

Příloha 1 – Skript pro vygenerování 1000 vstupních bodů, stratifikovaně rozdělených do tříd

```
import arcpy

arcpy.env.overwriteOutput = 1

arcpy.env.workspace = "E:/B/NaturCuni/Diplomka/vrstvy"

"lesy_dissolve"

arcpy.CreateRandomPoints_management(out_path="E:/B/NaturCuni/Diplomka/body_arcmap", out_name="body500_les",
constraining_feature_class="lesy_dissolve", constraining_extent="0 0 250 250", number_of_points_or_field="370",
minimum_allowed_distance="0 Meters",
create_multipart_output="POINT", multipart_size="0")

"body500_les"

arcpy.CalculateField_management(in_table="body500_les",
field="CID", expression="1", expression_type="VB", code_block="")

"voda_dissolve"

arcpy.CreateRandomPoints_management(out_path="E:/B/NaturCuni/Diplomka/body_arcmap", out_name="body500_voda",
constraining_feature_class="voda_dissolve", constraining_extent="0 0 250 250", number_of_points_or_field="10",
minimum_allowed_distance="0 Meters",
create_multipart_output="POINT", multipart_size="0")

"body500_voda"

arcpy.CalculateField_management(in_table="body500_voda",
field="CID", expression="5", expression_type="VB", code_block="")

"ornaBEZ_dissolve"

arcpy.CreateRandomPoints_management(out_path="E:/B/NaturCuni/Diplomka/body_arcmap", out_name="body500_orna",
constraining_feature_class="ornaBEZ_dissolve",
constraining_extent="0 0 250 250", number_of_points_or_field="130",
minimum_allowed_distance="0 Meters",
create_multipart_output="POINT", multipart_size="0")
```

```

"body500_orna"

arcpy.CalculateField_management(in_table="body500_orna",
field="CID", expression="4", expression_type="VB", code_block="")

"puda_s_vege_celkove_dissolve"

arcpy.CreateRandomPoints_management(out_path="E:/B/NaturCuni/Diplom
ka/body_arcmap", out_name="body500_ornaS",
constraining_feature_class="puda_s_vege_celkove_dissolve",
constraining_extent="0 0 250 250", number_of_points_or_field="420",
minimum_allowed_distance="0 Meters",
create_multipart_output="POINT", multipart_size="0")

"body500_ornaS"

arcpy.CalculateField_management(in_table="body500_ornaS",
field="CID", expression="3", expression_type="VB", code_block="")

"zastavba_dissolve"

arcpy.CreateRandomPoints_management(out_path="E:/B/NaturCuni/Diplom
ka/body_arcmap", out_name="body500_zastavba",
constraining_feature_class="zastavba_dissolve",
constraining_extent="0 0 250 250", number_of_points_or_field="70",
minimum_allowed_distance="0 Meters",
create_multipart_output="POINT", multipart_size="0")

"body500_zastavba"

arcpy.CalculateField_management(in_table="body500_zastavba",
field="CID", expression="2", expression_type="VB", code_block="")

arcpy.Merge_management(inputs="body500_les;body500_zastavba;body500
_ornaS;body500_orna;body500_voda",
output="E:/B/NaturCuni/Diplomka/body_arcmap/body500_vse.shp",
field_mappings='CID "CID" true true false 10 Long 0 10
,First,#,body500_les,CID,-1,-1,body500_zastavba,CID,-1,-
1,body500_ornaS,CID,-1,-1,body500_orna,CID,-1,-1,body500_voda,CID,-
1,-1')

```

PŘÍLOHA 2 – script pro vytvoření klasifikace Maximum Likelihood

```
import arcpy, random, sys

pocet_trenovacich = int(sys.argv[1])
pokus = int(sys.argv[2])

arcpy.env.overwriteOutput = 1
cesta = "C:/Users/Edmyon/Desktop/Diplomka"
database = cesta + "/validace.gdb"
dataset = database + "/UTM33"
arcpy.env.workspace = dataset

zakladni_body = "body1000_vse"
grnd_true_attrib = "GrndTruth"

#psti = [0.371754081, 0.071711226, 0.423855961, 0.131571104,
0.001107627]
psti = [0.37, 0.07, 0.42, 0.13, 0.01]

def select_random(fc, attrib, value, count, output):
    # Vybere z $fc nahodnych $count zaznamu takovych, ze maji
    hodnotu atributu $attrib rovnu $value a ulozi se jako $output.
```

```

# pridame pole

try:
    arcpy.AddField_management(fc, "SEL", "SHORT")
except:
    pass

arcpy.CalculateField_management(fc, "SEL", 0)

# zjistime, kolik je zaznamu splnujicich podminku

arcpy.MakeFeatureLayer_management(fc, "tmp", attrib + "=" +
str(value))

res = arcpy.GetCount_management("tmp")

avail_cnt = int(res.getOutput(0))

arcpy.Delete_management("tmp")

if count > avail_cnt:

    # chci jich vybrat vic, nez jich je

    indices = avail_cnt * [1]

    # vyberu vsechny

else:

    sel_idx = random.sample(range(0, avail_cnt), count)

    indices = avail_cnt * [0]

    for i in sel_idx:

        indices[i] = 1

        # pole priznaku: ma-li byt vybran, bude v nem 1

```

```

cur = arcpy.UpdateCursor(fc, attrib + "=" + str(value))

i = 0

for row in cur:

    row.setValue("SEL", indices[i])

    cur.updateRow(row)

    i += 1

del cur

arcpy.MakeFeatureLayer_management(fc, "tmp")

arcpy.SelectLayerByAttribute_management("tmp", "NEW_SELECTION",
"SEL = 1")

arcpy.CopyFeatures_management("tmp", output)

arcpy.Delete_management("tmp")

arcpy.DeleteField_management(output, "SEL")

arcpy.DeleteField_management(fc, "SEL")

def get_table_value(table, row_no, column_name):

    # Vybere z tabulky $table hodnotu na radku $row_no (cislovani
radku od 1) a ve sloupci s nazvem $column_name

    cur = arcpy.SearchCursor(table)

    i = 1

    while i <= row_no:

        row = cur.next()

        i += 1

    ret_val = row.getValue(column_name)

```

```

del cur

return ret_val

#program zacina tady

try:

    arcpy.CheckOutExtension("spatial")

    select_random(zakladni_body, grnd_true_attrib, 1,
int(round(pocet_trenovacich * psti[0])), "tren_clas_1")

    select_random(zakladni_body, grnd_true_attrib, 2,
int(round(pocet_trenovacich * psti[1])), "tren_clas_2")

    select_random(zakladni_body, grnd_true_attrib, 4,
int(round(pocet_trenovacich * psti[3])), "tren_clas_4")

    select_random(zakladni_body, grnd_true_attrib, 5,
int(round(pocet_trenovacich * psti[4])), "tren_clas_5")

arcpy.Merge_management("tren_clas_1;tren_clas_2;tren_clas_4;tren_cl
as_5", "trenovaci_mezikrok")

pocet = arcpy.GetCount_management("trenovaci_mezikrok")

pocet_int = int(float(pocet.getOutput(0)))

odecist = pocet_trenovacich - pocet_int

select_random(zakladni_body, grnd_true_attrib, 3, odecist,
"tren_clas_vygen3")

arcpy.Merge_management("tren_clas_vygen3;trenovaci_mezikrok",
"trenovaci")

```

```

    arcpy.Erase_analysis(zakladni_body,"trenovaci", cesta +
"/validacni.shp", 0.10)

    arcpy.gp.CreateSignatures_sa(database + "/snimek", "trenovaci",
cesta + "/signature_body.GSG", "COVARIANCE", grnd_true_attrib)

    arcpy.gp.MLClassify_sa(database + "/snimek", cesta +
"/signature_body.GSG", database + "/klas_snimek", "0.0", "EQUAL",
"", "")

    arcpy.gp.UpdateAccuracyAssessmentPoints_sa(database +
"/klas_snimek", cesta + "/validacni.shp", cesta +
"/validacni_update.shp", "CLASSIFIED")

    arcpy.gp.ComputeConfusionMatrix_sa(cesta +
"/validacni_update.shp", database + "/matice")

    kappa = get_table_value(database + "/matice", 8, "Kappa")

    celkova_presnost = get_table_value(database + "/matice", 7,
"U_Accuracy")

    zpracovatelska_presnost1 = get_table_value(database +
"/matice", 7, "C_1")

    zpracovatelska_presnost2 = get_table_value(database +
"/matice", 7, "C_2")

    zpracovatelska_presnost3 = get_table_value(database +
"/matice", 7, "C_3")

    zpracovatelska_presnost4 = get_table_value(database +
"/matice", 7, "C_4")

```

```
zpracovatelska_presnost5 = get_table_value(database +
"/matice", 7, "C_5")

uzivatelska_presnost1 = get_table_value(database + "/matice",
1, "U_Accuracy")

uzivatelska_presnost2 = get_table_value(database + "/matice",
2, "U_Accuracy")

uzivatelska_presnost3 = get_table_value(database + "/matice",
3, "U_Accuracy")

uzivatelska_presnost4 = get_table_value(database + "/matice",
4, "U_Accuracy")

uzivatelska_presnost5 = get_table_value(database + "/matice",
5, "U_Accuracy")

print pocet_trenovacich, pokus, kappa, celkova_presnost,
zpracovatelska_presnost1, zpracovatelska_presnost2,
zpracovatelska_presnost3, zpracovatelska_presnost4,
zpracovatelska_presnost5, uzivatelska_presnost1,
uzivatelska_presnost2, uzivatelska_presnost3,
uzivatelska_presnost4, uzivatelska_presnost5

except:

    print pocet_trenovacich, pokus, " chyba"
```


PŘÍLOHA 3 – script pro vytvoření klasifikace SVM

```
import arcpy, random, sys

pocet_trenovacich = int(sys.argv[1])
pokus = int(sys.argv[2])

arcpy.env.overwriteOutput = 1
cesta = "C:\Users\Edmyon\Desktop\Diplomka\data"
arcpy.env.workspace = cesta

zakladni_body = "body1000_vse.shp"
grnd_true_attrib = "GrndTruth"

#psti = [0.371754081, 0.071711226, 0.423855961, 0.131571104,
0.001107627]
psti = [0.37, 0.07, 0.42, 0.13, 0.01]

def select_random(fc, attrib, value, count, output):
    # Vybere z $fc nahodnych $count zaznamu takovych, ze maji hodnotu
    atributu $attrib rovnu $value a ulozi se jako $output.

    # pridame pole
    try:
        arcpy.AddField_management(fc, "SEL", "SHORT")
    except:
        pass

    arcpy.CalculateField_management(fc, "SEL", 0)
    # zjistime, kolik je zaznamu splnujicich podminku
    arcpy.MakeFeatureLayer_management(fc, "tmp", attrib + "=" +
str(value))
    res = arcpy.GetCount_management("tmp")
    avail_cnt = int(res.getOutput(0))
    arcpy.Delete_management("tmp")

    if count > avail_cnt:
        # chci jich vybrat vic, nez jich je
        indices = avail_cnt * [1]
        # vyberu vsechny

    else:
        sel_idx = random.sample(range(0, avail_cnt), count)
        indices = avail_cnt * [0]
        for i in sel_idx:
            indices[i] = 1
            # pole priznaku: ma-li byt vybrany, bude v nem 1

    cur = arcpy.UpdateCursor(fc, attrib + "=" + str(value))
    i = 0
    for row in cur:
        row.setValue("SEL", indices[i])
        cur.updateRow(row)
```

```

        i += 1
    del cur

    arcpy.MakeFeatureLayer_management(fc, "tmp")
    arcpy.SelectLayerByAttribute_management("tmp", "NEW_SELECTION",
"SEL = 1")
    arcpy.CopyFeatures_management("tmp", output)
    arcpy.Delete_management("tmp")

    arcpy.DeleteField_management(output, "SEL")
    arcpy.DeleteField_management(fc, "SEL")

def get_table_value(table, row_no, column_name):
    # Vybere z tabulky $table hodnotu na radku $row_no (cislovani
radku od 1) a ve sloupci s nazvem $column_name

    cur = arcpy.SearchCursor(table)
    i = 1
    while i <= row_no:
        row = cur.next()
        i += 1

    ret_val = row.getValue(column_name)
    del cur

    return ret_val

#program zacina tady
try:
    arcpy.CheckOutExtension("spatial")

    select_random(zakladni_body, grnd_true_attrib, 1,
int(round(pocet_trenovacich * psti[0])), "tren_clas_1.shp")
    select_random(zakladni_body, grnd_true_attrib, 2,
int(round(pocet_trenovacich * psti[1])), "tren_clas_2.shp")
    select_random(zakladni_body, grnd_true_attrib, 4,
int(round(pocet_trenovacich * psti[3])), "tren_clas_4.shp")
    select_random(zakladni_body, grnd_true_attrib, 5,
int(round(pocet_trenovacich * psti[4])), "tren_clas_5.shp")

    arcpy.Merge_management(["tren_clas_1.shp", "tren_clas_2.shp",
"tren_clas_4.shp", "tren_clas_5.shp"], "trenovaci_mezikrok.shp")

    pocet = arcpy.GetCount_management("trenovaci_mezikrok.shp")
    pocet_int = int(float(pocet.getOutput(0)))
    odecist = pocet_trenovacich - pocet_int

    select_random(zakladni_body, grnd_true_attrib, 3, odecist,
"tren_clas_vygen3.shp")
    arcpy.Merge_management(["tren_clas_vygen3.shp",
"trenovaci_mezikrok.shp"], "trenovaci.shp")

    arcpy.Erase_analysis(zakladni_body, "trenovaci.shp",
"validacni.shp", 0.10)

    # varianta 1 -- vyuziva segmentaci

```

```

# seg_raster = arcpy.gp.SegmentMeanShift_sa("snimek.img",
"snimek_sms.img", 18, 18, 20)
#
arcpy.gp.TrainSupportVectorMachineClassifier_sa("snimek_sms.img",
"trenovaci.shp", "svm_params.ecd", "", "", "COLOR;MEAN")
# classified_raster = arcpy.sa.ClassifyRaster("snimek_sms.img",
"svm_params.ecd")
# classified_raster.save("snimek_classified.img")

# varianta 2 -- nevyuziva segmentaci
arcpy.gp.TrainSupportVectorMachineClassifier_sa("snimek.img",
"trenovaci.shp", "svm_params.ecd", "", "", "COLOR;MEAN")
classified_raster = arcpy.sa.ClassifyRaster("snimek.img",
"svm_params.ecd")
classified_raster.save("snimek_classified.img")
# vyberete si variantu 1 nebo variantu 2 a tu druhou
smazete/zakomentujete

arcpy.gp.UpdateAccuracyAssessmentPoints_sa("snimek_classified.img",
"validacni.shp", "validacni_update.shp", "CLASSIFIED")

#"validacni_1_update"
arcpy.gp.ComputeConfusionMatrix_sa("validacni_update.shp",
"matice.dbf")

kappa = get_table_value("matice.dbf", 8, "Kappa")
celkova_presnost = get_table_value("matice.dbf", 7, "U_Accuracy")
zpracovatelska_presnost1 = get_table_value("matice.dbf", 7, "C_1")
zpracovatelska_presnost2 = get_table_value("matice.dbf", 7, "C_2")
zpracovatelska_presnost3 = get_table_value("matice.dbf", 7, "C_3")
zpracovatelska_presnost4 = get_table_value("matice.dbf", 7, "C_4")
zpracovatelska_presnost5 = get_table_value("matice.dbf", 7, "C_5")
uzivatelska_presnost1 = get_table_value("matice.dbf", 1,
"U_Accuracy")
uzivatelska_presnost2 = get_table_value("matice.dbf", 2,
"U_Accuracy")
uzivatelska_presnost3 = get_table_value("matice.dbf", 3,
"U_Accuracy")
uzivatelska_presnost4 = get_table_value("matice.dbf", 4,
"U_Accuracy")
uzivatelska_presnost5 = get_table_value("matice.dbf", 5,
"U_Accuracy")

print pocet_trenovacich, pokus, kappa, celkova_presnost,
zpracovatelska_presnost1, zpracovatelska_presnost2,
zpracovatelska_presnost3, zpracovatelska_presnost4,
zpracovatelska_presnost5, uzivatelska_presnost1,
uzivatelska_presnost2, uzivatelska_presnost3, uzivatelska_presnost4,
uzivatelska_presnost5
except:
print pocet_trenovacich, pokus, " chyba"

```

PŘÍLOHA 4 – script pro vytvoření klasifikace MLC s úpravou množství validačních bodů

```
import arcpy, random, sys

pocet_trenovacich = int(sys.argv[1])
pokus = int(sys.argv[2])
pocet_validacnich = int(sys.argv[3])

arcpy.env.overwriteOutput = 1
cesta = "C:/Users/Edmyon/Desktop/Diplomka"
databaze = cesta + "/validace.gdb"
dataset = databaze + "/UTM33"
arcpy.env.workspace = dataset

zakladni_body = "body1000_vse"
grnd_true_attrib = "GrndTruth"

#psti = [0.371754081, 0.071711226, 0.423855961, 0.131571104,
0.001107627]
psti = [0.37, 0.07, 0.42, 0.13, 0.01]

def select_random(fc, attrib, value, count, output):
    # Vybere z $fc nahodnych $count zaznamu takovych, ze maji hodnotu
    atributu $attrib rovnu $value a ulozi se jako $output.

    # pridame pole
    try:
        arcpy.AddField_management(fc, "SEL", "SHORT")
    except:
        pass

    arcpy.CalculateField_management(fc, "SEL", 0)
    # zjistime, kolik je zaznamu splnujicich podminku
    arcpy.MakeFeatureLayer_management(fc, "tmp", attrib + "=" +
str(value))
    res = arcpy.GetCount_management("tmp")
    avail_cnt = int(res.getOutput(0))
    arcpy.Delete_management("tmp")

    if count > avail_cnt:
        # chci jich vybrat vic, nez jich je
        indices = avail_cnt * [1]
        # vyberu vsechny
    else:
        sel_idx = random.sample(range(0, avail_cnt), count)
        indices = avail_cnt * [0]
        for i in sel_idx:
            indices[i] = 1
            # pole priznaku: ma-li byt vybrany, bude v nem 1

    cur = arcpy.UpdateCursor(fc, attrib + "=" + str(value))
```

```

i = 0
for row in cur:
    row.setValue("SEL", indices[i])
    cur.updateRow(row)
    i += 1
del cur

arcpy.MakeFeatureLayer_management(fc, "tmp")
arcpy.SelectLayerByAttribute_management("tmp", "NEW_SELECTION",
"SEL = 1")
arcpy.CopyFeatures_management("tmp", output)
arcpy.Delete_management("tmp")

arcpy.DeleteField_management(output, "SEL")
arcpy.DeleteField_management(fc, "SEL")

def get_table_value(table, row_no, column_name):
    # Vybere z tabulky $table hodnotu na radku $row_no (cislovani
radku od 1) a ve sloupci s nazvem $column_name

    cur = arcpy.SearchCursor(table)
    i = 1
    while i <= row_no:
        row = cur.next()
        i += 1

    ret_val = row.getValue(column_name)
    del cur

    return ret_val

#program zacina tady

arcpy.CheckOutExtension("spatial")

select_random(zakladni_body, grnd_true_attrib, 1,
int(round(pocet_trenovacich * psti[0])), "tren_clas_1")
select_random(zakladni_body, grnd_true_attrib, 2,
int(round(pocet_trenovacich * psti[1])), "tren_clas_2")
select_random(zakladni_body, grnd_true_attrib, 4,
int(round(pocet_trenovacich * psti[3])), "tren_clas_4")
select_random(zakladni_body, grnd_true_attrib, 5, max(4,
int(round(pocet_trenovacich * psti[4]))), "tren_clas_5")

arcpy.Merge_management("tren_clas_1;tren_clas_2;tren_clas_4;tren_clas_
5", "trenovaci_mezikrok")

pocet = arcpy.GetCount_management("trenovaci_mezikrok")
pocet_int = int(float(pocet.getOutput(0)))
odecist = pocet_trenovacich - pocet_int

select_random(zakladni_body, grnd_true_attrib, 3, odecist,
"tren_clas_vygen3")
arcpy.Merge_management("tren_clas_vygen3;trenovaci_mezikrok",
"trenovaci")

#tady zacnu vybirat validacni

```

```

select_random(zakladni_body, grnd_true_attrib, 1,
int(round(pocet_validacnich * psti[0])), "vali_clas_1")
select_random(zakladni_body, grnd_true_attrib, 2,
int(round(pocet_validacnich * psti[1])), "vali_clas_2")
select_random(zakladni_body, grnd_true_attrib, 4,
int(round(pocet_validacnich * psti[3])), "vali_clas_4")
select_random(zakladni_body, grnd_true_attrib, 5, max(4,
int(round(pocet_validacnich * psti[4]))), "vali_clas_5")

arcpy.Merge_management("vali_clas_1;vali_clas_2;vali_clas_4;vali_clas_5", "validacni_mezikrok")

pocet_vali = arcpy.GetCount_management("validacni_mezikrok")
pocet_int2 = int(float(pocet_vali.getOutput(0)))
odecist2 = pocet_validacnich - pocet_int2

select_random(zakladni_body, grnd_true_attrib, 3, odecist2,
"vali_clas_vygen3")
arcpy.Merge_management("vali_clas_vygen3;validacni_mezikrok", cesta +
"/validacni.shp")

#tady zacina klasifikace

arcpy.gp.CreateSignatures_sa(database + "/snimek", "trenovaci", cesta
+ "/signature_body.GSG", "COVARIANCE", grnd_true_attrib)

arcpy.gp.MLClassify_sa(database + "/snimek", cesta +
"/signature_body.GSG", database + "/klas_snimek", "0.0", "EQUAL", "",
"")

#hodnoceni presnosti
arcpy.gp.UpdateAccuracyAssessmentPoints_sa(database + "/klas_snimek",
cesta + "/validacni.shp", cesta + "/validacni_update.shp",
"CLASSIFIED")

#chybova matice
arcpy.gp.ComputeConfusionMatrix_sa(cesta + "/validacni_update.shp",
database + "/matice")

kappa = get_table_value(database + "/matice", 8, "Kappa")
celkova_presnost = get_table_value(database + "/matice", 7,
"U_Accuracy")
zpracovatelska_presnost1 = get_table_value(database + "/matice", 7,
"C_1")
zpracovatelska_presnost2 = get_table_value(database + "/matice", 7,
"C_2")
zpracovatelska_presnost3 = get_table_value(database + "/matice", 7,
"C_3")
zpracovatelska_presnost4 = get_table_value(database + "/matice", 7,
"C_4")
zpracovatelska_presnost5 = get_table_value(database + "/matice", 7,
"C_5")
uzivatelska_presnost1 = get_table_value(database + "/matice", 1,
"U_Accuracy")
uzivatelska_presnost2 = get_table_value(database + "/matice", 2,
"U_Accuracy")

```

```
uzivatelska_presnost3 = get_table_value(database + "/matice", 3,  
"U_Accuracy")  
uzivatelska_presnost4 = get_table_value(database + "/matice", 4,  
"U_Accuracy")  
uzivatelska_presnost5 = get_table_value(database + "/matice", 5,  
"U_Accuracy")  
  
print pocet_trenovacich, pokus, kappa, celkova_presnost,  
zpracovatelska_presnost1, zpracovatelska_presnost2,  
zpracovatelska_presnost3, zpracovatelska_presnost4,  
zpracovatelska_presnost5, uzivatelska_presnost1,  
uzivatelska_presnost2, uzivatelska_presnost3, uzivatelska_presnost4,  
uzivatelska_presnost5
```