the report. If you consider the data-generating function, what kind of decision tree would we want to describe this data?

[61]:
```
Xtrain, Xtest, Ytrain, Ytest = train_test_split(X, Y, test_size=0.2,␣
↪random_state=0)
```

Let's do some training and prediction on the generated dataset.

[62]:
```
depths = [x for x in range(13)]
criterions = ['squared_error']
best_score = -1
scores = {}

for (d, c) in itertools.product(depths, criterions):
    model = TreeRegressor(max_depth=d, criterion=c)
    model.fit(Xtrain, Ytrain)
    score = cross_validate(model, Xtrain, Ytrain,
                           scoring='neg_mean_squared_error')['test_score']
    score = aggregate_scores(score)
    print(f'INFO. TreeRegressor max_depth={d} reached score: {score:.4f}')
```

```
INFO. TreeRegressor max_depth=0 reached score: -0.2473
INFO. TreeRegressor max_depth=1 reached score: -0.2473
INFO. TreeRegressor max_depth=2 reached score: -0.2473
INFO. TreeRegressor max_depth=3 reached score: -0.2473
INFO. TreeRegressor max_depth=4 reached score: -0.2473
INFO. TreeRegressor max_depth=5 reached score: -0.2473
```

```
INFO. TreeRegressor max_depth=6 reached score: -0.2473
INFO. TreeRegressor max_depth=7 reached score: -0.2473
INFO. TreeRegressor max_depth=8 reached score: -0.2473
INFO. TreeRegressor max_depth=9 reached score: -0.2473
INFO. TreeRegressor max_depth=10 reached score: -0.2473
INFO. TreeRegressor max_depth=11 reached score: -0.2473
INFO. TreeRegressor max_depth=12 reached score: -0.2473
```

1.5.3 Step 3. Predicting apartment prices using decision tree regression

Let's load the dataset again.

[44]:
```
# Read the CSV file using Pandas.
filename = os.path.join(data_dir, 'sberbank.csv')
alldata = pd.read_csv(filename)

# Convert the timestamp string to an integer representing the year.
def get_year(timestamp):
    return int(timestamp[:4])
alldata['year'] = alldata.timestamp.apply(get_year)

# Select the 9 input columns and the output column.
selected_columns = ['price_doc', 'year', 'full_sq', 'life_sq', 'floor',␣
 ↪'num_room', 'kitch_sq', 'full_all']
alldata = alldata[selected_columns]
alldata = alldata.dropna()

# Shuffle.
alldata_shuffled = alldata.sample(frac=1.0, random_state=0)

# Separate the input and output columns.
X = alldata_shuffled.drop('price_doc', axis=1)
# For the output, we'll use the log of the sales price.
Y = alldata_shuffled['price_doc'].apply(np.log)

# Split into training and test sets.
Xtrain, Xtest, Ytrain, Ytest = train_test_split(X, Y, test_size=0.2,␣
 ↪random_state=0)
Xtrain.head()
# Ytrain.head()
```

[44]:

	year	full_sq	life_sq	floor	num_room	kitch_sq	full_all
11042	2013	74	74.0	2.0	2.0	1.0	122862
11309	2013	60	1.0	23.0	2.0	1.0	156377
12432	2013	31	1.0	22.0	1.0	1.0	13890
9674	2013	37	20.0	12.0	1.0	10.0	156377
13000	2013	40	40.0	15.0	1.0	1.0	21155

Let's do cross-validation on the dataset.

[45]:
```
depths = [x for x in range(13)]
criterions = ['maj_sum', 'info_gain', 'gini']
best_score = -1
models = {}
scores = {}

for (d, c) in itertools.product(depths, criterions):
    model = TreeRegressor(max_depth=d, criterion=c)
    score = cross_validate(model, Xtrain, Ytrain,
                           scoring='neg_mean_squared_error')['test_score']
    score = aggregate_scores(score)
    scores[f'd={d}, c={c}'] = score
    models[f'd={d}, c={c}'] = model
    print(f'INFO. Tree with depth {d} with criterion {c} achieved: {score:.4f}')
    if score > best_score:
        best_model = f'd={d}, c={c}'
        best_score = score
print(f'INFO. Tree with {best_model} achieved best score: {scores[best_model]:.
 ↪4f}')
```

```
INFO. Tree with depth 0 with criterion maj_sum achieved: -0.3716
INFO. Tree with depth 0 with criterion info_gain achieved: -0.3716
INFO. Tree with depth 0 with criterion gini achieved: -0.3716
INFO. Tree with depth 1 with criterion maj_sum achieved: -0.3716
INFO. Tree with depth 1 with criterion info_gain achieved: -0.3716
INFO. Tree with depth 1 with criterion gini achieved: -0.3716
INFO. Tree with depth 2 with criterion maj_sum achieved: -0.3716
INFO. Tree with depth 2 with criterion info_gain achieved: -0.3716
INFO. Tree with depth 2 with criterion gini achieved: -0.3716
INFO. Tree with depth 3 with criterion maj_sum achieved: -0.3716
INFO. Tree with depth 3 with criterion info_gain achieved: -0.3716
INFO. Tree with depth 3 with criterion gini achieved: -0.3716
INFO. Tree with depth 4 with criterion maj_sum achieved: -0.3716
INFO. Tree with depth 4 with criterion info_gain achieved: -0.3716
INFO. Tree with depth 4 with criterion gini achieved: -0.3716
INFO. Tree with depth 5 with criterion maj_sum achieved: -0.3716
INFO. Tree with depth 5 with criterion info_gain achieved: -0.3716
INFO. Tree with depth 5 with criterion gini achieved: -0.3716
INFO. Tree with depth 6 with criterion maj_sum achieved: -0.3716
INFO. Tree with depth 6 with criterion info_gain achieved: -0.3716
INFO. Tree with depth 6 with criterion gini achieved: -0.3716
INFO. Tree with depth 7 with criterion maj_sum achieved: -0.3716
INFO. Tree with depth 7 with criterion info_gain achieved: -0.3716
INFO. Tree with depth 7 with criterion gini achieved: -0.3716
INFO. Tree with depth 8 with criterion maj_sum achieved: -0.3716
INFO. Tree with depth 8 with criterion info_gain achieved: -0.3716
```

```
INFO. Tree with depth 8 with criterion gini achieved: -0.3716
INFO. Tree with depth 9 with criterion maj_sum achieved: -0.3716
INFO. Tree with depth 9 with criterion info_gain achieved: -0.3716
INFO. Tree with depth 9 with criterion gini achieved: -0.3716
INFO. Tree with depth 10 with criterion maj_sum achieved: -0.3716
INFO. Tree with depth 10 with criterion info_gain achieved: -0.3716
INFO. Tree with depth 10 with criterion gini achieved: -0.3716
INFO. Tree with depth 11 with criterion maj_sum achieved: -0.3716
INFO. Tree with depth 11 with criterion info_gain achieved: -0.3716
INFO. Tree with depth 11 with criterion gini achieved: -0.3716
INFO. Tree with depth 12 with criterion maj_sum achieved: -0.3716
INFO. Tree with depth 12 with criterion info_gain achieved: -0.3716
INFO. Tree with depth 12 with criterion gini achieved: -0.3716
INFO. Tree with d=0, c=maj_sum achieved best score: -0.3716
```

Once the best model is identified, let's train and evaluate it on the dataset.

[48]:
```
from sklearn.metrics import mean_squared_error

models[best_model].fit(Xtrain, Ytrain)
neg_mse = mean_squared_error(Ytest, models[best_model].predict(Xtest))
print(f'INFO. Trained TreeRegressor reached MSE of: {neg_mse:.4f}')
```

```
INFO. Trained TreeRegressor reached MSE of: 0.4022
```

> For the report. In your submitted report, please describe what tree depth you used and the evaluation score you got on the test set.

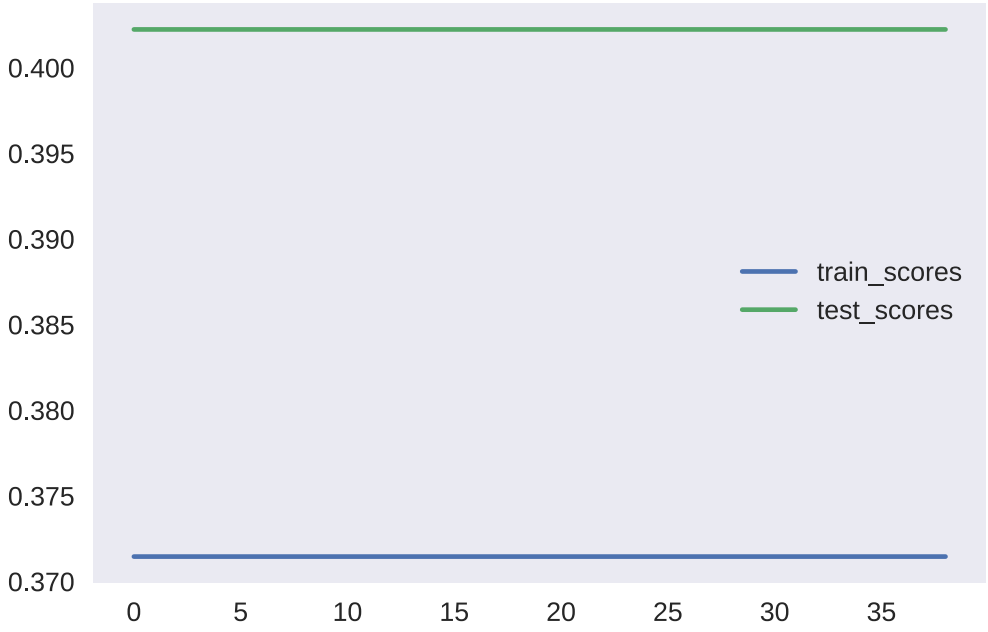
1.5.4 Step 4. Underfitting and overfitting

[57]:
```
eval_scores_train = {}
eval_scores_test = {}

for model_id in models.keys():
    models[model_id].fit(Xtrain, Ytrain)
    neg_mse_train = mean_squared_error(Ytrain, models[model_id].predict(Xtrain))
    neg_mse_test = mean_squared_error(Ytest, models[model_id].predict(Xtest))
    eval_scores_train[model_id] = neg_mse_train
    eval_scores_test[model_id] = neg_mse_test
eval_scores_test.values()
```

[57]:
```
dict_values([0.402233142382155, 0.402233142382155, 0.402233142382155,
0.402233142382155, 0.402233142382155, 0.402233142382155, 0.402233142382155,
0.402233142382155, 0.402233142382155, 0.402233142382155, 0.402233142382155,
0.402233142382155, 0.402233142382155, 0.402233142382155, 0.402233142382155,
0.402233142382155, 0.402233142382155, 0.402233142382155, 0.402233142382155,
0.402233142382155, 0.402233142382155, 0.402233142382155, 0.402233142382155,
0.402233142382155, 0.402233142382155, 0.402233142382155, 0.402233142382155,
```

```
0.402233142382155, 0.402233142382155, 0.402233142382155, 0.402233142382155,
0.402233142382155, 0.402233142382155, 0.402233142382155, 0.402233142382155,
0.402233142382155, 0.402233142382155, 0.402233142382155, 0.402233142382155])
```

[58]:
```
# Plotting
# train_scores = eval_scores_train['gini' in eval_scores_train.keys()]
# test_scores = eval_scores_test['gini' in eval_scores_test.keys()]
plt.plot(list(eval_scores_train.values()), label='train_scores')
plt.plot(list(eval_scores_test.values()), label='test_scores')
plt.legend()
plt.grid()
plt.show()
```



For the report. Please include this plot in the report, and comment on the differences between the two curves.

1.6 Converting Notebook to PDF

This part and the following can be ignored.

[97]:
```
!apt-get update
!apt-get install inkscape
!add-apt-repository universe
!add-apt-repository ppa:inkscape.dev/stable
!apt-get update
```

```
!apt install inkscape
```

```
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:2 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ InRelease
Ign:3 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64
InRelease
Hit:4 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic InRelease
Hit:5 http://archive.ubuntu.com/ubuntu bionic InRelease
Ign:6 https://developer.download.nvidia.com/compute/machine-
learning/repos/ubuntu1804/x86_64  InRelease
Hit:7 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64
Release
Hit:8 http://archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:9 https://developer.download.nvidia.com/compute/machine-
learning/repos/ubuntu1804/x86_64  Release
Hit:10 http://ppa.launchpad.net/cran/libgit2/ubuntu bionic InRelease
Hit:11 http://archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:12 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic InRelease
Hit:13 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic InRelease
Reading package lists… Done
Reading package lists… Done
Building dependency tree
Reading state information… Done
The following additional packages will be installed:
  aspell aspell-en dictionaries-common emacsen-common enchant fig2dev gawk
  ghostscript gsfonts hunspell-en-us imagemagick imagemagick-6-common
  imagemagick-6.q16 libaspell15 libatkmm-1.6-1v5 libcairomm-1.0-1v5
  libcdr-0.1-1 libdbus-glib-1-2 libdjvulibre-text libdjvulibre21
  libenchant1c2a libgail-common libgail18 libglibmm-2.4-1v5 libgsl23
  libgslcblas0 libgtk2.0-0 libgtk2.0-bin libgtk2.0-common libgtkmm-2.4-1v5
  libgtkspell0 libhunspell-1.6-0 libimage-magick-perl libimage-magick-q16-perl
  liblqr-1-0 libmagick++-6.q16-7 libmagickcore-6.q16-3
  libmagickcore-6.q16-3-extra libmagickwand-6.q16-3 libnetpbm10
  libpangomm-1.4-1v5 libpoppler-glib8 librevenge-0.0-0 libsigc++-2.0-0v5
  libsigsegv2 libtext-iconv-perl libvisio-0.1-1 libwmf-bin libwmf0.2-7
  libwpd-0.10-10 libwpg-0.3-3 netpbm python-bs4 python-chardet python-html5lib
  python-lxml python-pkg-resources python-scour python-six python-webencodings
  python3-pkg-resources python3-scour python3-six scour transfig
Suggested packages:
  aspell-doc spellutils wordlist xfig gawk-doc ghostscript-x hunspell
  openoffice.org-hunspell | openoffice.org-core imagemagick-doc autotrace
  cups-bsd | lpr | lprng enscript gimp gnuplot grads hp2xx html2ps mplayer
  povray radiance sane-utils ufraw-batch dia libsvg-perl libxml-xql-perl
  pstoedit python-uniconvertor libenchant-voikko gsl-ref-psdoc | gsl-doc-pdf
  | gsl-doc-info | gsl-ref-html gvfs libjxr-tools libwmf0.2-7-gtk
  python-genshi python-lxml-dbg python-lxml-doc python-setuptools
  python3-setuptools gir1.2-rsvg-2.0 python3-gi-cairo
```

```
The following NEW packages will be installed:
  aspell aspell-en dictionaries-common emacsen-common enchant fig2dev gawk
  ghostscript gsfonts hunspell-en-us imagemagick imagemagick-6-common
  imagemagick-6.q16 inkscape libaspell15 libatkmm-1.6-1v5 libcairomm-1.0-1v5
  libcdr-0.1-1 libdbus-glib-1-2 libdjvulibre-text libdjvulibre21
  libenchant1c2a libgail-common libgail18 libglibmm-2.4-1v5 libgsl23
  libgslcblas0 libgtk2.0-0 libgtk2.0-bin libgtk2.0-common libgtkmm-2.4-1v5
  libgtkspell0 libhunspell-1.6-0 libimage-magick-perl libimage-magick-q16-perl
  liblqr-1-0 libmagick++-6.q16-7 libmagickcore-6.q16-3
  libmagickcore-6.q16-3-extra libmagickwand-6.q16-3 libnetpbm10
  libpangomm-1.4-1v5 libpoppler-glib8 librevenge-0.0-0 libsigc++-2.0-0v5
  libsigsegv2 libtext-iconv-perl libvisio-0.1-1 libwmf-bin libwmf0.2-7
  libwpd-0.10-10 libwpg-0.3-3 netpbm python-bs4 python-chardet python-html5lib
  python-lxml python-pkg-resources python-scour python-six python-webencodings
  python3-pkg-resources python3-scour python3-six scour transfig
0 upgraded, 66 newly installed, 0 to remove and 79 not upgraded.
Need to get 33.7 MB of archives.
After this operation, 189 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 libsigsegv2 amd64
2.12-1 [14.7 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic/main amd64 gawk amd64
1:4.1.4+dfsg-1build1 [401 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64
imagemagick-6-common all 8:6.9.7.4+dfsg-16ubuntu6.12 [60.3 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic/main amd64 liblqr-1-0 amd64
0.4.2-2.1 [27.7 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64
libmagickcore-6.q16-3 amd64 8:6.9.7.4+dfsg-16ubuntu6.12 [1,621 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 libimage-
magick-q16-perl amd64 8:6.9.7.4+dfsg-16ubuntu6.12 [96.8 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 libimage-
magick-perl all 8:6.9.7.4+dfsg-16ubuntu6.12 [52.9 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64
libmagickwand-6.q16-3 amd64 8:6.9.7.4+dfsg-16ubuntu6.12 [292 kB]
Get:9 http://archive.ubuntu.com/ubuntu bionic/main amd64 libtext-iconv-perl
amd64 1.7-5build6 [13.0 kB]
Get:10 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libaspell15
amd64 0.60.7~20110707-4ubuntu0.2 [310 kB]
Get:11 http://archive.ubuntu.com/ubuntu bionic/main amd64 emacsen-common all
2.0.8 [17.6 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic/main amd64 dictionaries-common
all 1.27.2 [186 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 aspell amd64
0.60.7~20110707-4ubuntu0.2 [87.7 kB]
Get:14 http://archive.ubuntu.com/ubuntu bionic/main amd64 aspell-en all
2017.08.24-0-0.1 [298 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic/main amd64 hunspell-en-us all
1:2017.08.24 [168 kB]
```

```
Get:16 http://archive.ubuntu.com/ubuntu bionic/main amd64 libhunspell-1.6-0
amd64 1.6.2-1 [154 kB]
Get:17 http://archive.ubuntu.com/ubuntu bionic/main amd64 libenchant1c2a amd64
1.6.0-11.1 [64.4 kB]
Get:18 http://archive.ubuntu.com/ubuntu bionic/main amd64 enchant amd64
1.6.0-11.1 [12.2 kB]
Get:19 http://archive.ubuntu.com/ubuntu bionic/universe amd64 fig2dev amd64
1:3.2.6a-6ubuntu1 [629 kB]
Get:20 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 ghostscript
amd64 9.26~dfsg+0-0ubuntu0.18.04.15 [51.4 kB]
Get:21 http://archive.ubuntu.com/ubuntu bionic/main amd64 gsfonts all
1:8.11+urwcyr1.0.7~pre44-4.4 [3,120 kB]
Get:22 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64
imagemagick-6.q16 amd64 8:6.9.7.4+dfsg-16ubuntu6.12 [423 kB]
Get:23 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 imagemagick
amd64 8:6.9.7.4+dfsg-16ubuntu6.12 [14.2 kB]
Get:24 http://archive.ubuntu.com/ubuntu bionic/main amd64 libsigc++-2.0-0v5
amd64 2.10.0-2 [10.9 kB]
Get:25 http://archive.ubuntu.com/ubuntu bionic/main amd64 libglibmm-2.4-1v5
amd64 2.56.0-1 [516 kB]
Get:26 http://archive.ubuntu.com/ubuntu bionic/main amd64 libatkmm-1.6-1v5 amd64
2.24.2-3 [62.6 kB]
Get:27 http://archive.ubuntu.com/ubuntu bionic/main amd64 libcairomm-1.0-1v5
amd64 1.12.2-3 [37.1 kB]
Get:28 http://archive.ubuntu.com/ubuntu bionic/main amd64 librevenge-0.0-0 amd64
0.0.4-6ubuntu2 [203 kB]
Get:29 http://archive.ubuntu.com/ubuntu bionic/main amd64 libcdr-0.1-1 amd64
0.1.4-1build1 [333 kB]
Get:30 http://archive.ubuntu.com/ubuntu bionic/main amd64 libdbus-glib-1-2 amd64
0.110-2 [58.3 kB]
Get:31 http://archive.ubuntu.com/ubuntu bionic/universe amd64 libgslcblas0 amd64
2.4+dfsg-6 [79.7 kB]
Get:32 http://archive.ubuntu.com/ubuntu bionic/universe amd64 libgsl23 amd64
2.4+dfsg-6 [823 kB]
Get:33 http://archive.ubuntu.com/ubuntu bionic/main amd64 libgtk2.0-common all
2.24.32-1ubuntu1 [125 kB]
Get:34 http://archive.ubuntu.com/ubuntu bionic/main amd64 libgtk2.0-0 amd64
2.24.32-1ubuntu1 [1,769 kB]
Get:35 http://archive.ubuntu.com/ubuntu bionic/main amd64 libpangomm-1.4-1v5
amd64 2.40.1-4 [42.5 kB]
Get:36 http://archive.ubuntu.com/ubuntu bionic/main amd64 libgtkmm-2.4-1v5 amd64
1:2.24.5-2 [666 kB]
Get:37 http://archive.ubuntu.com/ubuntu bionic/universe amd64 libgtkspell0 amd64
2.0.16-1.2 [20.4 kB]
Get:38 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64
libmagick++-6.q16-7 amd64 8:6.9.7.4+dfsg-16ubuntu6.12 [139 kB]
Get:39 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libpoppler-
glib8 amd64 0.62.0-2ubuntu2.12 [108 kB]
```

```
Get:40 http://archive.ubuntu.com/ubuntu bionic/main amd64 libvisio-0.1-1 amd64
0.1.6-1build1 [210 kB]
Get:41 http://archive.ubuntu.com/ubuntu bionic/main amd64 libwpd-0.10-10 amd64
0.10.2-2 [161 kB]
Get:42 http://archive.ubuntu.com/ubuntu bionic/main amd64 libwpg-0.3-3 amd64
0.3.1-3 [44.6 kB]
Get:43 http://archive.ubuntu.com/ubuntu bionic/universe amd64 inkscape amd64
0.92.3-1 [16.7 MB]
Get:44 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libdjvulibre-
text all 3.5.27.1-8ubuntu0.4 [49.4 kB]
Get:45 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libdjvulibre21
amd64 3.5.27.1-8ubuntu0.4 [561 kB]
Get:46 http://archive.ubuntu.com/ubuntu bionic/main amd64 libgail18 amd64
2.24.32-1ubuntu1 [14.2 kB]
Get:47 http://archive.ubuntu.com/ubuntu bionic/main amd64 libgail-common amd64
2.24.32-1ubuntu1 [112 kB]
Get:48 http://archive.ubuntu.com/ubuntu bionic/main amd64 libgtk2.0-bin amd64
2.24.32-1ubuntu1 [7,536 B]
Get:49 http://archive.ubuntu.com/ubuntu bionic/main amd64 libwmf0.2-7 amd64
0.2.8.4-12 [150 kB]
Get:50 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64
libmagickcore-6.q16-3-extra amd64 8:6.9.7.4+dfsg-16ubuntu6.12 [62.4 kB]
Get:51 http://archive.ubuntu.com/ubuntu bionic/main amd64 libnetpbm10 amd64
2:10.0-15.3build1 [58.0 kB]
Get:52 http://archive.ubuntu.com/ubuntu bionic/universe amd64 libwmf-bin amd64
0.2.8.4-12 [17.0 kB]
Get:53 http://archive.ubuntu.com/ubuntu bionic/main amd64 netpbm amd64
2:10.0-15.3build1 [1,017 kB]
Get:54 http://archive.ubuntu.com/ubuntu bionic/main amd64 python-bs4 all 4.6.0-1
[67.9 kB]
Get:55 http://archive.ubuntu.com/ubuntu bionic/main amd64 python-pkg-resources
all 39.0.1-2 [128 kB]
Get:56 http://archive.ubuntu.com/ubuntu bionic/main amd64 python-chardet all
3.0.4-1 [80.3 kB]
Get:57 http://archive.ubuntu.com/ubuntu bionic/main amd64 python-six all
1.11.0-2 [11.3 kB]
Get:58 http://archive.ubuntu.com/ubuntu bionic/main amd64 python-webencodings
all 0.5-2 [10.3 kB]
Get:59 http://archive.ubuntu.com/ubuntu bionic/main amd64 python-html5lib all
0.999999999-1 [83.6 kB]
Get:60 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 python-lxml
amd64 4.2.1-1ubuntu0.6 [898 kB]
Get:61 http://archive.ubuntu.com/ubuntu bionic/main amd64 python3-six all
1.11.0-2 [11.4 kB]
Get:62 http://archive.ubuntu.com/ubuntu bionic/main amd64 python3-pkg-resources
all 39.0.1-2 [98.8 kB]
Get:63 http://archive.ubuntu.com/ubuntu bionic/universe amd64 python3-scour all
0.36-2 [44.8 kB]
```

```
Get:64 http://archive.ubuntu.com/ubuntu bionic/universe amd64 scour all 0.36-2
[7,372 B]
Get:65 http://archive.ubuntu.com/ubuntu bionic/universe amd64 python-scour all
0.36-2 [42.7 kB]
Get:66 http://archive.ubuntu.com/ubuntu bionic/universe amd64 transfig all
1:3.2.6a-6ubuntu1 [2,920 B]
Fetched 33.7 MB in 1s (47.3 MB/s)
Extracting templates from packages: 100%
Preconfiguring packages …
Selecting previously unselected package libsigsegv2:amd64.
(Reading database … 182395 files and directories currently installed.)
Preparing to unpack …/libsigsegv2_2.12-1_amd64.deb …
Unpacking libsigsegv2:amd64 (2.12-1) …
Setting up libsigsegv2:amd64 (2.12-1) …
Selecting previously unselected package gawk.
(Reading database … 182402 files and directories currently installed.)
Preparing to unpack …/00-gawk_1%3a4.1.4+dfsg-1build1_amd64.deb …
Unpacking gawk (1:4.1.4+dfsg-1build1) …
Selecting previously unselected package imagemagick-6-common.
Preparing to unpack
…/01-imagemagick-6-common_8%3a6.9.7.4+dfsg-16ubuntu6.12_all.deb …
Unpacking imagemagick-6-common (8:6.9.7.4+dfsg-16ubuntu6.12) …
Selecting previously unselected package liblqr-1-0:amd64.
Preparing to unpack …/02-liblqr-1-0_0.4.2-2.1_amd64.deb …
Unpacking liblqr-1-0:amd64 (0.4.2-2.1) …
Selecting previously unselected package libmagickcore-6.q16-3:amd64.
Preparing to unpack
…/03-libmagickcore-6.q16-3_8%3a6.9.7.4+dfsg-16ubuntu6.12_amd64.deb …
Unpacking libmagickcore-6.q16-3:amd64 (8:6.9.7.4+dfsg-16ubuntu6.12) …
Selecting previously unselected package libimage-magick-q16-perl.
Preparing to unpack …/04-libimage-
magick-q16-perl_8%3a6.9.7.4+dfsg-16ubuntu6.12_amd64.deb …
Unpacking libimage-magick-q16-perl (8:6.9.7.4+dfsg-16ubuntu6.12) …
Selecting previously unselected package libimage-magick-perl.
Preparing to unpack …/05-libimage-magick-
perl_8%3a6.9.7.4+dfsg-16ubuntu6.12_all.deb …
Unpacking libimage-magick-perl (8:6.9.7.4+dfsg-16ubuntu6.12) …
Selecting previously unselected package libmagickwand-6.q16-3:amd64.
Preparing to unpack
…/06-libmagickwand-6.q16-3_8%3a6.9.7.4+dfsg-16ubuntu6.12_amd64.deb …
Unpacking libmagickwand-6.q16-3:amd64 (8:6.9.7.4+dfsg-16ubuntu6.12) …
Selecting previously unselected package libtext-iconv-perl.
Preparing to unpack …/07-libtext-iconv-perl_1.7-5build6_amd64.deb …
Unpacking libtext-iconv-perl (1.7-5build6) …
Selecting previously unselected package libaspell15:amd64.
Preparing to unpack …/08-libaspell15_0.60.7~20110707-4ubuntu0.2_amd64.deb …
Unpacking libaspell15:amd64 (0.60.7~20110707-4ubuntu0.2) …
Selecting previously unselected package emacsen-common.
```

```
Preparing to unpack …/09-emacsen-common_2.0.8_all.deb …
Unpacking emacsen-common (2.0.8) …
Selecting previously unselected package dictionaries-common.
Preparing to unpack …/10-dictionaries-common_1.27.2_all.deb …
Adding 'diversion of /usr/share/dict/words to /usr/share/dict/words.pre-
dictionaries-common by dictionaries-common'
Unpacking dictionaries-common (1.27.2) …
Selecting previously unselected package aspell.
Preparing to unpack …/11-aspell_0.60.7~20110707-4ubuntu0.2_amd64.deb …
Unpacking aspell (0.60.7~20110707-4ubuntu0.2) …
Selecting previously unselected package aspell-en.
Preparing to unpack …/12-aspell-en_2017.08.24-0-0.1_all.deb …
Unpacking aspell-en (2017.08.24-0-0.1) …
Selecting previously unselected package hunspell-en-us.
Preparing to unpack …/13-hunspell-en-us_1%3a2017.08.24_all.deb …
Unpacking hunspell-en-us (1:2017.08.24) …
Selecting previously unselected package libhunspell-1.6-0:amd64.
Preparing to unpack …/14-libhunspell-1.6-0_1.6.2-1_amd64.deb …
Unpacking libhunspell-1.6-0:amd64 (1.6.2-1) …
Selecting previously unselected package libenchant1c2a:amd64.
Preparing to unpack …/15-libenchant1c2a_1.6.0-11.1_amd64.deb …
Unpacking libenchant1c2a:amd64 (1.6.0-11.1) …
Selecting previously unselected package enchant.
Preparing to unpack …/16-enchant_1.6.0-11.1_amd64.deb …
Unpacking enchant (1.6.0-11.1) …
Selecting previously unselected package fig2dev.
Preparing to unpack …/17-fig2dev_1%3a3.2.6a-6ubuntu1_amd64.deb …
Unpacking fig2dev (1:3.2.6a-6ubuntu1) …
Selecting previously unselected package ghostscript.
Preparing to unpack …/18-ghostscript_9.26~dfsg+0-0ubuntu0.18.04.15_amd64.deb
…
Unpacking ghostscript (9.26~dfsg+0-0ubuntu0.18.04.15) …
Selecting previously unselected package gsfonts.
Preparing to unpack …/19-gsfonts_1%3a8.11+urwcyr1.0.7~pre44-4.4_all.deb …
Unpacking gsfonts (1:8.11+urwcyr1.0.7~pre44-4.4) …
Selecting previously unselected package imagemagick-6.q16.
Preparing to unpack
…/20-imagemagick-6.q16_8%3a6.9.7.4+dfsg-16ubuntu6.12_amd64.deb …
Unpacking imagemagick-6.q16 (8:6.9.7.4+dfsg-16ubuntu6.12) …
Selecting previously unselected package imagemagick.
Preparing to unpack …/21-imagemagick_8%3a6.9.7.4+dfsg-16ubuntu6.12_amd64.deb
…
Unpacking imagemagick (8:6.9.7.4+dfsg-16ubuntu6.12) …
Selecting previously unselected package libsigc++-2.0-0v5:amd64.
Preparing to unpack …/22-libsigc++-2.0-0v5_2.10.0-2_amd64.deb …
Unpacking libsigc++-2.0-0v5:amd64 (2.10.0-2) …
Selecting previously unselected package libglibmm-2.4-1v5:amd64.
Preparing to unpack …/23-libglibmm-2.4-1v5_2.56.0-1_amd64.deb …
```

```
Unpacking libglibmm-2.4-1v5:amd64 (2.56.0-1) …
Selecting previously unselected package libatkmm-1.6-1v5:amd64.
Preparing to unpack …/24-libatkmm-1.6-1v5_2.24.2-3_amd64.deb …
Unpacking libatkmm-1.6-1v5:amd64 (2.24.2-3) …
Selecting previously unselected package libcairomm-1.0-1v5:amd64.
Preparing to unpack …/25-libcairomm-1.0-1v5_1.12.2-3_amd64.deb …
Unpacking libcairomm-1.0-1v5:amd64 (1.12.2-3) …
Selecting previously unselected package librevenge-0.0-0:amd64.
Preparing to unpack …/26-librevenge-0.0-0_0.0.4-6ubuntu2_amd64.deb …
Unpacking librevenge-0.0-0:amd64 (0.0.4-6ubuntu2) …
Selecting previously unselected package libcdr-0.1-1:amd64.
Preparing to unpack …/27-libcdr-0.1-1_0.1.4-1build1_amd64.deb …
Unpacking libcdr-0.1-1:amd64 (0.1.4-1build1) …
Selecting previously unselected package libdbus-glib-1-2:amd64.
Preparing to unpack …/28-libdbus-glib-1-2_0.110-2_amd64.deb …
Unpacking libdbus-glib-1-2:amd64 (0.110-2) …
Selecting previously unselected package libgslcblas0:amd64.
Preparing to unpack …/29-libgslcblas0_2.4+dfsg-6_amd64.deb …
Unpacking libgslcblas0:amd64 (2.4+dfsg-6) …
Selecting previously unselected package libgsl23:amd64.
Preparing to unpack …/30-libgsl23_2.4+dfsg-6_amd64.deb …
Unpacking libgsl23:amd64 (2.4+dfsg-6) …
Selecting previously unselected package libgtk2.0-common.
Preparing to unpack …/31-libgtk2.0-common_2.24.32-1ubuntu1_all.deb …
Unpacking libgtk2.0-common (2.24.32-1ubuntu1) …
Selecting previously unselected package libgtk2.0-0:amd64.
Preparing to unpack …/32-libgtk2.0-0_2.24.32-1ubuntu1_amd64.deb …
Unpacking libgtk2.0-0:amd64 (2.24.32-1ubuntu1) …
Selecting previously unselected package libpangomm-1.4-1v5:amd64.
Preparing to unpack …/33-libpangomm-1.4-1v5_2.40.1-4_amd64.deb …
Unpacking libpangomm-1.4-1v5:amd64 (2.40.1-4) …
Selecting previously unselected package libgtkmm-2.4-1v5:amd64.
Preparing to unpack …/34-libgtkmm-2.4-1v5_1%3a2.24.5-2_amd64.deb …
Unpacking libgtkmm-2.4-1v5:amd64 (1:2.24.5-2) …
Selecting previously unselected package libgtkspell0:amd64.
Preparing to unpack …/35-libgtkspell0_2.0.16-1.2_amd64.deb …
Unpacking libgtkspell0:amd64 (2.0.16-1.2) …
Selecting previously unselected package libmagick++-6.q16-7:amd64.
Preparing to unpack
…/36-libmagick++-6.q16-7_8%3a6.9.7.4+dfsg-16ubuntu6.12_amd64.deb …
Unpacking libmagick++-6.q16-7:amd64 (8:6.9.7.4+dfsg-16ubuntu6.12) …
Selecting previously unselected package libpoppler-glib8:amd64.
Preparing to unpack …/37-libpoppler-glib8_0.62.0-2ubuntu2.12_amd64.deb …
Unpacking libpoppler-glib8:amd64 (0.62.0-2ubuntu2.12) …
Selecting previously unselected package libvisio-0.1-1:amd64.
Preparing to unpack …/38-libvisio-0.1-1_0.1.6-1build1_amd64.deb …
Unpacking libvisio-0.1-1:amd64 (0.1.6-1build1) …
Selecting previously unselected package libwpd-0.10-10:amd64.
```

```
Preparing to unpack …/39-libwpd-0.10-10_0.10.2-2_amd64.deb …
Unpacking libwpd-0.10-10:amd64 (0.10.2-2) …
Selecting previously unselected package libwpg-0.3-3:amd64.
Preparing to unpack …/40-libwpg-0.3-3_0.3.1-3_amd64.deb …
Unpacking libwpg-0.3-3:amd64 (0.3.1-3) …
Selecting previously unselected package inkscape.
Preparing to unpack …/41-inkscape_0.92.3-1_amd64.deb …
Unpacking inkscape (0.92.3-1) …
Selecting previously unselected package libdjvulibre-text.
Preparing to unpack …/42-libdjvulibre-text_3.5.27.1-8ubuntu0.4_all.deb …
Unpacking libdjvulibre-text (3.5.27.1-8ubuntu0.4) …
Selecting previously unselected package libdjvulibre21:amd64.
Preparing to unpack …/43-libdjvulibre21_3.5.27.1-8ubuntu0.4_amd64.deb …
Unpacking libdjvulibre21:amd64 (3.5.27.1-8ubuntu0.4) …
Selecting previously unselected package libgail18:amd64.
Preparing to unpack …/44-libgail18_2.24.32-1ubuntu1_amd64.deb …
Unpacking libgail18:amd64 (2.24.32-1ubuntu1) …
Selecting previously unselected package libgail-common:amd64.
Preparing to unpack …/45-libgail-common_2.24.32-1ubuntu1_amd64.deb …
Unpacking libgail-common:amd64 (2.24.32-1ubuntu1) …
Selecting previously unselected package libgtk2.0-bin.
Preparing to unpack …/46-libgtk2.0-bin_2.24.32-1ubuntu1_amd64.deb …
Unpacking libgtk2.0-bin (2.24.32-1ubuntu1) …
Selecting previously unselected package libwmf0.2-7:amd64.
Preparing to unpack …/47-libwmf0.2-7_0.2.8.4-12_amd64.deb …
Unpacking libwmf0.2-7:amd64 (0.2.8.4-12) …
Selecting previously unselected package libmagickcore-6.q16-3-extra:amd64.
Preparing to unpack
…/48-libmagickcore-6.q16-3-extra_8%3a6.9.7.4+dfsg-16ubuntu6.12_amd64.deb …
Unpacking libmagickcore-6.q16-3-extra:amd64 (8:6.9.7.4+dfsg-16ubuntu6.12) …
Selecting previously unselected package libnetpbm10.
Preparing to unpack …/49-libnetpbm10_2%3a10.0-15.3build1_amd64.deb …
Unpacking libnetpbm10 (2:10.0-15.3build1) …
Selecting previously unselected package libwmf-bin.
Preparing to unpack …/50-libwmf-bin_0.2.8.4-12_amd64.deb …
Unpacking libwmf-bin (0.2.8.4-12) …
Selecting previously unselected package netpbm.
Preparing to unpack …/51-netpbm_2%3a10.0-15.3build1_amd64.deb …
Unpacking netpbm (2:10.0-15.3build1) …
Selecting previously unselected package python-bs4.
Preparing to unpack …/52-python-bs4_4.6.0-1_all.deb …
Unpacking python-bs4 (4.6.0-1) …
Selecting previously unselected package python-pkg-resources.
Preparing to unpack …/53-python-pkg-resources_39.0.1-2_all.deb …
Unpacking python-pkg-resources (39.0.1-2) …
Selecting previously unselected package python-chardet.
Preparing to unpack …/54-python-chardet_3.0.4-1_all.deb …
Unpacking python-chardet (3.0.4-1) …
```

```
Selecting previously unselected package python-six.
Preparing to unpack …/55-python-six_1.11.0-2_all.deb …
Unpacking python-six (1.11.0-2) …
Selecting previously unselected package python-webencodings.
Preparing to unpack …/56-python-webencodings_0.5-2_all.deb …
Unpacking python-webencodings (0.5-2) …
Selecting previously unselected package python-html5lib.
Preparing to unpack …/57-python-html5lib_0.999999999-1_all.deb …
Unpacking python-html5lib (0.999999999-1) …
Selecting previously unselected package python-lxml:amd64.
Preparing to unpack …/58-python-lxml_4.2.1-1ubuntu0.6_amd64.deb …
Unpacking python-lxml:amd64 (4.2.1-1ubuntu0.6) …
Selecting previously unselected package python3-six.
Preparing to unpack …/59-python3-six_1.11.0-2_all.deb …
Unpacking python3-six (1.11.0-2) …
Selecting previously unselected package python3-pkg-resources.
Preparing to unpack …/60-python3-pkg-resources_39.0.1-2_all.deb …
Unpacking python3-pkg-resources (39.0.1-2) …
Selecting previously unselected package python3-scour.
Preparing to unpack …/61-python3-scour_0.36-2_all.deb …
Unpacking python3-scour (0.36-2) …
Selecting previously unselected package scour.
Preparing to unpack …/62-scour_0.36-2_all.deb …
Unpacking scour (0.36-2) …
Selecting previously unselected package python-scour.
Preparing to unpack …/63-python-scour_0.36-2_all.deb …
Unpacking python-scour (0.36-2) …
Selecting previously unselected package transfig.
Preparing to unpack …/64-transfig_1%3a3.2.6a-6ubuntu1_all.deb …
Unpacking transfig (1:3.2.6a-6ubuntu1) …
Setting up imagemagick-6-common (8:6.9.7.4+dfsg-16ubuntu6.12) …
Setting up libpoppler-glib8:amd64 (0.62.0-2ubuntu2.12) …
Setting up libgtk2.0-common (2.24.32-1ubuntu1) …
Setting up gsfonts (1:8.11+urwcyr1.0.7~pre44-4.4) …
Setting up libhunspell-1.6-0:amd64 (1.6.2-1) …
Setting up libaspell15:amd64 (0.60.7~20110707-4ubuntu0.2) …
Setting up libdjvulibre-text (3.5.27.1-8ubuntu0.4) …
Setting up libdbus-glib-1-2:amd64 (0.110-2) …
Setting up libnetpbm10 (2:10.0-15.3build1) …
Setting up emacsen-common (2.0.8) …
Setting up python3-six (1.11.0-2) …
Setting up ghostscript (9.26~dfsg+0-0ubuntu0.18.04.15) …
Setting up python3-pkg-resources (39.0.1-2) …
Setting up python-pkg-resources (39.0.1-2) …
Setting up librevenge-0.0-0:amd64 (0.0.4-6ubuntu2) …
Setting up gawk (1:4.1.4+dfsg-1build1) …
Setting up liblqr-1-0:amd64 (0.4.2-2.1) …
Setting up python-six (1.11.0-2) …
```

```
Setting up python-bs4 (4.6.0-1) …
Setting up libtext-iconv-perl (1.7-5build6) …
Setting up python-lxml:amd64 (4.2.1-1ubuntu0.6) …
Setting up libcdr-0.1-1:amd64 (0.1.4-1build1) …
Setting up netpbm (2:10.0-15.3build1) …
Setting up libgslcblas0:amd64 (2.4+dfsg-6) …
Setting up libwmf0.2-7:amd64 (0.2.8.4-12) …
Setting up libsigc++-2.0-0v5:amd64 (2.10.0-2) …
Setting up python-webencodings (0.5-2) …
Setting up libcairomm-1.0-1v5:amd64 (1.12.2-3) …
Setting up libgtk2.0-0:amd64 (2.24.32-1ubuntu1) …
Setting up python-chardet (3.0.4-1) …
Setting up libmagickcore-6.q16-3:amd64 (8:6.9.7.4+dfsg-16ubuntu6.12) …
Setting up libgail18:amd64 (2.24.32-1ubuntu1) …
Setting up libgsl23:amd64 (2.4+dfsg-6) …
Setting up libdjvulibre21:amd64 (3.5.27.1-8ubuntu0.4) …
Setting up dictionaries-common (1.27.2) …
Setting up libvisio-0.1-1:amd64 (0.1.6-1build1) …
Setting up libwpd-0.10-10:amd64 (0.10.2-2) …
Setting up fig2dev (1:3.2.6a-6ubuntu1) …
Setting up python3-scour (0.36-2) …
Setting up scour (0.36-2) …
Setting up libgail-common:amd64 (2.24.32-1ubuntu1) …
Setting up libwmf-bin (0.2.8.4-12) …
Setting up transfig (1:3.2.6a-6ubuntu1) …
Setting up python-scour (0.36-2) …
Setting up libmagickwand-6.q16-3:amd64 (8:6.9.7.4+dfsg-16ubuntu6.12) …
Setting up libglibmm-2.4-1v5:amd64 (2.56.0-1) …
Setting up libwpg-0.3-3:amd64 (0.3.1-3) …
Setting up python-html5lib (0.999999999-1) …
Setting up aspell (0.60.7~20110707-4ubuntu0.2) …
Setting up hunspell-en-us (1:2017.08.24) …
Setting up libatkmm-1.6-1v5:amd64 (2.24.2-3) …
Setting up libgtk2.0-bin (2.24.32-1ubuntu1) …
Setting up libenchant1c2a:amd64 (1.6.0-11.1) …
Setting up libimage-magick-q16-perl (8:6.9.7.4+dfsg-16ubuntu6.12) …
Setting up imagemagick-6.q16 (8:6.9.7.4+dfsg-16ubuntu6.12) …
update-alternatives: using /usr/bin/compare-im6.q16 to provide /usr/bin/compare
(compare) in auto mode
update-alternatives: using /usr/bin/compare-im6.q16 to provide /usr/bin/compare-
im6 (compare-im6) in auto mode
update-alternatives: using /usr/bin/animate-im6.q16 to provide /usr/bin/animate
(animate) in auto mode
update-alternatives: using /usr/bin/animate-im6.q16 to provide /usr/bin/animate-
im6 (animate-im6) in auto mode
update-alternatives: using /usr/bin/convert-im6.q16 to provide /usr/bin/convert
(convert) in auto mode
update-alternatives: using /usr/bin/convert-im6.q16 to provide /usr/bin/convert-
```

```
im6 (convert-im6) in auto mode
update-alternatives: using /usr/bin/composite-im6.q16 to provide
/usr/bin/composite (composite) in auto mode
update-alternatives: using /usr/bin/composite-im6.q16 to provide
/usr/bin/composite-im6 (composite-im6) in auto mode
update-alternatives: using /usr/bin/conjure-im6.q16 to provide /usr/bin/conjure
(conjure) in auto mode
update-alternatives: using /usr/bin/conjure-im6.q16 to provide /usr/bin/conjure-
im6 (conjure-im6) in auto mode
update-alternatives: using /usr/bin/import-im6.q16 to provide /usr/bin/import
(import) in auto mode
update-alternatives: using /usr/bin/import-im6.q16 to provide /usr/bin/import-
im6 (import-im6) in auto mode
update-alternatives: using /usr/bin/identify-im6.q16 to provide
/usr/bin/identify (identify) in auto mode
update-alternatives: using /usr/bin/identify-im6.q16 to provide
/usr/bin/identify-im6 (identify-im6) in auto mode
update-alternatives: using /usr/bin/stream-im6.q16 to provide /usr/bin/stream
(stream) in auto mode
update-alternatives: using /usr/bin/stream-im6.q16 to provide /usr/bin/stream-
im6 (stream-im6) in auto mode
update-alternatives: using /usr/bin/display-im6.q16 to provide /usr/bin/display
(display) in auto mode
update-alternatives: using /usr/bin/display-im6.q16 to provide /usr/bin/display-
im6 (display-im6) in auto mode
update-alternatives: using /usr/bin/montage-im6.q16 to provide /usr/bin/montage
(montage) in auto mode
update-alternatives: using /usr/bin/montage-im6.q16 to provide /usr/bin/montage-
im6 (montage-im6) in auto mode
update-alternatives: using /usr/bin/mogrify-im6.q16 to provide /usr/bin/mogrify
(mogrify) in auto mode
update-alternatives: using /usr/bin/mogrify-im6.q16 to provide /usr/bin/mogrify-
im6 (mogrify-im6) in auto mode
Setting up libpangomm-1.4-1v5:amd64 (2.40.1-4) …
Setting up libgtkmm-2.4-1v5:amd64 (1:2.24.5-2) …
Setting up libgtkspell0:amd64 (2.0.16-1.2) …
Setting up libmagick++-6.q16-7:amd64 (8:6.9.7.4+dfsg-16ubuntu6.12) …
Setting up libmagickcore-6.q16-3-extra:amd64 (8:6.9.7.4+dfsg-16ubuntu6.12) …
Setting up aspell-en (2017.08.24-0-0.1) …
Setting up imagemagick (8:6.9.7.4+dfsg-16ubuntu6.12) …
Setting up enchant (1.6.0-11.1) …
Setting up inkscape (0.92.3-1) …
Setting up libimage-magick-perl (8:6.9.7.4+dfsg-16ubuntu6.12) …
Processing triggers for libc-bin (2.27-3ubuntu1.3) …
/sbin/ldconfig.real: /usr/local/lib/python3.7/dist-
packages/ideep4py/lib/libmkldnn.so.0 is not a symbolic link

Processing triggers for man-db (2.8.3-2ubuntu0.1) …
```

```
Processing triggers for hicolor-icon-theme (0.17-2) …
Processing triggers for fontconfig (2.12.6-0ubuntu2) …
Processing triggers for mime-support (3.60ubuntu1) …
Processing triggers for dictionaries-common (1.27.2) …
aspell-autobuildhash: processing: en [en-common].
aspell-autobuildhash: processing: en [en-variant_0].
aspell-autobuildhash: processing: en [en-variant_1].
aspell-autobuildhash: processing: en [en-variant_2].
aspell-autobuildhash: processing: en [en-w_accents-only].
aspell-autobuildhash: processing: en [en-wo_accents-only].
aspell-autobuildhash: processing: en [en_AU-variant_0].
aspell-autobuildhash: processing: en [en_AU-variant_1].
aspell-autobuildhash: processing: en [en_AU-w_accents-only].
aspell-autobuildhash: processing: en [en_AU-wo_accents-only].
aspell-autobuildhash: processing: en [en_CA-variant_0].
aspell-autobuildhash: processing: en [en_CA-variant_1].
aspell-autobuildhash: processing: en [en_CA-w_accents-only].
aspell-autobuildhash: processing: en [en_CA-wo_accents-only].
aspell-autobuildhash: processing: en [en_GB-ise-w_accents-only].
aspell-autobuildhash: processing: en [en_GB-ise-wo_accents-only].
aspell-autobuildhash: processing: en [en_GB-ize-w_accents-only].
aspell-autobuildhash: processing: en [en_GB-ize-wo_accents-only].
aspell-autobuildhash: processing: en [en_GB-variant_0].
aspell-autobuildhash: processing: en [en_GB-variant_1].
aspell-autobuildhash: processing: en [en_US-w_accents-only].
aspell-autobuildhash: processing: en [en_US-wo_accents-only].
'universe' distribution component is already enabled for all sources.
 The Inkscape Stable PPA is intended to provide the current supported release of
Inkscape.
 More info: https://launchpad.net/~inkscape.dev/+archive/ubuntu/stable
Press [ENTER] to continue or Ctrl-c to cancel adding it.

Hit:1 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ InRelease
Hit:2 http://security.ubuntu.com/ubuntu bionic-security InRelease
Ign:3 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64
InRelease
Hit:4 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic InRelease
Hit:5 http://archive.ubuntu.com/ubuntu bionic InRelease
Ign:6 https://developer.download.nvidia.com/compute/machine-
learning/repos/ubuntu1804/x86_64  InRelease
Hit:7 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64
Release
Hit:8 https://developer.download.nvidia.com/compute/machine-
learning/repos/ubuntu1804/x86_64  Release
Hit:9 http://archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:10 http://ppa.launchpad.net/cran/libgit2/ubuntu bionic InRelease
Hit:11 http://archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:12 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic InRelease
```

```
Hit:13 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic InRelease
Get:14 http://ppa.launchpad.net/inkscape.dev/stable/ubuntu bionic InRelease
[15.4 kB]
Get:17 http://ppa.launchpad.net/inkscape.dev/stable/ubuntu bionic/main amd64
Packages [960 B]
Fetched 16.3 kB in 3s (5,617 B/s)
Reading package lists… Done
Hit:1 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ InRelease
Ign:2 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64
InRelease
Hit:3 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic InRelease
Hit:4 http://archive.ubuntu.com/ubuntu bionic InRelease
Hit:5 http://security.ubuntu.com/ubuntu bionic-security InRelease
Ign:6 https://developer.download.nvidia.com/compute/machine-
learning/repos/ubuntu1804/x86_64  InRelease
Hit:7 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64
Release
Hit:8 http://archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:9 https://developer.download.nvidia.com/compute/machine-
learning/repos/ubuntu1804/x86_64  Release
Hit:10 http://ppa.launchpad.net/cran/libgit2/ubuntu bionic InRelease
Hit:11 http://archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:12 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic InRelease
Hit:13 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic InRelease
Hit:14 http://ppa.launchpad.net/inkscape.dev/stable/ubuntu bionic InRelease
Reading package lists… Done
Reading package lists… Done
Building dependency tree
Reading state information… Done
The following packages were automatically installed and are no longer required:
  fig2dev libdbus-glib-1-2 libgtkmm-2.4-1v5 libgtkspell0 python-bs4
  python-chardet python-html5lib python-lxml python-pkg-resources python-scour
  python-six python-webencodings scour transfig
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  adwaita-icon-theme-full libgdl-3-5 libgdl-3-common libgtkmm-3.0-1v5
  libgtkspell3-3-0 python3-bs4 python3-chardet python3-html5lib python3-lxml
  python3-numpy python3-webencodings
Suggested packages:
  dia | dia-gnome libsvg-perl libxml-xql-perl python3-serial pstoedit
  python3-genshi python3-lxml-dbg python-lxml-doc python-numpy-doc
  python3-nose python3-numpy-dbg
The following NEW packages will be installed:
  adwaita-icon-theme-full libgdl-3-5 libgdl-3-common libgtkmm-3.0-1v5
  libgtkspell3-3-0 python3-bs4 python3-chardet python3-html5lib python3-lxml
  python3-numpy python3-webencodings
The following packages will be upgraded:
  inkscape
```

```
1 upgraded, 11 newly installed, 0 to remove and 79 not upgraded.
Need to get 29.4 MB of archives.
After this operation, 80.5 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/universe amd64 adwaita-icon-theme-
full all 3.28.0-1ubuntu1 [7,999 kB]
Get:2 http://ppa.launchpad.net/inkscape.dev/stable/ubuntu bionic/main amd64
inkscape amd64 1.0.2+r75+1~ubuntu18.04.1 [17.0 MB]
Get:3 http://archive.ubuntu.com/ubuntu bionic/universe amd64 libgdl-3-common all
3.28.0-1 [128 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic/universe amd64 libgdl-3-5 amd64
3.28.0-1 [75.8 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic/main amd64 libgtkmm-3.0-1v5 amd64
3.22.2-2 [850 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/universe amd64 libgtkspell3-3-0
amd64 3.0.9-2 [28.9 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/main amd64 python3-bs4 all 4.6.0-1
[67.8 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/main amd64 python3-chardet all
3.0.4-1 [80.3 kB]
Get:9 http://archive.ubuntu.com/ubuntu bionic/main amd64 python3-webencodings
all 0.5-2 [10.4 kB]
Get:10 http://archive.ubuntu.com/ubuntu bionic/main amd64 python3-html5lib all
0.999999999-1 [81.9 kB]
Get:11 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 python3-lxml
amd64 4.2.1-1ubuntu0.6 [1,098 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic/main amd64 python3-numpy amd64
1:1.13.3-2ubuntu1 [1,943 kB]
Fetched 29.4 MB in 1s (26.1 MB/s)
Selecting previously unselected package adwaita-icon-theme-full.
(Reading database … 185883 files and directories currently installed.)
Preparing to unpack …/00-adwaita-icon-theme-full_3.28.0-1ubuntu1_all.deb …
Unpacking adwaita-icon-theme-full (3.28.0-1ubuntu1) …
Selecting previously unselected package libgdl-3-common.
Preparing to unpack …/01-libgdl-3-common_3.28.0-1_all.deb …
Unpacking libgdl-3-common (3.28.0-1) …
Selecting previously unselected package libgdl-3-5:amd64.
Preparing to unpack …/02-libgdl-3-5_3.28.0-1_amd64.deb …
Unpacking libgdl-3-5:amd64 (3.28.0-1) …
Selecting previously unselected package libgtkmm-3.0-1v5:amd64.
Preparing to unpack …/03-libgtkmm-3.0-1v5_3.22.2-2_amd64.deb …
Unpacking libgtkmm-3.0-1v5:amd64 (3.22.2-2) …
Selecting previously unselected package libgtkspell3-3-0:amd64.
Preparing to unpack …/04-libgtkspell3-3-0_3.0.9-2_amd64.deb …
Unpacking libgtkspell3-3-0:amd64 (3.0.9-2) …
Preparing to unpack …/05-inkscape_1.0.2+r75+1~ubuntu18.04.1_amd64.deb …
Unpacking inkscape (1.0.2+r75+1~ubuntu18.04.1) over (0.92.3-1) …
Selecting previously unselected package python3-bs4.
Preparing to unpack …/06-python3-bs4_4.6.0-1_all.deb …
```

```
Unpacking python3-bs4 (4.6.0-1) …
Selecting previously unselected package python3-chardet.
Preparing to unpack …/07-python3-chardet_3.0.4-1_all.deb …
Unpacking python3-chardet (3.0.4-1) …
Selecting previously unselected package python3-webencodings.
Preparing to unpack …/08-python3-webencodings_0.5-2_all.deb …
Unpacking python3-webencodings (0.5-2) …
Selecting previously unselected package python3-html5lib.
Preparing to unpack …/09-python3-html5lib_0.999999999-1_all.deb …
Unpacking python3-html5lib (0.999999999-1) …
Selecting previously unselected package python3-lxml:amd64.
Preparing to unpack …/10-python3-lxml_4.2.1-1ubuntu0.6_amd64.deb …
Unpacking python3-lxml:amd64 (4.2.1-1ubuntu0.6) …
Selecting previously unselected package python3-numpy.
Preparing to unpack …/11-python3-numpy_1%3a1.13.3-2ubuntu1_amd64.deb …
Unpacking python3-numpy (1:1.13.3-2ubuntu1) …
Setting up python3-webencodings (0.5-2) …
Setting up python3-lxml:amd64 (4.2.1-1ubuntu0.6) …
Setting up python3-numpy (1:1.13.3-2ubuntu1) …
Setting up libgtkmm-3.0-1v5:amd64 (3.22.2-2) …
Setting up python3-bs4 (4.6.0-1) …
Setting up libgdl-3-common (3.28.0-1) …
Setting up python3-chardet (3.0.4-1) …
Setting up python3-html5lib (0.999999999-1) …
Setting up libgtkspell3-3-0:amd64 (3.0.9-2) …
Setting up libgdl-3-5:amd64 (3.28.0-1) …
Setting up inkscape (1.0.2+r75+1~ubuntu18.04.1) …
Setting up adwaita-icon-theme-full (3.28.0-1ubuntu1) …
Processing triggers for libc-bin (2.27-3ubuntu1.3) …
/sbin/ldconfig.real: /usr/local/lib/python3.7/dist-
packages/ideep4py/lib/libmkldnn.so.0 is not a symbolic link

Processing triggers for man-db (2.8.3-2ubuntu0.1) …
Processing triggers for hicolor-icon-theme (0.17-2) …
Processing triggers for mime-support (3.60ubuntu1) …
```

[107]:
```
# %%capture
import re

ASSIGNMENT_NAME = 'DAT340 - Assignment 1'
pdf_dir = os.path.join(os.path.abspath(''), 'drive', 'MyDrive')
pdf_dir = os.path.join(pdf_dir, 'Colab Notebooks', 'dat340', ASSIGNMENT_ID)
pdf_filename = re.escape(os.path.join(pdf_dir, ASSIGNMENT_NAME)) + '.ipynb'

!jupyter nbconvert --to pdf --TemplateExporter.exclude_input=False $pdf_filename
```

```
[NbConvertApp] WARNING | pattern '/content/drive/MyDrive/Colab
Notebooks/dat340/DAT340 - Assignment 1/DAT340 - Assignment 1.ipynb' matched no
```

```
files
This application is used to convert notebook files (*.ipynb)
        to various other formats.

        WARNING: THE COMMANDLINE INTERFACE MAY CHANGE IN FUTURE RELEASES.

Options
=======
The options below are convenience aliases to configurable class-options,
as listed in the "Equivalent to" description-line of the aliases.
To see all configurable class-options for some <cmd>, use:
    <cmd> --help-all

--debug
    set log level to logging.DEBUG (maximize logging output)
    Equivalent to: [--Application.log_level=10]
--show-config
    Show the application's configuration (human-readable format)
    Equivalent to: [--Application.show_config=True]
--show-config-json
    Show the application's configuration (json format)
    Equivalent to: [--Application.show_config_json=True]
--generate-config
    generate default config file
    Equivalent to: [--JupyterApp.generate_config=True]
-y
    Answer yes to any questions instead of prompting.
    Equivalent to: [--JupyterApp.answer_yes=True]
--execute
    Execute the notebook prior to export.
    Equivalent to: [--ExecutePreprocessor.enabled=True]
--allow-errors
    Continue notebook execution even if one of the cells throws an error and
include the error message in the cell output (the default behaviour is to abort
conversion). This flag is only relevant if '--execute' was specified, too.
    Equivalent to: [--ExecutePreprocessor.allow_errors=True]
--stdin
    read a single notebook file from stdin. Write the resulting notebook with
default basename 'notebook.*'
    Equivalent to: [--NbConvertApp.from_stdin=True]
--stdout
    Write notebook output to stdout instead of files.
    Equivalent to: [--NbConvertApp.writer_class=StdoutWriter]
--inplace
    Run nbconvert in place, overwriting the existing notebook (only
            relevant when converting to notebook format)
    Equivalent to: [--NbConvertApp.use_output_suffix=False
--NbConvertApp.export_format=notebook --FilesWriter.build_directory=]
```

```
--clear-output
    Clear output of current file and save in place,
            overwriting the existing notebook.
    Equivalent to: [--NbConvertApp.use_output_suffix=False
--NbConvertApp.export_format=notebook --FilesWriter.build_directory=
--ClearOutputPreprocessor.enabled=True]
--no-prompt
    Exclude input and output prompts from converted document.
    Equivalent to: [--TemplateExporter.exclude_input_prompt=True
--TemplateExporter.exclude_output_prompt=True]
--no-input
    Exclude input cells and output prompts from converted document.
            This mode is ideal for generating code-free reports.
    Equivalent to: [--TemplateExporter.exclude_output_prompt=True
--TemplateExporter.exclude_input=True]
--log-level=<Enum>
    Set the log level by value or name.
    Choices: any of [0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR',
'CRITICAL']
    Default: 30
    Equivalent to: [--Application.log_level]
--config=<Unicode>
    Full path of a config file.
    Default: ''
    Equivalent to: [--JupyterApp.config_file]
--to=<Unicode>
    The export format to be used, either one of the built-in formats
            ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook',
'pdf', 'python', 'rst', 'script', 'slides']
            or a dotted object name that represents the import path for an
            `Exporter` class
    Default: 'html'
    Equivalent to: [--NbConvertApp.export_format]
--template=<Unicode>
    Name of the template file to use
    Default: ''
    Equivalent to: [--TemplateExporter.template_file]
--writer=<DottedObjectName>
    Writer class used to write the
                                        results of the conversion
    Default: 'FilesWriter'
    Equivalent to: [--NbConvertApp.writer_class]
--post=<DottedOrNone>
    PostProcessor class used to write the
                                        results of the conversion
    Default: ''
    Equivalent to: [--NbConvertApp.postprocessor_class]
--output=<Unicode>
```

```
    overwrite base name use for output files.
                can only be used when converting one notebook at a time.
    Default: ''
    Equivalent to: [--NbConvertApp.output_base]
--output-dir=<Unicode>
    Directory to write output(s) to. Defaults
                                    to output to the directory of each notebook.
To recover
                                    previous default behaviour (outputting to the
current
                                    working directory) use . as the flag value.
    Default: ''
    Equivalent to: [--FilesWriter.build_directory]
--reveal-prefix=<Unicode>
    The URL prefix for reveal.js (version 3.x).
            This defaults to the reveal CDN, but can be any url pointing to a
copy
            of reveal.js.
            For speaker notes to work, this must be a relative path to a local
            copy of reveal.js: e.g., "reveal.js".
            If a relative path is given, it must be a subdirectory of the
            current directory (from which the server is run).
            See the usage documentation
            (https://nbconvert.readthedocs.io/en/latest/usage.html#reveal-js-
html-slideshow)
            for more details.
    Default: ''
    Equivalent to: [--SlidesExporter.reveal_url_prefix]
--nbformat=<Enum>
    The nbformat version to write.
            Use this to downgrade notebooks.
    Choices: any of [1, 2, 3, 4]
    Default: 4
    Equivalent to: [--NotebookExporter.nbformat_version]

Examples
--------

    The simplest way to use nbconvert is

            > jupyter nbconvert mynotebook.ipynb

            which will convert mynotebook.ipynb to the default format (probably
HTML).

            You can specify the export format with `--to`.
            Options include ['asciidoc', 'custom', 'html', 'latex', 'markdown',
'notebook', 'pdf', 'python', 'rst', 'script', 'slides'].
```

```
            > jupyter nbconvert --to latex mynotebook.ipynb

            Both HTML and LaTeX support multiple output templates. LaTeX
includes
            'base', 'article' and 'report'.  HTML includes 'basic' and 'full'.
You
            can specify the flavor of the format used.

            > jupyter nbconvert --to html --template basic mynotebook.ipynb

            You can also pipe the output to stdout, rather than a file

            > jupyter nbconvert mynotebook.ipynb --stdout

            PDF is generated via latex

            > jupyter nbconvert mynotebook.ipynb --to pdf

            You can get (and serve) a Reveal.js-powered slideshow

            > jupyter nbconvert myslides.ipynb --to slides --post serve

            Multiple notebooks can be given at the command line in a couple of
            different ways:

            > jupyter nbconvert notebook*.ipynb
            > jupyter nbconvert notebook1.ipynb notebook2.ipynb

            or you can specify the notebooks list in a config file, containing::

                c.NbConvertApp.notebooks = ["my_notebook.ipynb"]

            > jupyter nbconvert --config mycfg.py

To see all available configurables, use `--help-all`.
```

[ ]: