

Problem4

November 23, 2021

1 Problem 4

```
[1]: from sklearn import svm
from sklearn.model_selection import KFold
from sklearn.metrics import roc_curve, roc_auc_score
import pandas as pd
import numpy as np
from concurrent.futures import ProcessPoolExecutor
import matplotlib.pyplot as plt
```

1.1 a) Implement Kernels

```
[2]: def k_dirac(x, y):
    return 1 if x == y else 0
def k_uniform(x, y):
    return 1
def k_multitask(x, y, lmbd=0.5):
    return lmbd * k_dirac(x, y) + (1-lmbd) * k_uniform(x, y)
def k_linseq(x, y):
    return (np.array(list(x)) == np.array(list(y))).sum()
def k_poly(x, y, d=2):
    return (1 + k_linseq(x, y))**d
def join_kernel_matrices(*kernels):
    return np.array(kernels).prod(axis=0)
```

```
[3]: def kernel(k):
    return lambda X, Y: np.array([[
        k(x, y)
        for y in Y]
        for x in X])
```

1.2 b) Build SVM and do cross-validation

```
[4]: binding_data = pd.read_csv("BindingData.csv")
```

```
[5]: binding_data.head()
```

```
[5]:
```

	Epitope ID	binding information	allele	peptide sequence \
0	52886	Positive	HLA-A*01:01	QYDPVAALF
1	71650	Positive	HLA-A*01:01	VVEKQSGLY
2	150319	Positive	HLA-A*01:01	ITEAELTGY
3	141221	Positive	HLA-A*01:01	ATDSLNNY
4	24819	Positive	HLA-A*01:01	HSNLNDATY

	binding label
0	1
1	1
2	1
3	1
4	1

```
[6]: pa_data = binding_data[[" allele", " peptide sequence"]].to_numpy()
print(pa_data)
```

```
[['HLA-A*01:01' 'QYDPVAALF']
 ['HLA-A*01:01' 'VVEKQSGLY']
 ['HLA-A*01:01' 'ITEAELTGY']
 ...
 ['HLA-B*40:01' 'SEVKTLSY']
 ['HLA-B*40:01' 'IELYSLAKY']
 ['HLA-B*40:01' 'KEGKLVIIF']]
```

```
[7]: label_data = binding_data[" binding label"].to_numpy()
print(label_data)
```

```
[ 1  1  1 ... -1 -1 -1]
```

```
[8]: Cs = np.logspace(-4, 4, 9)
print(Cs)
```

```
[1.e-04 1.e-03 1.e-02 1.e-01 1.e+00 1.e+01 1.e+02 1.e+03 1.e+04]
```

```
[9]: def do_cv(C):
    kf = KFold(n_splits=10, shuffle=True)

    scores = dict()
    names = ["uniform", "dirac", "multitask"]
    for name in names:
        scores[name] = {
            "accuracy": [],
            "results": [],
            "truths": [],
            "classifiers": []
        }
```

```

for train_indices, test_indices in kf.split(pa_data):

    Xtrain = pa_data[train_indices]
    Ytrain = label_data[train_indices]
    Xtest = pa_data[test_indices]
    Ytest = label_data[test_indices]

    K_train_p = kernel(k_linseq)(Xtrain[:, 0], Xtrain[:, 0])
    K_test_p = kernel(k_linseq)(Xtest[:, 0], Xtrain[:, 0])

    for n, k in zip(
        names,
        [k_uniform, k_dirac, k_multitask]
    ):
        e = svm.SVC(
            kernel="precomputed",
            C=C
        )
        K_train_a = kernel(k)(Xtrain[:, 1], Xtrain[:, 1])
        K_test_a = kernel(k)(Xtest[:, 1], Xtrain[:, 1])

        Ktrain = join_kernel_matrices(K_train_p, K_train_a)
        Ktest = join_kernel_matrices(K_test_p, K_test_a)

        e.fit(Ktrain, Ytrain)
        score = e.score(Ktest, Ytest)
        scores[n]["accuracy"].append(score)
        print(f"Model {n} (C={C}): accuracy = {score:2.2f}")

        results = e.decision_function(Ktest)
        scores[n]["results"].append(results)
        scores[n]["truths"].append(Ytest)

        scores[n]["classifiers"].append(e)
    for n in scores.keys():
        scores[n]["accuracy"] = np.array(scores[n]["accuracy"]).mean()

    return scores
    #for name in names:
    #out.append(f"Mean accuracy for {name}-kernel and C={C}: {np.
    ↪array(scores[name]).mean()}")

```

```

[10]: futures = []
with ProcessPoolExecutor() as ppe:
    for C in Cs:

```

```

        futures.append((C, ppe.submit(do_cv, C)))
scores = dict()
for C, future in futures:
    scores[C] = future.result()

```

```

Model uniform (C=1000.0): accuracy = 0.42
Model uniform (C=100.0): accuracy = 0.42
Model uniform (C=0.0001): accuracy = 0.52
Model uniform (C=10.0): accuracy = 0.42
Model uniform (C=0.1): accuracy = 0.48
Model uniform (C=0.01): accuracy = 0.52
Model uniform (C=0.001): accuracy = 0.52
Model dirac (C=1000.0): accuracy = 0.52
Model dirac (C=100.0): accuracy = 0.52
Model dirac (C=0.0001): accuracy = 0.52
Model uniform (C=1.0): accuracy = 0.42
Model dirac (C=10.0): accuracy = 0.52
Model dirac (C=0.001): accuracy = 0.52
Model dirac (C=0.01): accuracy = 0.52
Model dirac (C=0.1): accuracy = 0.52
Model dirac (C=1.0): accuracy = 0.52
Model multitask (C=1000.0): accuracy = 0.56
Model multitask (C=0.0001): accuracy = 0.52
Model multitask (C=100.0): accuracy = 0.56
Model multitask (C=10.0): accuracy = 0.56
Model multitask (C=0.001): accuracy = 0.52
Model multitask (C=0.1): accuracy = 0.46
Model multitask (C=0.01): accuracy = 0.52
Model multitask (C=1.0): accuracy = 0.56
Model uniform (C=0.0001): accuracy = 0.51
Model uniform (C=1000.0): accuracy = 0.44
Model uniform (C=100.0): accuracy = 0.44
Model uniform (C=0.1): accuracy = 0.46
Model dirac (C=0.0001): accuracy = 0.51
Model dirac (C=1000.0): accuracy = 0.51
Model uniform (C=0.001): accuracy = 0.51
Model dirac (C=100.0): accuracy = 0.51
Model uniform (C=10.0): accuracy = 0.44
Model uniform (C=0.01): accuracy = 0.51
Model dirac (C=0.1): accuracy = 0.51
Model dirac (C=0.001): accuracy = 0.51
Model uniform (C=1.0): accuracy = 0.44
Model dirac (C=10.0): accuracy = 0.51
Model dirac (C=0.01): accuracy = 0.51
Model multitask (C=0.0001): accuracy = 0.51
Model dirac (C=1.0): accuracy = 0.51
Model multitask (C=1000.0): accuracy = 0.63

```

Model multitask (C=100.0): accuracy = 0.63
 Model multitask (C=0.1): accuracy = 0.52
 Model multitask (C=0.001): accuracy = 0.51
 Model multitask (C=0.01): accuracy = 0.51
 Model multitask (C=10.0): accuracy = 0.63
 Model multitask (C=1.0): accuracy = 0.63
 Model uniform (C=1000.0): accuracy = 0.48
 Model uniform (C=100.0): accuracy = 0.48
 Model uniform (C=0.1): accuracy = 0.50
 Model dirac (C=1000.0): accuracy = 0.55
 Model uniform (C=0.0001): accuracy = 0.55
 Model uniform (C=0.001): accuracy = 0.55
 Model dirac (C=100.0): accuracy = 0.55
 Model dirac (C=0.1): accuracy = 0.55
 Model uniform (C=1.0): accuracy = 0.48
 Model dirac (C=0.0001): accuracy = 0.55
 Model uniform (C=0.01): accuracy = 0.55
 Model uniform (C=10.0): accuracy = 0.48
 Model dirac (C=0.001): accuracy = 0.55
 Model dirac (C=1.0): accuracy = 0.55
 Model multitask (C=1000.0): accuracy = 0.58
 Model dirac (C=0.01): accuracy = 0.55
 Model dirac (C=10.0): accuracy = 0.55
 Model multitask (C=100.0): accuracy = 0.58
 Model multitask (C=0.1): accuracy = 0.55
 Model multitask (C=0.0001): accuracy = 0.55
 Model multitask (C=0.001): accuracy = 0.55
 Model multitask (C=1.0): accuracy = 0.58
 Model multitask (C=0.01): accuracy = 0.55
 Model multitask (C=10.0): accuracy = 0.58
 Model uniform (C=1000.0): accuracy = 0.40
 Model uniform (C=100.0): accuracy = 0.40
 Model dirac (C=1000.0): accuracy = 0.52
 Model uniform (C=0.1): accuracy = 0.45
 Model uniform (C=0.0001): accuracy = 0.52
 Model dirac (C=100.0): accuracy = 0.52
 Model uniform (C=0.001): accuracy = 0.52
 Model uniform (C=1.0): accuracy = 0.40
 Model uniform (C=0.01): accuracy = 0.52
 Model dirac (C=0.1): accuracy = 0.52
 Model dirac (C=0.0001): accuracy = 0.52
 Model dirac (C=0.001): accuracy = 0.52
 Model multitask (C=1000.0): accuracy = 0.48
 Model dirac (C=1.0): accuracy = 0.52
 Model uniform (C=10.0): accuracy = 0.40
 Model dirac (C=0.01): accuracy = 0.52
 Model multitask (C=100.0): accuracy = 0.48
 Model dirac (C=10.0): accuracy = 0.52

Model multitask (C=0.1): accuracy = 0.44
 Model multitask (C=0.0001): accuracy = 0.52
 Model multitask (C=0.001): accuracy = 0.52
 Model multitask (C=1.0): accuracy = 0.48
 Model multitask (C=0.01): accuracy = 0.52
 Model multitask (C=10.0): accuracy = 0.48
 Model uniform (C=1000.0): accuracy = 0.45
 Model uniform (C=0.1): accuracy = 0.48
 Model dirac (C=1000.0): accuracy = 0.52
 Model uniform (C=100.0): accuracy = 0.45
 Model uniform (C=0.0001): accuracy = 0.52
 Model dirac (C=0.1): accuracy = 0.52
 Model uniform (C=0.001): accuracy = 0.52
 Model dirac (C=100.0): accuracy = 0.52
 Model uniform (C=0.01): accuracy = 0.52
 Model dirac (C=0.0001): accuracy = 0.52
 Model multitask (C=1000.0): accuracy = 0.61
 Model dirac (C=0.001): accuracy = 0.52
 Model uniform (C=1.0): accuracy = 0.45
 Model dirac (C=0.01): accuracy = 0.52
 Model uniform (C=10.0): accuracy = 0.45
 Model multitask (C=0.1): accuracy = 0.58
 Model multitask (C=100.0): accuracy = 0.61
 Model dirac (C=1.0): accuracy = 0.52
 Model dirac (C=10.0): accuracy = 0.52
 Model multitask (C=0.0001): accuracy = 0.52
 Model multitask (C=0.001): accuracy = 0.52
 Model multitask (C=0.01): accuracy = 0.52
 Model multitask (C=1.0): accuracy = 0.61
 Model multitask (C=10.0): accuracy = 0.61
 Model uniform (C=1000.0): accuracy = 0.44
 Model uniform (C=0.1): accuracy = 0.46
 Model uniform (C=0.0001): accuracy = 0.50
 Model dirac (C=1000.0): accuracy = 0.51
 Model uniform (C=100.0): accuracy = 0.44
 Model dirac (C=0.0001): accuracy = 0.50
 Model dirac (C=0.1): accuracy = 0.51
 Model uniform (C=0.01): accuracy = 0.50
 Model uniform (C=0.001): accuracy = 0.50
 Model dirac (C=100.0): accuracy = 0.51
 Model dirac (C=0.01): accuracy = 0.50
 Model dirac (C=0.001): accuracy = 0.50
 Model multitask (C=1000.0): accuracy = 0.51
 Model uniform (C=1.0): accuracy = 0.44
 Model multitask (C=0.0001): accuracy = 0.50
 Model multitask (C=0.1): accuracy = 0.49
 Model uniform (C=10.0): accuracy = 0.44
 Model dirac (C=1.0): accuracy = 0.51

Model multitask (C=100.0): accuracy = 0.51
 Model multitask (C=0.01): accuracy = 0.50
 Model multitask (C=0.001): accuracy = 0.50
 Model dirac (C=10.0): accuracy = 0.51
 Model multitask (C=1.0): accuracy = 0.51
 Model multitask (C=10.0): accuracy = 0.51
 Model uniform (C=1000.0): accuracy = 0.45
 Model dirac (C=1000.0): accuracy = 0.56
 Model uniform (C=0.1): accuracy = 0.50
 Model uniform (C=0.0001): accuracy = 0.55
 Model dirac (C=0.1): accuracy = 0.56
 Model uniform (C=0.01): accuracy = 0.55
 Model uniform (C=100.0): accuracy = 0.45
 Model uniform (C=0.001): accuracy = 0.55
 Model dirac (C=0.0001): accuracy = 0.55
 Model multitask (C=1000.0): accuracy = 0.55
 Model dirac (C=0.01): accuracy = 0.55
 Model dirac (C=0.001): accuracy = 0.55
 Model dirac (C=100.0): accuracy = 0.56
 Model uniform (C=1.0): accuracy = 0.46
 Model uniform (C=10.0): accuracy = 0.45
 Model multitask (C=0.1): accuracy = 0.55
 Model multitask (C=0.0001): accuracy = 0.55
 Model dirac (C=1.0): accuracy = 0.56
 Model dirac (C=10.0): accuracy = 0.56
 Model multitask (C=0.01): accuracy = 0.55
 Model multitask (C=0.001): accuracy = 0.55
 Model multitask (C=100.0): accuracy = 0.55
 Model multitask (C=1.0): accuracy = 0.55
 Model multitask (C=10.0): accuracy = 0.55
 Model uniform (C=1000.0): accuracy = 0.45
 Model dirac (C=1000.0): accuracy = 0.60
 Model uniform (C=0.0001): accuracy = 0.62
 Model uniform (C=0.1): accuracy = 0.47
 Model uniform (C=0.01): accuracy = 0.62
 Model uniform (C=0.001): accuracy = 0.62
 Model dirac (C=0.0001): accuracy = 0.62
 Model multitask (C=1000.0): accuracy = 0.55
 Model dirac (C=0.1): accuracy = 0.60
 Model uniform (C=100.0): accuracy = 0.45
 Model dirac (C=0.01): accuracy = 0.62
 Model dirac (C=0.001): accuracy = 0.62
 Model uniform (C=1.0): accuracy = 0.46
 Model dirac (C=100.0): accuracy = 0.60
 Model uniform (C=10.0): accuracy = 0.45
 Model dirac (C=1.0): accuracy = 0.60
 Model multitask (C=0.0001): accuracy = 0.62
 Model multitask (C=0.1): accuracy = 0.54

Model multitask (C=0.01): accuracy = 0.62
 Model dirac (C=10.0): accuracy = 0.60
 Model multitask (C=0.001): accuracy = 0.62
 Model multitask (C=100.0): accuracy = 0.55
 Model multitask (C=1.0): accuracy = 0.55
 Model multitask (C=10.0): accuracy = 0.55
 Model uniform (C=1000.0): accuracy = 0.44
 Model dirac (C=1000.0): accuracy = 0.47
 Model uniform (C=0.0001): accuracy = 0.49
 Model uniform (C=0.01): accuracy = 0.49
 Model multitask (C=1000.0): accuracy = 0.60
 Model uniform (C=0.001): accuracy = 0.49
 Model dirac (C=0.0001): accuracy = 0.49
 Model uniform (C=0.1): accuracy = 0.49
 Model dirac (C=0.01): accuracy = 0.49
 Model uniform (C=100.0): accuracy = 0.44
 Model dirac (C=0.001): accuracy = 0.49
 Model uniform (C=10.0): accuracy = 0.44
 Model dirac (C=0.1): accuracy = 0.47
 Model uniform (C=1.0): accuracy = 0.44
 Model dirac (C=100.0): accuracy = 0.47
 Model dirac (C=10.0): accuracy = 0.47
 Model multitask (C=0.0001): accuracy = 0.49
 Model dirac (C=1.0): accuracy = 0.47
 Model multitask (C=0.01): accuracy = 0.49
 Model multitask (C=0.1): accuracy = 0.45
 Model multitask (C=0.001): accuracy = 0.49
 Model multitask (C=100.0): accuracy = 0.60
 Model multitask (C=10.0): accuracy = 0.60
 Model multitask (C=1.0): accuracy = 0.60
 Model uniform (C=1000.0): accuracy = 0.43
 Model dirac (C=1000.0): accuracy = 0.50
 Model multitask (C=1000.0): accuracy = 0.56
 Model uniform (C=0.0001): accuracy = 0.52
 Model uniform (C=0.1): accuracy = 0.47
 Model uniform (C=0.01): accuracy = 0.52
 Model uniform (C=0.001): accuracy = 0.52
 Model dirac (C=0.0001): accuracy = 0.52
 Model dirac (C=0.1): accuracy = 0.50
 Model uniform (C=1.0): accuracy = 0.43
 Model dirac (C=0.001): accuracy = 0.52
 Model uniform (C=100.0): accuracy = 0.43
 Model dirac (C=0.01): accuracy = 0.52
 Model dirac (C=1.0): accuracy = 0.50
 Model uniform (C=10.0): accuracy = 0.43
 Model dirac (C=100.0): accuracy = 0.50
 Model multitask (C=0.0001): accuracy = 0.52
 Model multitask (C=0.1): accuracy = 0.53


```

Model multitask (C=0.01): accuracy = 0.52
Model multitask (C=0.001): accuracy = 0.52
Model dirac (C=10.0): accuracy = 0.50
Model multitask (C=1.0): accuracy = 0.56
Model multitask (C=100.0): accuracy = 0.56
Model multitask (C=10.0): accuracy = 0.56
Model uniform (C=10000.0): accuracy = 0.43
Model dirac (C=10000.0): accuracy = 0.54
Model multitask (C=10000.0): accuracy = 0.56
Model uniform (C=10000.0): accuracy = 0.40
Model dirac (C=10000.0): accuracy = 0.60
Model multitask (C=10000.0): accuracy = 0.53
Model uniform (C=10000.0): accuracy = 0.48
Model dirac (C=10000.0): accuracy = 0.52
Model multitask (C=10000.0): accuracy = 0.55
Model uniform (C=10000.0): accuracy = 0.44
Model dirac (C=10000.0): accuracy = 0.51
Model multitask (C=10000.0): accuracy = 0.51
Model uniform (C=10000.0): accuracy = 0.40
Model dirac (C=10000.0): accuracy = 0.58
Model multitask (C=10000.0): accuracy = 0.52
Model uniform (C=10000.0): accuracy = 0.47
Model dirac (C=10000.0): accuracy = 0.50
Model multitask (C=10000.0): accuracy = 0.56
Model uniform (C=10000.0): accuracy = 0.42
Model dirac (C=10000.0): accuracy = 0.50
Model multitask (C=10000.0): accuracy = 0.54
Model uniform (C=10000.0): accuracy = 0.43
Model dirac (C=10000.0): accuracy = 0.54
Model multitask (C=10000.0): accuracy = 0.57
Model uniform (C=10000.0): accuracy = 0.43
Model dirac (C=10000.0): accuracy = 0.47
Model multitask (C=10000.0): accuracy = 0.55
Model uniform (C=10000.0): accuracy = 0.46
Model dirac (C=10000.0): accuracy = 0.53
Model multitask (C=10000.0): accuracy = 0.60

```

The values above are for the individual cross validation steps, so they appear 10 times each. We will print the averaged values next:

```

[11]: output = []
      for C, model in scores.items():
          for name, results in model.items():
              output.append(f"Model {name} (C={C}): accuracy = {results['accuracy']:2.
↪2f}")
      print("\n".join(sorted(output)))

```

```

Model dirac (C=0.0001): accuracy = 0.53

```

```

Model dirac (C=0.001): accuracy = 0.53
Model dirac (C=0.01): accuracy = 0.53
Model dirac (C=0.1): accuracy = 0.53
Model dirac (C=1.0): accuracy = 0.53
Model dirac (C=10.0): accuracy = 0.53
Model dirac (C=100.0): accuracy = 0.53
Model dirac (C=1000.0): accuracy = 0.53
Model dirac (C=10000.0): accuracy = 0.53
Model multitask (C=0.0001): accuracy = 0.53
Model multitask (C=0.001): accuracy = 0.53
Model multitask (C=0.01): accuracy = 0.53
Model multitask (C=0.1): accuracy = 0.51
Model multitask (C=1.0): accuracy = 0.56
Model multitask (C=10.0): accuracy = 0.56
Model multitask (C=100.0): accuracy = 0.56
Model multitask (C=1000.0): accuracy = 0.56
Model multitask (C=10000.0): accuracy = 0.55
Model uniform (C=0.0001): accuracy = 0.53
Model uniform (C=0.001): accuracy = 0.53
Model uniform (C=0.01): accuracy = 0.53
Model uniform (C=0.1): accuracy = 0.48
Model uniform (C=1.0): accuracy = 0.44
Model uniform (C=10.0): accuracy = 0.44
Model uniform (C=100.0): accuracy = 0.44
Model uniform (C=1000.0): accuracy = 0.44
Model uniform (C=10000.0): accuracy = 0.43

```

```

[12]: def scores_to_accuracy_dict(scores):
        accuracies = {n: [] for n in scores[1].keys()}
        for C, model in sorted(scores.items(), key=lambda x: x[0]):
            for name, results in model.items():
                accuracies[name].append(results["accuracy"])
        return accuracies

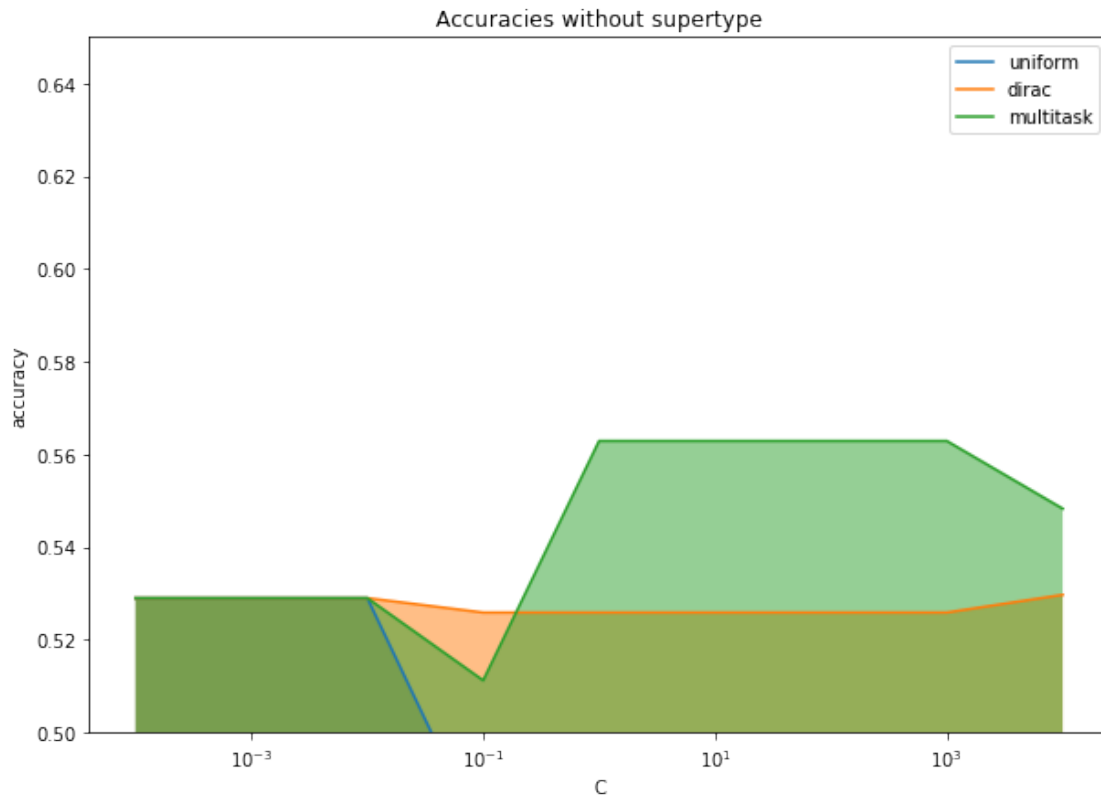
```

```

[13]: def plot_accuracies(accuracies, Cs=C_s, size=(10, 7), ylim=(0.5, 0.65),
    ↪ title=""):
        plt.figure(figsize=size)
        for name, accuracy in accuracies.items():
            plt.plot(Cs, accuracy, label=name)
            plt.fill_between(Cs, accuracy, alpha=0.5)
        plt.xlabel("C")
        plt.ylabel("accuracy")
        plt.xscale("log")
        plt.ylim(ylim)
        plt.title(title)
        plt.legend()
        plt.show()

```

```
[14]: accuracies = scores_to_accuracy_dict(scores)
      plot_accuracies(accuracies, title="Accuracies without supertype")
```



1.3 d1) ROC and AUC + e1) AUC and Accuracy

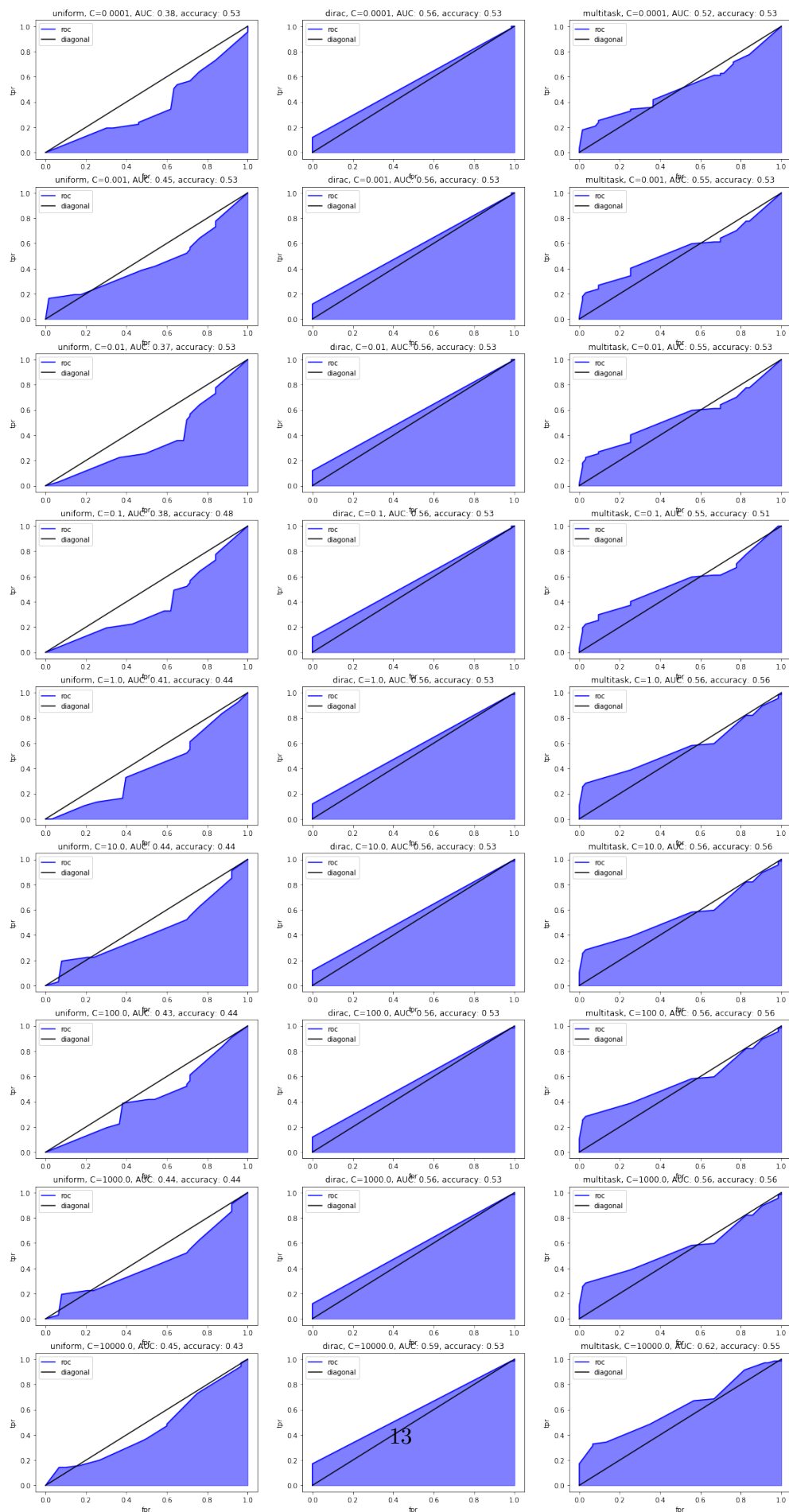
For better readability we will only generate ROC-curves for the fires CV-run. We could easily do more, by just looping over all ten results, but this way we would get $9 * 10 * 3 = 270$ plots, which is not really readable. One could also plot all CV runs in one plot, but that would also be confusing.

```
[18]: def calc_roc(scores):
      out = {C: dict() for C in scores.keys()}
      for C, model in scores.items():
          for name, model_stats in model.items():
              result = model_stats["results"][0]
              truth = model_stats["truths"][0]
              fpr, tpr, t = roc_curve(truth, result)
              auc = roc_auc_score(truth, result)
              out[C][name] = {
                  "tpr": tpr,
                  "fpr": fpr,
                  "auc": auc,
                  "accuracy": model_stats["accuracy"]
```

```
    }  
    return out
```

```
[23]: def plot_roc(roc_data):  
    fig, axss = plt.subplots(  
        len(roc_data.keys()),  
        len(roc_data[1].keys()),  
        figsize=(20, 40)  
    )  
    for (C, model), axss in zip(roc_data.items(), axss):  
        for (name, model_stats), ax in zip(model.items(), axss):  
            ax.plot(model_stats["fpr"], model_stats["tpr"], label="roc",  
→color="blue")  
            ax.fill_between(model_stats["fpr"], model_stats["tpr"],  
→color="blue", alpha=0.5)  
            ax.plot([0,1], [0,1], label="diagonal", color="black")  
            ax.legend()  
            ax.set_xlabel("fpr")  
            ax.set_ylabel("tpr")  
            ax.set_title(f"{name}, C={C}, AUC: {model_stats['auc']:2.2f},  
→accuracy: {model_stats['accuracy']:2.2f}")
```

```
[24]: roc_data = calc_roc(scores)  
plot_roc(roc_data)
```



2 c) Add supertype data

```
[25]: supertypes_data = pd.read_csv("supertype.csv")
```

```
[26]: supertypes_data
```

```
[26]:
```

	Supertype	Motif \
0	A1	x[TI(SVLM)]xxxxxx[WFY]
1	A2	x[LIVMATQ]xxxxxx[LIVMAT]
2	A3	x[AILMVST]xxxxxx[RK]
3	A24	x[YF(WIVLMT)]xxxxxx[FI(YWLM)]
4	B7	x[P]xxxxxx[ALIMVFWY]
5	B27	x[RKH]xxxxxx[FLY(WMI)]
6	B44	x[E(D)]xxxxxx[FWYLIMVA]
7	B58	x[AST]xxxxxx[FWY(LIV)]
8	B62	x[QL(IVMP)]xxxxxx[FWY(MIV)]

	Genotypes
0	A*0101, A*0102, A*2501, A*2601, A*2604, A*3201...
1	A*0201, A*0202, A*0203, A*0204, A*0205, A*0206...
2	A*0301, A*1101, A*3101, A*3301, A*6801
3	A*2301, A*2402, A*2403, A*2404, A*3001, A*3002...
4	B*0702, B*0703, B*0704, B*0705, B*1508, B*3501...
5	B*1401, B*1402, B*1503, B*1509, B*1510, B*1518...
6	B*18, B*3701, B*4001, B*4006, B*4101, B*4402, ...
7	B*1516, B*1517, B*5701, B*5702, B*58
8	B*1301, B*1302, B*1501, B*1502, B*1506, B*1512...

```
[27]: allele_supertype_mapping = []
for i, row in supertypes_data.iterrows():
    s = row["Supertype"]
    for allele in row["Genotypes"].split(", "):
        a = f"HLA-{allele[:-2]}:{allele[-2:]}"
        allele_supertype_mapping.append((a, s))
```

```
[28]: supertype_df = pd.DataFrame(allele_supertype_mapping, columns=["allele", "supertype"])
supertype_df.head()
```

```
[28]:
```

	allele	supertype
0	HLA-A*01:01	A1
1	HLA-A*01:02	A1
2	HLA-A*25:01	A1

```
3 HLA-A*26:01      A1
4 HLA-A*26:04      A1
```

```
[29]: combined_data = binding_data.join(supertype_df.set_index("allele"), on="␣
      ↪allele")
      combined_data.head()
```

```
[29]:  Epitope ID  binding information      allele  peptide sequence \
0         52886          Positive  HLA-A*01:01      QYDPVAALF
1         71650          Positive  HLA-A*01:01      VVEKQSGLY
2        150319          Positive  HLA-A*01:01      ITEAELTGY
3        141221          Positive  HLA-A*01:01      ATDSLNEY
4         24819          Positive  HLA-A*01:01      HSNLNDATY

      binding label supertype
0                1      A1
1                1      A1
2                1      A1
3                1      A1
4                1      A1
```

```
[30]: spa_data = combined_data[["supertype", " peptide sequence", " allele"]].
      ↪to_numpy()
      label_data = combined_data[" binding label"].to_numpy()
```

```
[31]: def do_cv2(C):
      kf = KFold(n_splits=10, shuffle=True)

      scores = dict()
      names = ["uniform", "dirac", "multitask"]
      for name in names:
          scores[name] = {
              "accuracy": [],
              "results": [],
              "truths": [],
              "classifiers": []
          }

      for train_indices, test_indices in kf.split(pa_data):

          Xtrain = spa_data[train_indices]
          Ytrain = label_data[train_indices]
          Xtest = spa_data[test_indices]
          Ytest = label_data[test_indices]
```

```

K_train_s = kernel(k_dirac)(Xtrain[:, 0], Xtrain[:, 0])
K_test_s = kernel(k_dirac)(Xtest[:, 0], Xtrain[:, 0])
K_train_p = kernel(k_linseq)(Xtrain[:, 1], Xtrain[:, 1])
K_test_p = kernel(k_linseq)(Xtest[:, 1], Xtrain[:, 1])

for n, k in zip(
    names,
    [k_uniform, k_dirac, k_multitask]
):
    e = svm.SVC(
        kernel="precomputed",
        C=C
    )
    K_train_a = kernel(k)(Xtrain[:, 2], Xtrain[:, 2])
    K_test_a = kernel(k)(Xtest[:, 2], Xtrain[:, 2])

    Ktrain = join_kernel_matrices(K_train_s, K_train_p, K_train_a)
    Ktest = join_kernel_matrices(K_test_s, K_test_p, K_test_a)

    e.fit(Ktrain, Ytrain)
    score = e.score(Ktest, Ytest)
    scores[n]["accuracy"].append(score)
    print(f"Model {n} (C={C}): accuracy = {score:2.2f}")

    results = e.decision_function(Ktest)
    scores[n]["results"].append(results)
    scores[n]["truths"].append(Ytest)

    scores[n]["classifiers"].append(e)

for n in scores.keys():
    scores[n]["accuracy"] = np.array(scores[n]["accuracy"]).mean()

return scores

```

```

[32]: futures = []
with ProcessPoolExecutor() as ppe:
    for C in Cs:
        futures.append((C, ppe.submit(do_cv2, C)))
scores = dict()
for C, future in futures:
    scores[C] = future.result()

```

Model uniform (C=0.01): accuracy = 0.52
Model uniform (C=0.001): accuracy = 0.52

Model uniform (C=0.1): accuracy = 0.63
 Model uniform (C=1.0): accuracy = 0.64
 Model uniform (C=0.0001): accuracy = 0.52
 Model uniform (C=10.0): accuracy = 0.62
 Model dirac (C=0.01): accuracy = 0.52
 Model dirac (C=0.001): accuracy = 0.52
 Model dirac (C=0.1): accuracy = 0.64
 Model dirac (C=1.0): accuracy = 0.63
 Model dirac (C=0.0001): accuracy = 0.52
 Model uniform (C=1000.0): accuracy = 0.57
 Model uniform (C=100.0): accuracy = 0.57
 Model dirac (C=10.0): accuracy = 0.58
 Model multitask (C=0.01): accuracy = 0.52
 Model multitask (C=0.001): accuracy = 0.52
 Model dirac (C=1000.0): accuracy = 0.54
 Model multitask (C=0.1): accuracy = 0.61
 Model multitask (C=0.0001): accuracy = 0.52
 Model dirac (C=100.0): accuracy = 0.54
 Model multitask (C=1.0): accuracy = 0.65
 Model multitask (C=10.0): accuracy = 0.60
 Model multitask (C=1000.0): accuracy = 0.55
 Model multitask (C=100.0): accuracy = 0.55
 Model uniform (C=0.01): accuracy = 0.51
 Model uniform (C=0.0001): accuracy = 0.51
 Model uniform (C=0.001): accuracy = 0.51
 Model dirac (C=0.01): accuracy = 0.51
 Model uniform (C=0.1): accuracy = 0.62
 Model dirac (C=0.0001): accuracy = 0.51
 Model uniform (C=1.0): accuracy = 0.63
 Model dirac (C=0.001): accuracy = 0.51
 Model dirac (C=0.1): accuracy = 0.60
 Model uniform (C=10.0): accuracy = 0.65
 Model dirac (C=1.0): accuracy = 0.62
 Model multitask (C=0.01): accuracy = 0.51
 Model multitask (C=0.0001): accuracy = 0.51
 Model multitask (C=0.001): accuracy = 0.51
 Model dirac (C=10.0): accuracy = 0.60
 Model multitask (C=0.1): accuracy = 0.62
 Model uniform (C=1000.0): accuracy = 0.65
 Model multitask (C=1.0): accuracy = 0.64
 Model uniform (C=100.0): accuracy = 0.65
 Model dirac (C=1000.0): accuracy = 0.60
 Model dirac (C=100.0): accuracy = 0.60
 Model multitask (C=10.0): accuracy = 0.60
 Model multitask (C=1000.0): accuracy = 0.60
 Model multitask (C=100.0): accuracy = 0.60
 Model uniform (C=0.0001): accuracy = 0.55
 Model uniform (C=0.01): accuracy = 0.55

Model uniform (C=0.1): accuracy = 0.60
 Model dirac (C=0.0001): accuracy = 0.55
 Model dirac (C=0.01): accuracy = 0.55
 Model uniform (C=0.001): accuracy = 0.55
 Model uniform (C=1.0): accuracy = 0.61
 Model dirac (C=0.1): accuracy = 0.60
 Model dirac (C=0.001): accuracy = 0.55
 Model dirac (C=1.0): accuracy = 0.56
 Model multitask (C=0.0001): accuracy = 0.55
 Model multitask (C=0.01): accuracy = 0.55
 Model uniform (C=10.0): accuracy = 0.65
 Model multitask (C=0.1): accuracy = 0.62
 Model multitask (C=0.001): accuracy = 0.55
 Model multitask (C=1.0): accuracy = 0.59
 Model dirac (C=10.0): accuracy = 0.52
 Model uniform (C=1000.0): accuracy = 0.64
 Model uniform (C=100.0): accuracy = 0.64
 Model dirac (C=1000.0): accuracy = 0.52
 Model multitask (C=10.0): accuracy = 0.55
 Model dirac (C=100.0): accuracy = 0.52
 Model multitask (C=1000.0): accuracy = 0.55
 Model multitask (C=100.0): accuracy = 0.55
 Model uniform (C=0.0001): accuracy = 0.52
 Model dirac (C=0.0001): accuracy = 0.52
 Model uniform (C=0.01): accuracy = 0.52
 Model uniform (C=0.1): accuracy = 0.68
 Model dirac (C=0.01): accuracy = 0.52
 Model uniform (C=0.001): accuracy = 0.52
 Model dirac (C=0.1): accuracy = 0.63
 Model multitask (C=0.0001): accuracy = 0.52
 Model uniform (C=1.0): accuracy = 0.58
 Model dirac (C=0.001): accuracy = 0.52
 Model multitask (C=0.01): accuracy = 0.52
 Model dirac (C=1.0): accuracy = 0.57
 Model multitask (C=0.1): accuracy = 0.67
 Model multitask (C=0.001): accuracy = 0.52
 Model uniform (C=10.0): accuracy = 0.62
 Model multitask (C=1.0): accuracy = 0.58
 Model dirac (C=10.0): accuracy = 0.56
 Model uniform (C=1000.0): accuracy = 0.62
 Model multitask (C=10.0): accuracy = 0.58
 Model uniform (C=100.0): accuracy = 0.62
 Model dirac (C=1000.0): accuracy = 0.57
 Model dirac (C=100.0): accuracy = 0.57
 Model multitask (C=1000.0): accuracy = 0.57
 Model multitask (C=100.0): accuracy = 0.57
 Model uniform (C=0.0001): accuracy = 0.52
 Model uniform (C=0.1): accuracy = 0.64

Model dirac (C=0.0001): accuracy = 0.52
 Model uniform (C=0.01): accuracy = 0.52
 Model dirac (C=0.1): accuracy = 0.62
 Model dirac (C=0.01): accuracy = 0.52
 Model uniform (C=0.001): accuracy = 0.52
 Model multitask (C=0.0001): accuracy = 0.52
 Model dirac (C=0.001): accuracy = 0.52
 Model uniform (C=1.0): accuracy = 0.58
 Model multitask (C=0.1): accuracy = 0.64
 Model multitask (C=0.01): accuracy = 0.52
 Model dirac (C=1.0): accuracy = 0.59
 Model multitask (C=0.001): accuracy = 0.52
 Model uniform (C=10.0): accuracy = 0.63
 Model multitask (C=1.0): accuracy = 0.62
 Model dirac (C=10.0): accuracy = 0.61
 Model multitask (C=10.0): accuracy = 0.62
 Model uniform (C=100.0): accuracy = 0.60
 Model uniform (C=1000.0): accuracy = 0.62
 Model dirac (C=100.0): accuracy = 0.61
 Model dirac (C=1000.0): accuracy = 0.61
 Model multitask (C=100.0): accuracy = 0.62
 Model multitask (C=1000.0): accuracy = 0.62
 Model uniform (C=0.0001): accuracy = 0.50
 Model uniform (C=0.1): accuracy = 0.63
 Model dirac (C=0.0001): accuracy = 0.50
 Model dirac (C=0.1): accuracy = 0.59
 Model uniform (C=0.01): accuracy = 0.50
 Model dirac (C=0.01): accuracy = 0.50
 Model uniform (C=0.001): accuracy = 0.50
 Model multitask (C=0.0001): accuracy = 0.50
 Model multitask (C=0.1): accuracy = 0.58
 Model dirac (C=0.001): accuracy = 0.50
 Model uniform (C=1.0): accuracy = 0.58
 Model multitask (C=0.01): accuracy = 0.50
 Model dirac (C=1.0): accuracy = 0.52
 Model multitask (C=0.001): accuracy = 0.50
 Model multitask (C=1.0): accuracy = 0.53
 Model uniform (C=10.0): accuracy = 0.56
 Model dirac (C=10.0): accuracy = 0.55
 Model multitask (C=10.0): accuracy = 0.57
 Model uniform (C=100.0): accuracy = 0.57
 Model uniform (C=1000.0): accuracy = 0.57
 Model dirac (C=100.0): accuracy = 0.57
 Model dirac (C=1000.0): accuracy = 0.57
 Model multitask (C=100.0): accuracy = 0.58
 Model multitask (C=1000.0): accuracy = 0.58
 Model uniform (C=0.0001): accuracy = 0.55
 Model uniform (C=0.1): accuracy = 0.61

Model dirac (C=0.0001): accuracy = 0.55
 Model dirac (C=0.1): accuracy = 0.61
 Model uniform (C=0.01): accuracy = 0.55
 Model dirac (C=0.01): accuracy = 0.55
 Model uniform (C=0.001): accuracy = 0.55
 Model multitask (C=0.0001): accuracy = 0.55
 Model multitask (C=0.1): accuracy = 0.61
 Model dirac (C=0.001): accuracy = 0.55
 Model multitask (C=0.01): accuracy = 0.55
 Model uniform (C=1.0): accuracy = 0.62
 Model dirac (C=1.0): accuracy = 0.60
 Model multitask (C=0.001): accuracy = 0.55
 Model multitask (C=1.0): accuracy = 0.62
 Model uniform (C=10.0): accuracy = 0.65
 Model dirac (C=10.0): accuracy = 0.62
 Model multitask (C=10.0): accuracy = 0.60
 Model uniform (C=100.0): accuracy = 0.65
 Model uniform (C=1000.0): accuracy = 0.64
 Model dirac (C=100.0): accuracy = 0.61
 Model dirac (C=1000.0): accuracy = 0.59
 Model multitask (C=100.0): accuracy = 0.59
 Model multitask (C=1000.0): accuracy = 0.58
 Model uniform (C=0.0001): accuracy = 0.62
 Model uniform (C=0.1): accuracy = 0.64
 Model dirac (C=0.0001): accuracy = 0.62
 Model dirac (C=0.1): accuracy = 0.63
 Model uniform (C=0.01): accuracy = 0.62
 Model multitask (C=0.0001): accuracy = 0.62
 Model dirac (C=0.01): accuracy = 0.62
 Model multitask (C=0.1): accuracy = 0.63
 Model uniform (C=0.001): accuracy = 0.62
 Model dirac (C=0.001): accuracy = 0.62
 Model multitask (C=0.01): accuracy = 0.62
 Model uniform (C=1.0): accuracy = 0.63
 Model multitask (C=0.001): accuracy = 0.62
 Model dirac (C=1.0): accuracy = 0.62
 Model multitask (C=1.0): accuracy = 0.62
 Model uniform (C=10.0): accuracy = 0.59
 Model dirac (C=10.0): accuracy = 0.57
 Model multitask (C=10.0): accuracy = 0.60
 Model uniform (C=1000.0): accuracy = 0.63
 Model uniform (C=100.0): accuracy = 0.63
 Model dirac (C=1000.0): accuracy = 0.60
 Model dirac (C=100.0): accuracy = 0.60
 Model multitask (C=1000.0): accuracy = 0.62
 Model multitask (C=100.0): accuracy = 0.62
 Model uniform (C=0.0001): accuracy = 0.49
 Model uniform (C=0.1): accuracy = 0.61

Model dirac (C=0.0001): accuracy = 0.49
 Model dirac (C=0.1): accuracy = 0.60
 Model uniform (C=0.01): accuracy = 0.49
 Model multitask (C=0.0001): accuracy = 0.49
 Model dirac (C=0.01): accuracy = 0.49
 Model multitask (C=0.1): accuracy = 0.61
 Model uniform (C=0.001): accuracy = 0.49
 Model dirac (C=0.001): accuracy = 0.49
 Model multitask (C=0.01): accuracy = 0.49
 Model multitask (C=0.001): accuracy = 0.49
 Model uniform (C=1.0): accuracy = 0.60
 Model dirac (C=1.0): accuracy = 0.57
 Model multitask (C=1.0): accuracy = 0.57
 Model uniform (C=10.0): accuracy = 0.53
 Model dirac (C=10.0): accuracy = 0.53
 Model multitask (C=10.0): accuracy = 0.54
 Model uniform (C=100.0): accuracy = 0.51
 Model uniform (C=1000.0): accuracy = 0.51
 Model dirac (C=100.0): accuracy = 0.53
 Model dirac (C=1000.0): accuracy = 0.53
 Model multitask (C=100.0): accuracy = 0.55
 Model multitask (C=1000.0): accuracy = 0.55
 Model uniform (C=0.0001): accuracy = 0.52
 Model uniform (C=0.1): accuracy = 0.56
 Model dirac (C=0.0001): accuracy = 0.52
 Model dirac (C=0.1): accuracy = 0.56
 Model uniform (C=0.01): accuracy = 0.52
 Model dirac (C=0.01): accuracy = 0.52
 Model multitask (C=0.0001): accuracy = 0.52
 Model multitask (C=0.1): accuracy = 0.56
 Model uniform (C=0.001): accuracy = 0.52
 Model dirac (C=0.001): accuracy = 0.52
 Model multitask (C=0.01): accuracy = 0.52
 Model multitask (C=0.001): accuracy = 0.52
 Model uniform (C=1.0): accuracy = 0.50
 Model dirac (C=1.0): accuracy = 0.54
 Model multitask (C=1.0): accuracy = 0.55
 Model uniform (C=10.0): accuracy = 0.53
 Model dirac (C=10.0): accuracy = 0.53
 Model multitask (C=10.0): accuracy = 0.54
 Model uniform (C=100.0): accuracy = 0.50
 Model uniform (C=1000.0): accuracy = 0.50
 Model dirac (C=100.0): accuracy = 0.53
 Model dirac (C=1000.0): accuracy = 0.53
 Model multitask (C=100.0): accuracy = 0.54
 Model multitask (C=1000.0): accuracy = 0.54
 Model uniform (C=10000.0): accuracy = 0.63
 Model dirac (C=10000.0): accuracy = 0.61

```

Model multitask (C=10000.0): accuracy = 0.59
Model uniform (C=10000.0): accuracy = 0.52
Model dirac (C=10000.0): accuracy = 0.54
Model multitask (C=10000.0): accuracy = 0.52
Model uniform (C=10000.0): accuracy = 0.64
Model dirac (C=10000.0): accuracy = 0.62
Model multitask (C=10000.0): accuracy = 0.64
Model uniform (C=10000.0): accuracy = 0.55
Model dirac (C=10000.0): accuracy = 0.52
Model multitask (C=10000.0): accuracy = 0.54
Model uniform (C=10000.0): accuracy = 0.60
Model dirac (C=10000.0): accuracy = 0.58
Model multitask (C=10000.0): accuracy = 0.62
Model uniform (C=10000.0): accuracy = 0.52
Model dirac (C=10000.0): accuracy = 0.57
Model multitask (C=10000.0): accuracy = 0.57
Model uniform (C=10000.0): accuracy = 0.60
Model dirac (C=10000.0): accuracy = 0.57
Model multitask (C=10000.0): accuracy = 0.58
Model uniform (C=10000.0): accuracy = 0.64
Model dirac (C=10000.0): accuracy = 0.60
Model multitask (C=10000.0): accuracy = 0.60
Model uniform (C=10000.0): accuracy = 0.60
Model dirac (C=10000.0): accuracy = 0.54
Model multitask (C=10000.0): accuracy = 0.58
Model uniform (C=10000.0): accuracy = 0.62
Model dirac (C=10000.0): accuracy = 0.61
Model multitask (C=10000.0): accuracy = 0.60

```

The values above are for the individual cross validation steps, so they appear 10 times each. We will print the averaged values next:

```

[33]: output = []
      for C, model in scores.items():
          for name, results in model.items():
              output.append(f"Model {name} (C={C}): accuracy = {results['accuracy']:2.
→2f}")
      print("\n".join(sorted(output)))

```

```

Model dirac (C=0.0001): accuracy = 0.53
Model dirac (C=0.001): accuracy = 0.53
Model dirac (C=0.01): accuracy = 0.53
Model dirac (C=0.1): accuracy = 0.61
Model dirac (C=1.0): accuracy = 0.58
Model dirac (C=10.0): accuracy = 0.57
Model dirac (C=100.0): accuracy = 0.57
Model dirac (C=1000.0): accuracy = 0.57
Model dirac (C=10000.0): accuracy = 0.58

```

```

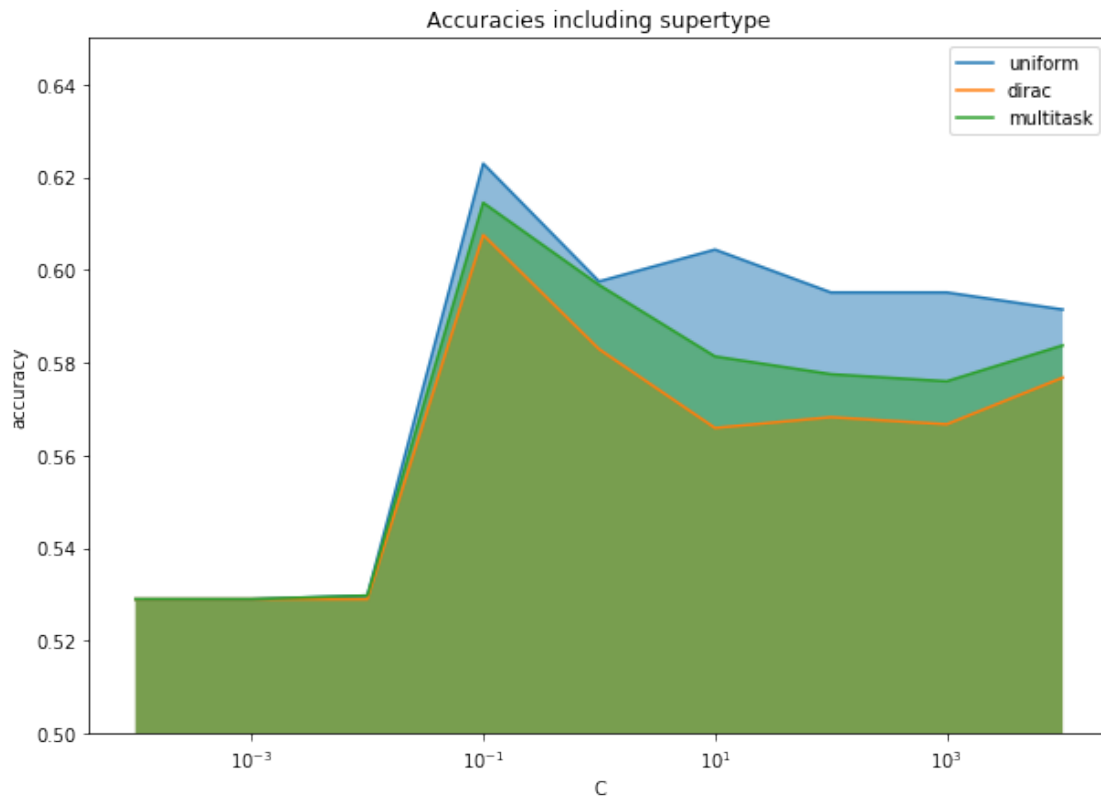
Model multitask (C=0.0001): accuracy = 0.53
Model multitask (C=0.001): accuracy = 0.53
Model multitask (C=0.01): accuracy = 0.53
Model multitask (C=0.1): accuracy = 0.61
Model multitask (C=1.0): accuracy = 0.60
Model multitask (C=10.0): accuracy = 0.58
Model multitask (C=100.0): accuracy = 0.58
Model multitask (C=1000.0): accuracy = 0.58
Model multitask (C=10000.0): accuracy = 0.58
Model uniform (C=0.0001): accuracy = 0.53
Model uniform (C=0.001): accuracy = 0.53
Model uniform (C=0.01): accuracy = 0.53
Model uniform (C=0.1): accuracy = 0.62
Model uniform (C=1.0): accuracy = 0.60
Model uniform (C=10.0): accuracy = 0.60
Model uniform (C=100.0): accuracy = 0.60
Model uniform (C=1000.0): accuracy = 0.60
Model uniform (C=10000.0): accuracy = 0.59

```

```

[34]: accuracies = scores_to_accuracy_dict(scores)
      plot_accuracies(accuracies, title="Accuracies including supertype")

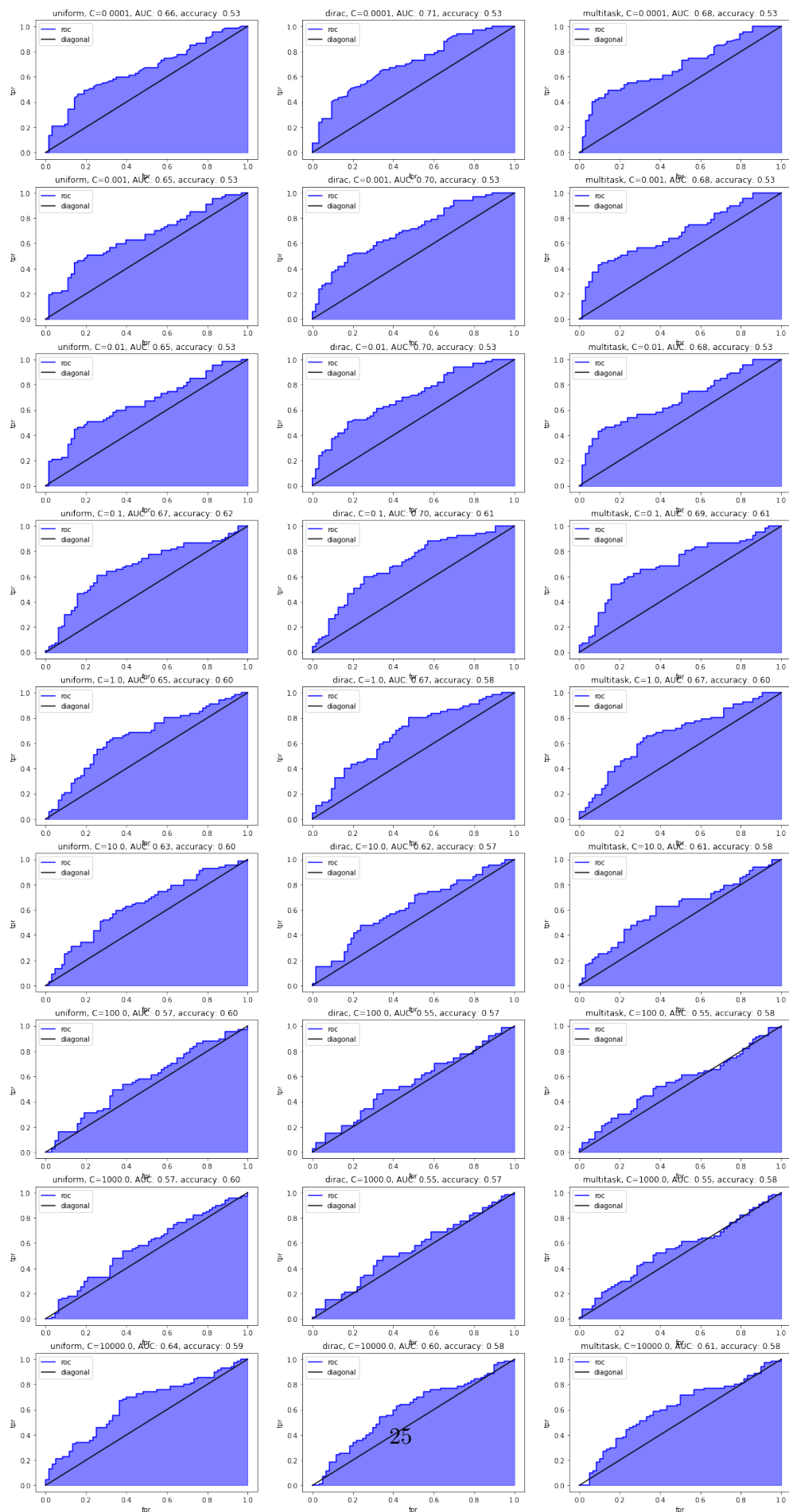
```



Already from the accuracies one can see that the supertypes really help. Starting from $C = 10^{-1}$ all the models get reasonable accuracies (not great, but significantly better than random). Previously this was not the case and some accuracies even fell below 0.5. Looking at the ROC curves (below) gives a very similar picture: With the supertypes the curves are actually better than random, whereas before they looked pretty much like random performance.

2.1 d2) ROC and AUC + e2) AUC and Accuracy

```
[35]: roc_data = calc_roc(scores)
      plot_roc(roc_data)
```

2.2 d)

First up one can clearly say, that including the supertypes really helped. Both accuracies and AUC values increased after adding the supertype. Looking at the different kernels and values of C , there is not as much difference. For small C there is very little difference between the kernels, which is to be expected, since the model is strongly regularized. In general the best C seems to be about 0.1 where all three kernels do a reasonably good job. The best model is dirac kernel, $C = 0.1$ with an AUC of 0.7 and an accuracy of 0.61. This is not really clear cut, though since the model next to it (uniform, $C = 0.1$) achieves a slightly higher accuracy (0.62), but loses on the AUC (0.67).

2.3 e)

One can see, that the accuracies and AUC values are heavily correlated. In general higher accuracy also means higher AUC. This is not always the case though. For example for iniform kernel and $C = 1000$ we get accuracy 0.6 and AUC 0.57. However for multitask $C = 10000$ we get accuracy 0.58 and AUC 0.61. This clearly shows, that accuracy can improve, while AUC get worse.

This is because accuracy and AUC measure different quantities. Accuracy measures how many predictions were correct, and does not compare in what way they are wrong if they are wrong. AUC measures the true positive rate one can get while tolerating a certain false positive rate. This is achieved by altering the constant threshold added to the result of the SVM. This threshold is however not added for the computation of accuracy and therefore these metrics are fundamentally different.

Both, however, have the property that they can range from 0 to 1 where 1 is perfect, 0.5 is as good as random and anything less than that should really not happen. Therefore they are correlated, because a good model should have high accuracy and AUC. Therefore seeing one of these values gives you an idea about the quality of the model, as well as about the other metric, because they both depend on the model being good.