Mudcard

- no questions

Supervised ML algorithms

By the end of this week, you will be able to

- Summarize how decision trees, random forests, and support vector machines work
- Describe how the predictions of these techniques behave in classification and regression
- Describe which hyper-parameters should be tuned

A decision tree in regression

In [1]:
```
import numpy as np
from sklearn.ensemble import RandomForestRegressor
np.random.seed(10)
def true_fun(X):
    return np.cos(1.5 * np.pi * X)

n_samples = 30

X = np.random.rand(n_samples)
y = true_fun(X) + np.random.randn(n_samples) * 0.1

X_new = np.linspace(0, 1, 1000)

reg = RandomForestRegressor(n_estimators=1,max_depth=1)
reg.fit(X[:, np.newaxis],y)
y_new = reg.predict(X_new[:, np.newaxis])
```

In [ ]:
```
help(RandomForestRegressor)
```

In [2]:
```
##############################################################
# HUGE thanks to Drew Solomon and Yifei Song (DSI alumni)
# for preparing the visualizations in this lecture!
##############################################################
# check out helper_functions.ipynb for more details
%run ./helper_functions.ipynb

hyperparameters = {
    'n_estimators': [1, 3, 10, 30],
    'max_depth': [1, 2, 3, 10, 30]
}

vis(X, y, RandomForestRegressor, hyperparameters, X_new)
```

```
interactive(children=(SelectionSlider(description='n_estimators', options=(1, 3, 10, 30), value=1), SelectionS…
```

How to avoid overfitting with random forests?

- tune some (or all) of following hyperparameters:
  - max_depth
  - max_features
- With sklearn random forests, **do not tune n_estimators**!
  - the larger this value is, the better the forest will be
  - set n_estimators to maybe 100 while tuning hyperparameters
  - increase it if necessary once the best hyperparameters are found

ML algo	suitable for large datasets?	behaviour wrt outliers	non-linear?	params to tune	smooth predictions	easy to interpret?
linear regression	yes	linear extrapolation	no	l1 and/or l2 reg	yes	yes
logistic regression	yes	scales with distance from the decision boundary	no	l1 and/or l2 reg	yes	yes
random forest regression	so so	constant	yes	max_features, max_depth	no	so so
random forest classification	tbd	tbd	tbd	tbd	tbd	tbd
SVM rbf regression	tbd	tbd	tbd	tbd	tbd	tbd
SVM rbf classification	tbd	tbd	tbd	tbd	tbd	tbd

A random forest in classification

In [3]:
```
from sklearn.datasets import make_moons
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import ParameterGrid

# create the data
X,y = make_moons(noise=0.2, random_state=1,n_samples=200)
# set the hyperparameters
clf = RandomForestClassifier(n_estimators=1,max_depth=3,random_state=0)
# fit the model
clf.fit(X,y)
# predict new data
#y_new = clf.predict(X_new)
# predict probabilities
#y_new = clf.predict_proba(X_new)
```

Out[3]:
```
▾                 RandomForestClassifier
RandomForestClassifier(max_depth=3, n_estimators=1, random_state=0)
```

In [ ]:
```
help(RandomForestClassifier)
```

In [4]:
```
# initialize RandomForestClassifier
ML_algo = RandomForestClassifier(random_state=42)

# set RF parameter grid
hyperparameters = {
    'n_estimators': [1, 3, 10, 30],
    'max_depth': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
}

plot_clf_contour(hyperparameters, X, y)
```

```
interactive(children=(SelectionSlider(description='n_estimators', options=(1, 3, 10, 30), value=1), SelectionS…
Out[4]:  FigureWidget({
             'data': [{'colorbar': {'title': {'text': 'predicted probability'}},
                      'colorscale': [[0.0, 'rgb(103,0,31)'], [0.1, 'rgb(178,24,43)'],
                                    [0.2, 'rgb(214,96,77)'], [0.3, 'rgb(244,165,130)'],
                                    [0.4, 'rgb(253,219,199)'], [0.5, 'rgb(247,247,247)'],
                                    [0.6, 'rgb(209,229,240)'], [0.7, 'rgb(146,197,222)'],
                                    [0.8, 'rgb(67,147,195)'], [0.9, 'rgb(33,102,172)'],
                                    [1.0, 'rgb(5,48,97)']],
                      'contours': {'end': 1, 'size': 0.05, 'start': 0},
                      'type': 'contour',
                      'uid': 'ef117a7d-d955-4461-872b-1e9e83fe48a3',
                      'x': array([-1.64483117, -1.62483117, -1.60483117, ...,  2.75516883,  2.77516883,
                                   2.79516883]),
                      'y': array([-1.44154690e+00, -1.42154690e+00, -1.40154690e+00, -1.38154690e+00,
                                  -1.36154690e+00, -1.34154690e+00, -1.32154690e+00, -1.30154690e+00,
                                  -1.28154690e+00, -1.26154690e+00, -1.24154690e+00, -1.22154690e+00,
                                  -1.20154690e+00, -1.18154690e+00, -1.16154690e+00, -1.14154690e+00,
                                  -1.12154690e+00, -1.10154690e+00, -1.08154690e+00, -1.06154690e+00,
                                  -1.04154690e+00, -1.02154690e+00, -1.00154690e+00, -9.81546901e-01,
                                  -9.61546901e-01, -9.41546901e-01, -9.21546901e-01, -9.01546901e-01,
                                  -8.81546901e-01, -8.61546901e-01, -8.41546901e-01, -8.21546901e-01,
                                  -8.01546901e-01, -7.81546901e-01, -7.61546901e-01, -7.41546901e-01,
                                  -7.21546901e-01, -7.01546901e-01, -6.81546901e-01, -6.61546901e-01,
                                  -6.41546901e-01, -6.21546901e-01, -6.01546901e-01, -5.81546901e-01,
                                  -5.61546901e-01, -5.41546901e-01, -5.21546901e-01, -5.01546901e-01,
                                  -4.81546901e-01, -4.61546901e-01, -4.41546901e-01, -4.21546901e-01,
                                  -4.01546901e-01, -3.81546901e-01, -3.61546901e-01, -3.41546901e-01,
                                  -3.21546901e-01, -3.01546901e-01, -2.81546901e-01, -2.61546901e-01,
                                  -2.41546901e-01, -2.21546901e-01, -2.01546901e-01, -1.81546901e-01,
                                  -1.61546901e-01, -1.41546901e-01, -1.21546901e-01, -1.01546901e-01,
                                  -8.15469010e-02, -6.15469010e-02, -4.15469010e-02, -2.15469010e-02,
                                  -1.54690100e-03,  1.84530990e-02,  3.84530990e-02,  5.84530990e-02,
                                   7.84530990e-02,  9.84530990e-02,  1.18453099e-01,  1.38453099e-01,
                                   1.58453099e-01,  1.78453099e-01,  1.98453099e-01,  2.18453099e-01,
                                   2.38453099e-01,  2.58453099e-01,  2.78453099e-01,  2.98453099e-01,
                                   3.18453099e-01,  3.38453099e-01,  3.58453099e-01,  3.78453099e-01,
                                   3.98453099e-01,  4.18453099e-01,  4.38453099e-01,  4.58453099e-01,
                                   4.78453099e-01,  4.98453099e-01,  5.18453099e-01,  5.38453099e-01,
                                   5.58453099e-01,  5.78453099e-01,  5.98453099e-01,  6.18453099e-01,
                                   6.38453099e-01,  6.58453099e-01,  6.78453099e-01,  6.98453099e-01,
                                   7.18453099e-01,  7.38453099e-01,  7.58453099e-01,  7.78453099e-01,
                                   7.98453099e-01,  8.18453099e-01,  8.38453099e-01,  8.58453099e-01,
                                   8.78453099e-01,  8.98453099e-01,  9.18453099e-01,  9.38453099e-01,
                                   9.58453099e-01,  9.78453099e-01,  9.98453099e-01,  1.01845310e+00,
                                   1.03845310e+00,  1.05845310e+00,  1.07845310e+00,  1.09845310e+00,
                                   1.11845310e+00,  1.13845310e+00,  1.15845310e+00,  1.17845310e+00,
                                   1.19845310e+00,  1.21845310e+00,  1.23845310e+00,  1.25845310e+00,
                                   1.27845310e+00,  1.29845310e+00,  1.31845310e+00,  1.33845310e+00,
                                   1.35845310e+00,  1.37845310e+00,  1.39845310e+00,  1.41845310e+00,
                                   1.43845310e+00,  1.45845310e+00,  1.47845310e+00,  1.49845310e+00,
                                   1.51845310e+00,  1.53845310e+00,  1.55845310e+00,  1.57845310e+00,
                                   1.59845310e+00,  1.61845310e+00,  1.63845310e+00,  1.65845310e+00,
                                   1.67845310e+00,  1.69845310e+00,  1.71845310e+00,  1.73845310e+00,
                                   1.75845310e+00,  1.77845310e+00,  1.79845310e+00,  1.81845310e+00,
                                   1.83845310e+00,  1.85845310e+00,  1.87845310e+00]),
                      'z': array([[0.86597938, 0.86597938, 0.86597938, ..., 0.86597938, 0.86597938,
                                   0.86597938],
                                  [0.86597938, 0.86597938, 0.86597938, ..., 0.86597938, 0.86597938,
                                   0.86597938],
                                  [0.86597938, 0.86597938, 0.86597938, ..., 0.86597938, 0.86597938,
                                   0.86597938],
                                  ...,
                                  [0.16504854, 0.16504854, 0.16504854, ..., 0.16504854, 0.16504854,
                                   0.16504854],
                                  [0.16504854, 0.16504854, 0.16504854, ..., 0.16504854, 0.16504854,
                                   0.16504854],
                                  [0.16504854, 0.16504854, 0.16504854, ..., 0.16504854, 0.16504854,
                                   0.16504854]])},
                     {'marker': {'color': array([0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0,
                                                 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0,
                                                 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0,
                                                 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1,
                                                 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0,
                                                 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1,
                                                 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1,
                                                 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1,
                                                 1, 1, 0, 1, 1, 0, 1, 0]),
                                 'colorscale': [[0, 'rgb(255,0,0)'], [1, 'rgb(0,0,255)']],
                                 'line': {'width': 1},
                                 'size': 8},
                      'mode': 'markers',
                      'type': 'scatter',
                      'uid': '16bb4045-d377-47bf-be8d-25dbd1c4cee0',
                      'x': array([-0.31410929,  0.39443922,  0.48606504, -0.12805768,  1.73330291,
                                   2.08500896, -1.08258853,  1.46899598,  0.54077095,  2.04624573,
                                   1.26754011,  0.84766004,  1.12118285, -0.78670168,  0.00903277,
                                   0.44164301,  1.43162307,  0.42529121,  0.47635606, -0.21866384,
                 -0.08003084,  0.89546191,  0.22021464, -0.86869386,  0.720033  ,
                  1.13779598,  0.8517701 ,  0.78598197,  1.9446092 ,  0.57102512,
                 -0.0874615 ,  1.36354658,  2.01180011,  1.74821345,  2.02576178,
                 -1.01470657,  1.66026241, -0.95942153, -0.69886079,  1.69148155,
                  0.1386857 , -0.08753717,  2.30359131,  0.89058213, -0.18985218,
                  0.87929191, -0.18902626,  0.62843809,  1.98598879,  0.42815849,
                 -0.76592918, -0.1766055 , -0.16502413,  0.09064031,  0.58215419,
                  1.59947816,  1.46796344,  0.07718396,  2.05560682,  0.45951857,
                  0.14430234, -0.46286897, -0.75771806,  0.31091269,  0.44194998,
                  0.95163996,  0.4400647 ,  0.087738  , -0.19010962, -0.91944812,
                 -0.04820292, -0.77905887,  1.82882388,  1.24333486,  1.85387971,
                 -0.04940048,  0.8753758 , -0.12773072, -0.97433768, -0.04547499,
                  1.30029351, -1.14483117, -1.00043214,  0.53227789,  2.18759633,
                  1.18059461, -0.60654542,  0.28976789,  2.0755619 ,  0.36715992,
                 -0.70800052,  0.50273757,  1.65490162,  1.84352533,  1.30164148,
                  0.22470731,  1.94238198,  2.03105931,  0.82243733,  1.85498244,
                 -0.85944022,  0.48924579, -1.09621553,  1.09430083,  0.9242366 ,
                 -0.59077915, -0.78977567,  0.95483132,  0.90948993, -0.11689905,
                  1.23484548,  1.38018566, -0.57539165, -0.22204826, -0.37269637,
                  1.51305089, -1.05739763,  0.3571857 , -0.12866277, -0.10547563,
                 -0.53502926, -0.91608495,  0.9249523 ,  1.25270494,  0.95080015,
                 -0.84234887,  1.53095717,  0.78535597,  1.59884341,  0.99957283,
                  0.64464005, -1.0699057 ,  2.00213519, -0.40614721,  0.33469936,
                  0.65214022,  0.03407413,  0.96458548,  1.6415562 , -0.02042865,
                  0.91079629,  0.51978855, -0.08485074,  2.01666972, -0.19347125,
                  1.00515377,  0.35275976,  1.31814843,  0.18031667, -0.10900477,
                 -0.02504993,  0.74639483, -0.70470324,  0.18864587,  0.77469984,
                  1.12402685,  1.86928319, -0.04793449,  0.96031217,  0.39875679,
                  1.73646864,  0.44796085,  0.53978437, -0.10049249,  1.733278  ,
                  1.19167168, -0.05156596,  0.26728811,  1.04684185, -0.23701974,
                  0.80800707, -0.08867949, -0.7399108 ,  1.35458806,  2.1841648 ,
                  0.85739515, -0.723131  , -0.73175367, -0.79048783, -0.16666084,
                  0.94541406,  1.09982239,  1.1253117 ,  0.61537428,  0.95447778,
                 -0.14118295,  0.85190312,  1.41971143,  0.65454195,  0.57056522,
                 -0.8452877 ,  0.33104374,  0.61072908,  1.81211192, -0.55011263,
                  0.59224252,  0.75936198, -0.97647202,  0.98687934,  0.3016817 ]),
     'y': array([ 8.95676502e-01,  1.30271685e+00,  9.76101642e-01,  2.80754689e-01,
                  1.66286038e-01,  4.83073987e-01,  2.47423182e-01,  1.01815788e-01,
                  8.57912687e-01, -3.95794592e-01, -1.39296564e-01,  5.79894859e-01,
                 -8.49468200e-02,  5.48867708e-01,  1.65797662e-01, -3.23969818e-01,
                 -6.09361126e-01,  8.62038970e-01, -3.02042473e-01,  1.39728522e+00,
                  1.05710297e+00,  4.46290153e-01,  1.01588641e+00, -1.12110147e-01,
                  8.55213463e-01, -1.02457679e-01,  8.67682891e-01,  7.94216700e-01,
                  4.59400469e-01,  7.56741666e-01,  9.20982998e-01, -1.64729837e-01,
                  4.72726656e-01,  3.73867195e-02, -1.23232491e-01, -3.04171205e-01,
                 -2.04263083e-01,  1.13052981e+00,  1.01221924e+00, -7.09509758e-02,
                  1.31825405e-01,  3.07844103e-01,  3.04118060e-02,  4.77480270e-01,
                  1.04646340e+00,  8.11853040e-01,  2.41093668e-01,  1.04966761e+00,
                  1.86882595e-01, -1.07540976e-01, -2.10770022e-01, -3.46502525e-01,
                 -3.76075277e-01,  6.92055325e-01,  8.53930735e-01,  2.84801371e-02,
                 -2.15304911e-01,  3.57049546e-01,  6.38006337e-01, -6.04515343e-01,
                  2.46243758e-01,  1.11798900e+00,  9.65635921e-01,  1.26746270e+00,
                  4.22991878e-01,  3.13781170e-02, -3.52662177e-01,  4.91608299e-01,
                  1.04758959e+00,  7.76299277e-01,  7.10105796e-01,  4.08723519e-01,
                  8.97416399e-02, -4.93166179e-01, -3.66669160e-01, -5.52977376e-02,
                  5.93118901e-01,  1.04503159e-01, -1.29355939e-03, -1.08716725e-01,
                 -5.25859404e-02,  3.62101610e-01,  6.84687315e-01, -1.22552202e-01,
                  2.71883094e-01,  5.97562281e-01,  6.06191800e-01, -1.73247735e-01,
                 -1.41934849e-01,  8.72540820e-01, -5.56155777e-02,  9.17276323e-01,
                  8.54102820e-02, -8.43174348e-02, -5.64113827e-01,  6.27124515e-01,
                  1.07722614e-01,  4.42084952e-01, -6.51429811e-01, -4.02356158e-01,
                  6.76901169e-01,  6.26934448e-01,  4.40131467e-01, -5.72951934e-01,
                  2.76547398e-01,  1.12835077e+00,  4.39880646e-01, -1.34719371e-01,
                  1.48935910e-01,  9.39175504e-02,  1.03706366e-01, -3.32122026e-01,
                  5.55939460e-01,  8.26443689e-01,  1.05542037e+00, -4.98012648e-01,
                  6.27313630e-01,  1.10288121e-02,  4.48665676e-01,  8.98269904e-01,
                  6.72548646e-01,  3.28530778e-02, -5.50122171e-01, -5.26482279e-01,
                 -4.31831643e-01,  4.20378766e-01, -1.83055312e-01, -4.25805639e-01,
                 -5.10323198e-01, -3.12246583e-02,  7.96693007e-01,  9.22300096e-02,
                 -9.63987055e-03,  5.51468553e-01,  9.53790352e-03,  1.04250894e+00,
                 -5.89005437e-02,  2.49179428e-01, -6.88284440e-01,  5.34667518e-01,
                 -7.67328228e-01, -4.19835169e-01, -5.52648582e-02, -7.57917988e-02,
                  8.27541934e-01,  2.76803239e-01,  6.61983680e-01, -4.42786128e-01,
                 -1.27738317e-02,  4.63710978e-01,  7.47544687e-01,  7.48051104e-01,
                  8.37130021e-01,  8.33279422e-01,  6.76723456e-01,  2.01105398e-01,
                 -8.72580477e-02,  9.96844701e-01, -9.41546901e-01,  1.45153766e-02,
                  1.92049493e-01,  5.85451365e-01, -8.76116454e-02,  2.13837813e-02,
                 -5.86054148e-01,  3.95119944e-01,  9.61960204e-01,  9.98725154e-02,
                  1.67763889e-01,  9.31751591e-01,  6.88423021e-01,  1.16445600e+00,
                  1.02240319e+00,  3.53953319e-01,  3.26688594e-01, -6.16580586e-01,
                  2.52639694e-01,  3.31220690e-02,  7.11549555e-01,  1.21050094e+00,
                 -2.98959425e-01, -6.04284386e-01, -5.45748891e-01, -4.59946460e-01,
                 -6.39418951e-01,  3.38253353e-01,  6.85005047e-01, -1.71769176e-02,
                  9.55617397e-01, -2.93420674e-01,  7.63525878e-01, -5.69513908e-01,
                 -5.43282867e-01,  4.00014876e-01,  7.37285482e-01, -6.30330997e-01,
                 -1.59939504e-01,  9.08485666e-01, -5.29767648e-01,  6.73980692e-01])},
    {'colorscale': [[0, 'rgb(0,0,0)'], [1, 'rgb(0,0,0)']],
              'contours': {'coloring': 'lines', 'end': 0.5, 'showlabels': False, 'start': 0.5},
              'line': {'width': 5},
              'ncontours': 1,
              'showscale': False,
              'type': 'contour',
              'uid': '44944db3-9df9-4164-b3fd-b7e0343e63d9',
              'x': array([-1.64483117, -1.62483117, -1.60483117, ...,  2.75516883,  2.77516883,
                           2.79516883]),
              'y': array([-1.44154690e+00, -1.42154690e+00, -1.40154690e+00, -1.38154690e+00,
                          -1.36154690e+00, -1.34154690e+00, -1.32154690e+00, -1.30154690e+00,
                          -1.28154690e+00, -1.26154690e+00, -1.24154690e+00, -1.22154690e+00,
                          -1.20154690e+00, -1.18154690e+00, -1.16154690e+00, -1.14154690e+00,
                          -1.12154690e+00, -1.10154690e+00, -1.08154690e+00, -1.06154690e+00,
                          -1.04154690e+00, -1.02154690e+00, -1.00154690e+00, -9.81546901e-01,
                          -9.61546901e-01, -9.41546901e-01, -9.21546901e-01, -9.01546901e-01,
                          -8.81546901e-01, -8.61546901e-01, -8.41546901e-01, -8.21546901e-01,
                          -8.01546901e-01, -7.81546901e-01, -7.61546901e-01, -7.41546901e-01,
                          -7.21546901e-01, -7.01546901e-01, -6.81546901e-01, -6.61546901e-01,
                          -6.41546901e-01, -6.21546901e-01, -6.01546901e-01, -5.81546901e-01,
                          -5.61546901e-01, -5.41546901e-01, -5.21546901e-01, -5.01546901e-01,
                          -4.81546901e-01, -4.61546901e-01, -4.41546901e-01, -4.21546901e-01,
                          -4.01546901e-01, -3.81546901e-01, -3.61546901e-01, -3.41546901e-01,
                          -3.21546901e-01, -3.01546901e-01, -2.81546901e-01, -2.61546901e-01,
                          -2.41546901e-01, -2.21546901e-01, -2.01546901e-01, -1.81546901e-01,
                          -1.61546901e-01, -1.41546901e-01, -1.21546901e-01, -1.01546901e-01,
                          -8.15469010e-02, -6.15469010e-02, -4.15469010e-02, -2.15469010e-02,
                          -1.54690100e-03,  1.84530990e-02,  3.84530990e-02,  5.84530990e-02,
                           7.84530990e-02,  9.84530990e-02,  1.18453099e-01,  1.38453099e-01,
                           1.58453099e-01,  1.78453099e-01,  1.98453099e-01,  2.18453099e-01,
                           2.38453099e-01,  2.58453099e-01,  2.78453099e-01,  2.98453099e-01,
                           3.18453099e-01,  3.38453099e-01,  3.58453099e-01,  3.78453099e-01,
                           3.98453099e-01,  4.18453099e-01,  4.38453099e-01,  4.58453099e-01,
                           4.78453099e-01,  4.98453099e-01,  5.18453099e-01,  5.38453099e-01,
                           5.58453099e-01,  5.78453099e-01,  5.98453099e-01,  6.18453099e-01,
                           6.38453099e-01,  6.58453099e-01,  6.78453099e-01,  6.98453099e-01,
                           7.18453099e-01,  7.38453099e-01,  7.58453099e-01,  7.78453099e-01,
                           7.98453099e-01,  8.18453099e-01,  8.38453099e-01,  8.58453099e-01,
                           8.78453099e-01,  8.98453099e-01,  9.18453099e-01,  9.38453099e-01,
                           9.58453099e-01,  9.78453099e-01,  9.98453099e-01,  1.01845310e+00,
                           1.03845310e+00,  1.05845310e+00,  1.07845310e+00,  1.09845310e+00,
                           1.11845310e+00,  1.13845310e+00,  1.15845310e+00,  1.17845310e+00,
                           1.19845310e+00,  1.21845310e+00,  1.23845310e+00,  1.25845310e+00,
                           1.27845310e+00,  1.29845310e+00,  1.31845310e+00,  1.33845310e+00,
                           1.35845310e+00,  1.37845310e+00,  1.39845310e+00,  1.41845310e+00,
                           1.43845310e+00,  1.45845310e+00,  1.47845310e+00,  1.49845310e+00,
                           1.51845310e+00,  1.53845310e+00,  1.55845310e+00,  1.57845310e+00,
                           1.59845310e+00,  1.61845310e+00,  1.63845310e+00,  1.65845310e+00,
                           1.67845310e+00,  1.69845310e+00,  1.71845310e+00,  1.73845310e+00,
                           1.75845310e+00,  1.77845310e+00,  1.79845310e+00,  1.81845310e+00,
                           1.83845310e+00,  1.85845310e+00,  1.87845310e+00]),
              'z': array([[0.86597938, 0.86597938, 0.86597938, ..., 0.86597938, 0.86597938,
                           0.86597938],
                          [0.86597938, 0.86597938, 0.86597938, ..., 0.86597938, 0.86597938,
                           0.86597938],
                          [0.86597938, 0.86597938, 0.86597938, ..., 0.86597938, 0.86597938,
                           0.86597938],
                          ...,
                          [0.16504854, 0.16504854, 0.16504854, ..., 0.16504854, 0.16504854,
                           0.16504854],
                          [0.16504854, 0.16504854, 0.16504854, ..., 0.16504854, 0.16504854,
                           0.16504854],
                          [0.16504854, 0.16504854, 0.16504854, ..., 0.16504854, 0.16504854,
                           0.16504854]])}],
    'layout': {'autosize': False,
               'font': {'family': 'arial, monospace', 'size': 13},
               'height': 480,
               'template': '...',
               'title': {'text': 'n_estimators = 1, max_depth = 1',
                         'x': 0.41,
                         'xanchor': 'center',
                         'y': 0.9,
                         'yanchor': 'top'},
               'width': 640,
               'xaxis': {'title': {'text': 'feature 1'}},
               'yaxis': {'title': {'text': 'feature 2'}}}
})
```

ML algo	suitable for large datasets?	behaviour wrt outliers	non-linear?	params to tune	smooth predictions	easy to interpret?
linear regression	yes	linear extrapolation	no	l1 and/or l2 reg	yes	yes
logistic regression	yes	scales with distance from the decision boundary	no	l1 and/or l2 reg	yes	yes
random forest regression	so so	constant	yes	max_features, max_depth	no	so so

ML algo	suitable for large datasets?	behaviour wrt outliers	non-linear?	params to tune	smooth predictions	easy to interpret?
random forest classification	so so	step-like, difficult to tell	yes	max_features, max_depth	no	so so
SVM rbf regression	tbd	tbd	tbd	tbd	tbd	tbd
SVM rbf classification	tbd	tbd	tbd	tbd	tbd	tbd

Quiz 1

Support Vector Machine

- very versatile technique, it comes in lots of flavors/types, read more about it here
- SVM classifier motivation
  - points in n dimensional space with class 0 and 1
  - we want to find the (n-1) dimensional hyperplane that best separates the points
  - this hyperplane is our (linear) decision boundary
- we cover SVMs with radial basis functions (rbf)
  - we apply a kernel function (a non-linear transformation) to the data points
  - the kernel function basically "smears" the points
  - gaussian rbf kernel: $\exp(-\gamma(|x-x'|)^2)$ where $\gamma > 0$

SVR

```
In [5]: import numpy as np
        from sklearn.svm import SVR
        np.random.seed(10)
        def true_fun(X):
            return np.cos(1.5 * np.pi * X)

        n_samples = 30

        X = np.random.rand(n_samples)
        y = true_fun(X) + np.random.randn(n_samples) * 0.1

        X_new = np.linspace(-0.5, 1.5, 2000)

        reg = SVR(gamma = 1, C = 1)
        reg.fit(X[:, np.newaxis],y)
        y_new = reg.predict(X_new[:, np.newaxis])
```

```
In [ ]: help(SVR)
```

```
In [6]: hyperparameters = {
            'gamma': [1e-3, 1e-1, 1e1, 1e3, 1e5],
            'C': [1e-2, 1e-1, 1e0, 1e1, 1e2]
        }

        vis(X, y, SVR, hyperparameters, X_new)
```

```
interactive(children=(SelectionSlider(description='gamma', options=(0.001, 0.1, 10.0, 1000.0, 100000.0), value…
```

Quiz 2

Let's measure how long it takes to fit a linear regression, random forest regression, and SVR as a function of `n_samples` using our toy regression dataset.

Check this stackoverflow post to figure out how to measure the execution time of a couple of lines of code.

Set n_estimators to 10 and max_depth to 3 in the random forest.

Set the gamma and C parameters to 1 in SVR.

Fit models with n_samples = 1000, 2000, 3000, 4000, 5000. Measure how long it takes to fit each model.

Plot the run time as a function of n_samples for the three models. You might need to adjust the y axis range to check some of the statements.

Which of these statements are true?

- The random forest run-time scales linearly with n_samples.
- The linear regression model is the fastest to fit.

- The SVR run-time scales worse than linear. (I.e., if we double n_sample, the fit time more than doubles.)

In [ ]:

ML algo	suitable for large datasets?	behaviour wrt outliers	non-linear?	params to tune	smooth predictions	easy to interpret?
linear regression	yes	linear extrapolation	no	l1 and/or l2 reg	yes	yes
logistic regression	yes	scales with distance from the decision boundary	no	l1 and/or l2 reg	yes	yes
random forest regression	so so	constant	yes	max_features, max_depth	no	so so
random forest classification	so so	step-like, difficult to tell	yes	max_features, max_depth	no	so so
SVM rbf regression	no	non-linear extrapolation	yes	C, gamma	yes	so so
SVM rbf classification	tbd	tbd	tbd	tbd	tbd	tbd

SVC

In [7]:
```
from sklearn.datasets import make_moons
import numpy as np
from sklearn.svm import SVC

# create the data
X,y = make_moons(noise=0.2, random_state=1,n_samples=200)
# set the hyperparameters
clf = SVC(gamma = 1, C = 1, probability=True)
# fit the model
clf.fit(X,y)
# predict new data
#y_new = clf.predict(X_new)
# predict probabilities
#y_new = clf.predict_proba(X_new)
```

Out[7]:
```
▾ SVC
SVC(C=1, gamma=1, probability=True)
```

In [ ]:
```
help(SVC)
```

In [8]:
```
# initialize RandomForestClassifier
ML_algo = SVC(probability=True)

# SVC parameter grid
hyperparameters = {
    'gamma': [1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3],
    'C': [1e-2, 1e-1, 1e0, 1e1, 1e2]
}

plot_clf_contour(hyperparameters, X, y)
```

```
interactive(children=(SelectionSlider(description='gamma', options=(0.001, 0.01, 0.1, 1.0, 10.0, 100.0, 1000.0…
Out[8]: FigureWidget({
    'data': [{'colorbar': {'title': {'text': 'predicted probability'}},
              'colorscale': [[0.0, 'rgb(103,0,31)'], [0.1, 'rgb(178,24,43)'],
                             [0.2, 'rgb(214,96,77)'], [0.3, 'rgb(244,165,130)'],
                             [0.4, 'rgb(253,219,199)'], [0.5, 'rgb(247,247,247)'],
                             [0.6, 'rgb(209,229,240)'], [0.7, 'rgb(146,197,222)'],
                             [0.8, 'rgb(67,147,195)'], [0.9, 'rgb(33,102,172)'],
                             [1.0, 'rgb(5,48,97)']],
              'contours': {'end': 1, 'size': 0.05, 'start': 0},
              'type': 'contour',
              'uid': 'd5676a93-ac65-4e0f-81b9-553be0222319',
              'x': array([-1.64483117, -1.62483117, -1.60483117, ...,  2.75516883,  2.77516883,
                           2.79516883]),
              'y': array([-1.44154690e+00, -1.42154690e+00, -1.40154690e+00, -1.38154690e+00,
                          -1.36154690e+00, -1.34154690e+00, -1.32154690e+00, -1.30154690e+00,
                          -1.28154690e+00, -1.26154690e+00, -1.24154690e+00, -1.22154690e+00,
                          -1.20154690e+00, -1.18154690e+00, -1.16154690e+00, -1.14154690e+00,
                          -1.12154690e+00, -1.10154690e+00, -1.08154690e+00, -1.06154690e+00,
                          -1.04154690e+00, -1.02154690e+00, -1.00154690e+00, -9.81546901e-01,
                          -9.61546901e-01, -9.41546901e-01, -9.21546901e-01, -9.01546901e-01,
                          -8.81546901e-01, -8.61546901e-01, -8.41546901e-01, -8.21546901e-01,
                          -8.01546901e-01, -7.81546901e-01, -7.61546901e-01, -7.41546901e-01,
                          -7.21546901e-01, -7.01546901e-01, -6.81546901e-01, -6.61546901e-01,
                          -6.41546901e-01, -6.21546901e-01, -6.01546901e-01, -5.81546901e-01,
                          -5.61546901e-01, -5.41546901e-01, -5.21546901e-01, -5.01546901e-01,
                          -4.81546901e-01, -4.61546901e-01, -4.41546901e-01, -4.21546901e-01,
                          -4.01546901e-01, -3.81546901e-01, -3.61546901e-01, -3.41546901e-01,
                          -3.21546901e-01, -3.01546901e-01, -2.81546901e-01, -2.61546901e-01,
                          -2.41546901e-01, -2.21546901e-01, -2.01546901e-01, -1.81546901e-01,
                          -1.61546901e-01, -1.41546901e-01, -1.21546901e-01, -1.01546901e-01,
                          -8.15469010e-02, -6.15469010e-02, -4.15469010e-02, -2.15469010e-02,
                          -1.54690100e-03,  1.84530990e-02,  3.84530990e-02,  5.84530990e-02,
                           7.84530990e-02,  9.84530990e-02,  1.18453099e-01,  1.38453099e-01,
                           1.58453099e-01,  1.78453099e-01,  1.98453099e-01,  2.18453099e-01,
                           2.38453099e-01,  2.58453099e-01,  2.78453099e-01,  2.98453099e-01,
                           3.18453099e-01,  3.38453099e-01,  3.58453099e-01,  3.78453099e-01,
                           3.98453099e-01,  4.18453099e-01,  4.38453099e-01,  4.58453099e-01,
                           4.78453099e-01,  4.98453099e-01,  5.18453099e-01,  5.38453099e-01,
                           5.58453099e-01,  5.78453099e-01,  5.98453099e-01,  6.18453099e-01,
                           6.38453099e-01,  6.58453099e-01,  6.78453099e-01,  6.98453099e-01,
                           7.18453099e-01,  7.38453099e-01,  7.58453099e-01,  7.78453099e-01,
                           7.98453099e-01,  8.18453099e-01,  8.38453099e-01,  8.58453099e-01,
                           8.78453099e-01,  8.98453099e-01,  9.18453099e-01,  9.38453099e-01,
                           9.58453099e-01,  9.78453099e-01,  9.98453099e-01,  1.01845310e+00,
                           1.03845310e+00,  1.05845310e+00,  1.07845310e+00,  1.09845310e+00,
                           1.11845310e+00,  1.13845310e+00,  1.15845310e+00,  1.17845310e+00,
                           1.19845310e+00,  1.21845310e+00,  1.23845310e+00,  1.25845310e+00,
                           1.27845310e+00,  1.29845310e+00,  1.31845310e+00,  1.33845310e+00,
                           1.35845310e+00,  1.37845310e+00,  1.39845310e+00,  1.41845310e+00,
                           1.43845310e+00,  1.45845310e+00,  1.47845310e+00,  1.49845310e+00,
                           1.51845310e+00,  1.53845310e+00,  1.55845310e+00,  1.57845310e+00,
                           1.59845310e+00,  1.61845310e+00,  1.63845310e+00,  1.65845310e+00,
                           1.67845310e+00,  1.69845310e+00,  1.71845310e+00,  1.73845310e+00,
                           1.75845310e+00,  1.77845310e+00,  1.79845310e+00,  1.81845310e+00,
                           1.83845310e+00,  1.85845310e+00,  1.87845310e+00]),
              'z': array([[0.5, 0.5, 0.5, ..., 0.5, 0.5, 0.5],
                          [0.5, 0.5, 0.5, ..., 0.5, 0.5, 0.5],
                          [0.5, 0.5, 0.5, ..., 0.5, 0.5, 0.5],
                          ...,
                          [0.5, 0.5, 0.5, ..., 0.5, 0.5, 0.5],
                          [0.5, 0.5, 0.5, ..., 0.5, 0.5, 0.5],
                          [0.5, 0.5, 0.5, ..., 0.5, 0.5, 0.5]])},
             {'marker': {'color': array([0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0,
                                         0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0,
                                         1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0,
                                         1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1,
                                         1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0,
                                         0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1,
                                         0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1,
                                         0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1,
                                         1, 1, 0, 1, 1, 0, 1, 0]),
                         'colorscale': [[0, 'rgb(255,0,0)'], [1, 'rgb(0,0,255)']],
                         'line': {'width': 1},
                         'size': 8},
              'mode': 'markers',
              'type': 'scatter',
              'uid': '474616ca-8c4c-419d-ba64-412c5c88cd61',
              'x': array([-0.31410929,  0.39443922,  0.48606504, -0.12805768,  1.73330291,
                           2.08500896, -1.08258853,  1.46899598,  0.54077095,  2.04624573,
                           1.26754011,  0.84766004,  1.12118285, -0.78670168,  0.00903277,
                           0.44164301,  1.43162307,  0.42529121,  0.47635606, -0.21866384,
                          -0.08003084,  0.89546191,  0.22021464, -0.86869386,  0.720033  ,
                           1.13779598,  0.8517701 ,  0.78598197,  1.9446092 ,  0.57102512,
                          -0.0874615 ,  1.36354658,  2.01180011,  1.74821345,  2.02576178,
                          -1.01470657,  1.66026241, -0.95942153, -0.69886079,  1.69148155,
                           0.1386857 , -0.08753717,  2.30359131,  0.89058213, -0.18985218,
                           0.87929191, -0.18902626,  0.62843809,  1.98598879,  0.42815849,
             -0.76592918, -0.1766055 , -0.16502413,  0.09064031,  0.58215419,
              1.59947816,  1.46796344,  0.07718396,  2.05560682,  0.45951857,
              0.14430234, -0.46286897, -0.75771806,  0.31091269,  0.44194998,
              0.95163996,  0.4400647 ,  0.087738  , -0.19010962, -0.91944812,
             -0.04820292, -0.77905887,  1.82882388,  1.24333486,  1.85387971,
             -0.04940048,  0.8753758 , -0.12773072, -0.97433768, -0.04547499,
              1.30029351, -1.14483117, -1.00043214,  0.53227789,  2.18759633,
              1.18059461, -0.60654542,  0.28976789,  2.0755619 ,  0.36715992,
             -0.70800052,  0.50273757,  1.65490162,  1.84352533,  1.30164148,
              0.22470731,  1.94238198,  2.03105931,  0.82243733,  1.85498244,
             -0.85944022,  0.48924579, -1.09621553,  1.09430083,  0.9242366 ,
             -0.59077915, -0.78977567,  0.95483132,  0.90948993, -0.11689905,
              1.23484548,  1.38018566, -0.57539165, -0.22204826, -0.37269637,
              1.51305089, -1.05739763,  0.3571857 , -0.12866277, -0.10547563,
             -0.53502926, -0.91608495,  0.9249523 ,  1.25270494,  0.95080015,
             -0.84234887,  1.53095717,  0.78535597,  1.59884341,  0.99957283,
              0.64464005, -1.0699057 ,  2.00213519, -0.40614721,  0.33469936,
              0.65214022,  0.03407413,  0.96458548,  1.6415562 , -0.02042865,
              0.91079629,  0.51978855, -0.08485074,  2.01666972, -0.19347125,
              1.00515377,  0.35275976,  1.31814843,  0.18031667, -0.10900477,
             -0.02504993,  0.74639483, -0.70470324,  0.18864587,  0.77469984,
              1.12402685,  1.86928319, -0.04793449,  0.96031217,  0.39875679,
              1.73646864,  0.44796085,  0.53978437, -0.10049249,  1.733278  ,
              1.19167168, -0.05156596,  0.26728811,  1.04684185, -0.23701974,
              0.80800707, -0.08867949, -0.7399108 ,  1.35458806,  2.1841648 ,
              0.85739515, -0.723131  , -0.73175367, -0.79048783, -0.16666084,
              0.94541406,  1.09982239,  1.1253117 ,  0.61537428,  0.95447778,
             -0.14118295,  0.85190312,  1.41971143,  0.65454195,  0.57056522,
             -0.8452877 ,  0.33104374,  0.61072908,  1.81211192, -0.55011263,
              0.59224252,  0.75936198, -0.97647202,  0.98687934,  0.3016817 ]),
 'y': array([ 8.95676502e-01,  1.30271685e+00,  9.76101642e-01,  2.80754689e-01,
              1.66286038e-01,  4.83073987e-01,  2.47423182e-01,  1.01815788e-01,
              8.57912687e-01, -3.95794592e-01, -1.39296564e-01,  5.79894859e-01,
             -8.49468200e-02,  5.48867708e-01,  1.65797662e-01, -3.23969818e-01,
             -6.09361126e-01,  8.62038970e-01, -3.02042473e-01,  1.39728522e+00,
              1.05710297e+00,  4.46290153e-01,  1.01588641e+00, -1.12110147e-01,
              8.55213463e-01, -1.02457679e-01,  8.67682891e-01,  7.94216700e-01,
              4.59400469e-01,  7.56741666e-01,  9.20982998e-01, -1.64729837e-01,
              4.72726656e-01,  3.73867195e-02, -1.23232491e-01, -3.04171205e-01,
             -2.04263083e-01,  1.13052981e+00,  1.01221924e+00, -7.09509758e-02,
              1.31825405e-01,  3.07844103e-01,  3.04118060e-02,  4.77480270e-01,
              1.04646340e+00,  8.11853040e-01,  2.41093668e-01,  1.04966761e+00,
              1.86882595e-01, -1.07540976e-01, -2.10770022e-01, -3.46502525e-01,
             -3.76075277e-01,  6.92055325e-01,  8.53930735e-01,  2.84801371e-02,
             -2.15304911e-01,  3.57049546e-01,  6.38006337e-01, -6.04515343e-01,
              2.46243758e-01,  1.11798900e+00,  9.65635921e-01,  1.26746270e+00,
              4.22991878e-01,  3.13781170e-02, -3.52662177e-01,  4.91608299e-01,
              1.04758959e+00,  7.76299277e-01,  7.10105796e-01,  4.08723519e-01,
              8.97416399e-02, -4.93166179e-01, -3.66669160e-01, -5.52977376e-02,
              5.93118901e-01,  1.04503159e-01, -1.29355939e-03, -1.08716725e-01,
             -5.25859404e-02,  3.62101610e-01,  6.84687315e-01, -1.22552202e-01,
              2.71883094e-01,  5.97562281e-01,  6.06191800e-01, -1.73247735e-01,
             -1.41934849e-01,  8.72540820e-01, -5.56155777e-02,  9.17276323e-01,
              8.54102820e-02, -8.43174348e-02, -5.64113827e-01,  6.27124515e-01,
              1.07722614e-01,  4.42084952e-01, -6.51429811e-01, -4.02356158e-01,
              6.76901169e-01,  6.26934448e-01,  4.40131467e-01, -5.72951934e-01,
              2.76547398e-01,  1.12835077e+00,  4.39880646e-01, -1.34719371e-01,
              1.48935910e-01,  9.39175504e-02,  1.03706366e-01, -3.32122026e-01,
              5.55939460e-01,  8.26443689e-01,  1.05542037e+00, -4.98012648e-01,
              6.27313630e-01,  1.10288121e-02,  4.48665676e-01,  8.98269904e-01,
              6.72548646e-01,  3.28530778e-02, -5.50122171e-01, -5.26482279e-01,
             -4.31831643e-01,  4.20378766e-01, -1.83055312e-01, -4.25805639e-01,
             -5.10323198e-01, -3.12246583e-02,  7.96693007e-01,  9.22300096e-02,
             -9.63987055e-03,  5.51468553e-01,  9.53790352e-03,  1.04250894e+00,
             -5.89005437e-02,  2.49179428e-01, -6.88284440e-01,  5.34667518e-01,
             -7.67328228e-01, -4.19835169e-01, -5.52648582e-02, -7.57917988e-02,
              8.27541934e-01,  2.76803239e-01,  6.61983680e-01, -4.42786128e-01,
             -1.27738317e-02,  4.63710978e-01,  7.47544687e-01,  7.48051104e-01,
              8.37130021e-01,  8.33279422e-01,  6.76723456e-01,  2.01105398e-01,
             -8.72580477e-02,  9.96844701e-01, -9.41546901e-01,  1.45153766e-02,
              1.92049493e-01,  5.85451365e-01, -8.76116454e-02,  2.13837813e-02,
             -5.86054148e-01,  3.95119944e-01,  9.61960204e-01,  9.98725154e-02,
              1.67763889e-01,  9.31751591e-01,  6.88423021e-01,  1.16445600e+00,
              1.02240319e+00,  3.53953319e-01,  3.26688594e-01, -6.16580586e-01,
              2.52639694e-01,  3.31220690e-02,  7.11549555e-01,  1.21050094e+00,
             -2.98959425e-01, -6.04284386e-01, -5.45748891e-01, -4.59946460e-01,
             -6.39418951e-01,  3.38253353e-01,  6.85005047e-01, -1.71769176e-02,
              9.55617397e-01, -2.93420674e-01,  7.63525878e-01, -5.69513908e-01,
             -5.43282867e-01,  4.00014876e-01,  7.37285482e-01, -6.30330997e-01,
             -1.59939504e-01,  9.08485666e-01, -5.29767648e-01,  6.73980692e-01])},
{'colorscale': [[0, 'rgb(0,0,0)'], [1, 'rgb(0,0,0)']],
 'contours': {'coloring': 'lines', 'end': 0.5, 'showlabels': False, 'start': 0.5},
 'line': {'width': 5},
 'ncontours': 1,
 'showscale': False,
 'type': 'contour',
 'uid': 'bf95008b-23bc-4881-bada-7c82b3fbdeba',
              'x': array([-1.64483117, -1.62483117, -1.60483117, ...,  2.75516883,  2.77516883,
                           2.79516883]),
              'y': array([-1.44154690e+00, -1.42154690e+00, -1.40154690e+00, -1.38154690e+00,
                          -1.36154690e+00, -1.34154690e+00, -1.32154690e+00, -1.30154690e+00,
                          -1.28154690e+00, -1.26154690e+00, -1.24154690e+00, -1.22154690e+00,
                          -1.20154690e+00, -1.18154690e+00, -1.16154690e+00, -1.14154690e+00,
                          -1.12154690e+00, -1.10154690e+00, -1.08154690e+00, -1.06154690e+00,
                          -1.04154690e+00, -1.02154690e+00, -1.00154690e+00, -9.81546901e-01,
                          -9.61546901e-01, -9.41546901e-01, -9.21546901e-01, -9.01546901e-01,
                          -8.81546901e-01, -8.61546901e-01, -8.41546901e-01, -8.21546901e-01,
                          -8.01546901e-01, -7.81546901e-01, -7.61546901e-01, -7.41546901e-01,
                          -7.21546901e-01, -7.01546901e-01, -6.81546901e-01, -6.61546901e-01,
                          -6.41546901e-01, -6.21546901e-01, -6.01546901e-01, -5.81546901e-01,
                          -5.61546901e-01, -5.41546901e-01, -5.21546901e-01, -5.01546901e-01,
                          -4.81546901e-01, -4.61546901e-01, -4.41546901e-01, -4.21546901e-01,
                          -4.01546901e-01, -3.81546901e-01, -3.61546901e-01, -3.41546901e-01,
                          -3.21546901e-01, -3.01546901e-01, -2.81546901e-01, -2.61546901e-01,
                          -2.41546901e-01, -2.21546901e-01, -2.01546901e-01, -1.81546901e-01,
                          -1.61546901e-01, -1.41546901e-01, -1.21546901e-01, -1.01546901e-01,
                          -8.15469010e-02, -6.15469010e-02, -4.15469010e-02, -2.15469010e-02,
                          -1.54690100e-03,  1.84530990e-02,  3.84530990e-02,  5.84530990e-02,
                           7.84530990e-02,  9.84530990e-02,  1.18453099e-01,  1.38453099e-01,
                           1.58453099e-01,  1.78453099e-01,  1.98453099e-01,  2.18453099e-01,
                           2.38453099e-01,  2.58453099e-01,  2.78453099e-01,  2.98453099e-01,
                           3.18453099e-01,  3.38453099e-01,  3.58453099e-01,  3.78453099e-01,
                           3.98453099e-01,  4.18453099e-01,  4.38453099e-01,  4.58453099e-01,
                           4.78453099e-01,  4.98453099e-01,  5.18453099e-01,  5.38453099e-01,
                           5.58453099e-01,  5.78453099e-01,  5.98453099e-01,  6.18453099e-01,
                           6.38453099e-01,  6.58453099e-01,  6.78453099e-01,  6.98453099e-01,
                           7.18453099e-01,  7.38453099e-01,  7.58453099e-01,  7.78453099e-01,
                           7.98453099e-01,  8.18453099e-01,  8.38453099e-01,  8.58453099e-01,
                           8.78453099e-01,  8.98453099e-01,  9.18453099e-01,  9.38453099e-01,
                           9.58453099e-01,  9.78453099e-01,  9.98453099e-01,  1.01845310e+00,
                           1.03845310e+00,  1.05845310e+00,  1.07845310e+00,  1.09845310e+00,
                           1.11845310e+00,  1.13845310e+00,  1.15845310e+00,  1.17845310e+00,
                           1.19845310e+00,  1.21845310e+00,  1.23845310e+00,  1.25845310e+00,
                           1.27845310e+00,  1.29845310e+00,  1.31845310e+00,  1.33845310e+00,
                           1.35845310e+00,  1.37845310e+00,  1.39845310e+00,  1.41845310e+00,
                           1.43845310e+00,  1.45845310e+00,  1.47845310e+00,  1.49845310e+00,
                           1.51845310e+00,  1.53845310e+00,  1.55845310e+00,  1.57845310e+00,
                           1.59845310e+00,  1.61845310e+00,  1.63845310e+00,  1.65845310e+00,
                           1.67845310e+00,  1.69845310e+00,  1.71845310e+00,  1.73845310e+00,
                           1.75845310e+00,  1.77845310e+00,  1.79845310e+00,  1.81845310e+00,
                           1.83845310e+00,  1.85845310e+00,  1.87845310e+00]),
              'z': array([[0.5, 0.5, 0.5, ..., 0.5, 0.5, 0.5],
                          [0.5, 0.5, 0.5, ..., 0.5, 0.5, 0.5],
                          [0.5, 0.5, 0.5, ..., 0.5, 0.5, 0.5],
                          ...,
                          [0.5, 0.5, 0.5, ..., 0.5, 0.5, 0.5],
                          [0.5, 0.5, 0.5, ..., 0.5, 0.5, 0.5],
                          [0.5, 0.5, 0.5, ..., 0.5, 0.5, 0.5]])}],
    'layout': {'autosize': False,
               'font': {'family': 'arial, monospace', 'size': 13},
               'height': 480,
               'template': '...',
               'title': {'text': 'gamma = 0.001, C = 0.01', 'x': 0.41, 'xanchor': 'center', 'y': 0.9, 'yanchor':
'top'},
               'width': 640,
               'xaxis': {'title': {'text': 'feature 1'}},
               'yaxis': {'title': {'text': 'feature 2'}}}
})
```

ML algo	suitable for large datasets?	behaviour wrt outliers	non-linear?	params to tune	smooth predictions	easy to interpret?
linear regression	yes	linear extrapolation	no	l1 and/or l2 reg	yes	yes
logistic regression	yes	scales with distance from the decision boundary	no	l1 and/or l2 reg	yes	yes
random forest regression	so so	constant	yes	max_features, max_depth	no	so so
random forest classification	so so	step-like, difficult to tell	yes	max_features, max_depth	no	so so
SVM rbf regression	no	non-linear extrapolation	yes	C, gamma	yes	so so
SVM rbf classification	no	50-50	yes	C, gamma	yes	so so

Quiz 3

Bias variance trade off

Which gamma value gives the best trade off between high bias and high variance? Work through the steps to answer the question.

- Use random_state = 42 where-ever necessary.
- Split X, y into X_train, X_val, y_train, y_val such that 70% of the points are in train.
- Fit SVC models with C = 1, and gamma = 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3 on the training set.
- Measure the validation accuracy for each gamma.
- Which gamma value gives the highest validation accuracy?

```
In [ ]:
```

Mud card