Learning

March 28, 2021

1 KM2

Poniższy kamień milowy przedstawia inżynierię danych oraz wstępne modelowanie dla danych "gender_voice_dataset.csv", które przedstawiają dane statystyczne nagrań głosowych różnych ludzi. Zadaniem jest klasyfikacja kolumny 'label', która przedstawia płeć osoby mówiącej.

1.1 Imports

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelBinarizer, PolynomialFeatures
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split, StratifiedKFold,␣
↪cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, precision_score, recall_score,␣
↪f1_score, roc_auc_score
```

1.2 Data upload and basic transforms

[2]:
```
%precision %.6f
data = pd.read_csv("gender_voice_dataset.csv")
LB = LabelBinarizer()
data["label"] = LB.fit_transform(data["label"])
y = np.array(data["label"])
X = data.drop(['label'],axis=1)
X.columns
```

[2]:
```
Index(['meanfreq', 'sd', 'median', 'Q25', 'Q75', 'IQR', 'skew', 'kurt',
       'sp.ent', 'sfm', 'mode', 'centroid', 'meanfun', 'minfun', 'maxfun',
       'meandom', 'mindom', 'maxdom', 'dfrange', 'modindx'],
      dtype='object')
```

[3]:
```
#from google.colab import drive
#drive.mount('/content/drive')
```

[4]:
```
print(len(X))
X.head()
```

```
3168
```

[4]:

	meanfreq	sd	median	Q25	Q75	IQR	skew	kurt	sp.ent	sfm	mode	centroid	meanfun	minfun	maxfun	meandom	mindom	maxdom	dfrange	modindx
0	0.059781	0.064241	0.032027	0.015071	0.090193	0.075122	12.863462	274.402905	0.893369	0.491918	0.000000	0.059781	0.084279	0.015702	0.275862	0.007812	0.007812	0.007812	0.000000	0.000000
1	0.066009	0.067310	0.040229	0.019414	0.092666	0.073252	22.423285	634.613855	0.892193	0.513724	0.000000	0.066009	0.107937	0.015826	0.250000	0.009014	0.007812	0.054688	0.046875	0.052632
2	0.077316	0.083829	0.036718	0.008701	0.131908	0.123207	30.757155	1024.927705	0.846389	0.478905	0.000000	0.077316	0.098706	0.015656	0.271186	0.007990	0.007812	0.015625	0.007812	0.046512
3	0.151228	0.072111	0.158011	0.096582	0.207955	0.111374	1.232831	4.177296	0.963322	0.727232	0.083878	0.151228	0.088965	0.017798	0.250000	0.201497	0.007812	0.562500	0.554688	0.247119
4	0.135120	0.079146	0.124656	0.078720	0.206045	0.127325	1.101174	4.333713	0.971955	0.783568	0.104261	0.135120	0.106398	0.016931	0.266667	0.712812	0.007812	5.484375	5.476562	0.208274

[5]:
```
y
```

[5]:
```
array([1, 1, 1, …, 0, 0, 0])
```

1 to mężyczyzni, 0 to kobiety.

1.3 Feature Engineering

Najpierw sprawdzamy prosty klasyfikator na już posiadanych cechach, by znaleźć numeryczną "ważność" cech.

[6]:
```
def split(X, y):
  X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.33,␣
↪random_state=42, stratify=y)
  X_val, X_test, y_val, y_test = train_test_split(X_val, y_val, stratify=y_val,␣
↪test_size=0.3, random_state=42)
  return X_train, X_val, X_test, y_train, y_val, y_test
```

[7]:
```
def plot_features(X_plot, importances_plot, indices_plot, std_plot):
  plt.figure(figsize=(16,16))
  plt.title("Feature importances")
  plt.bar(range(X_plot.shape[1]), importances_plot[indices_plot],
          color="r", yerr=std_plot[indices_plot], align="center")
  plt.xticks(range(X_plot.shape[1]), indices_plot)
  plt.xlim([-1, X_plot.shape[1]])
  plt.show()
  return
```

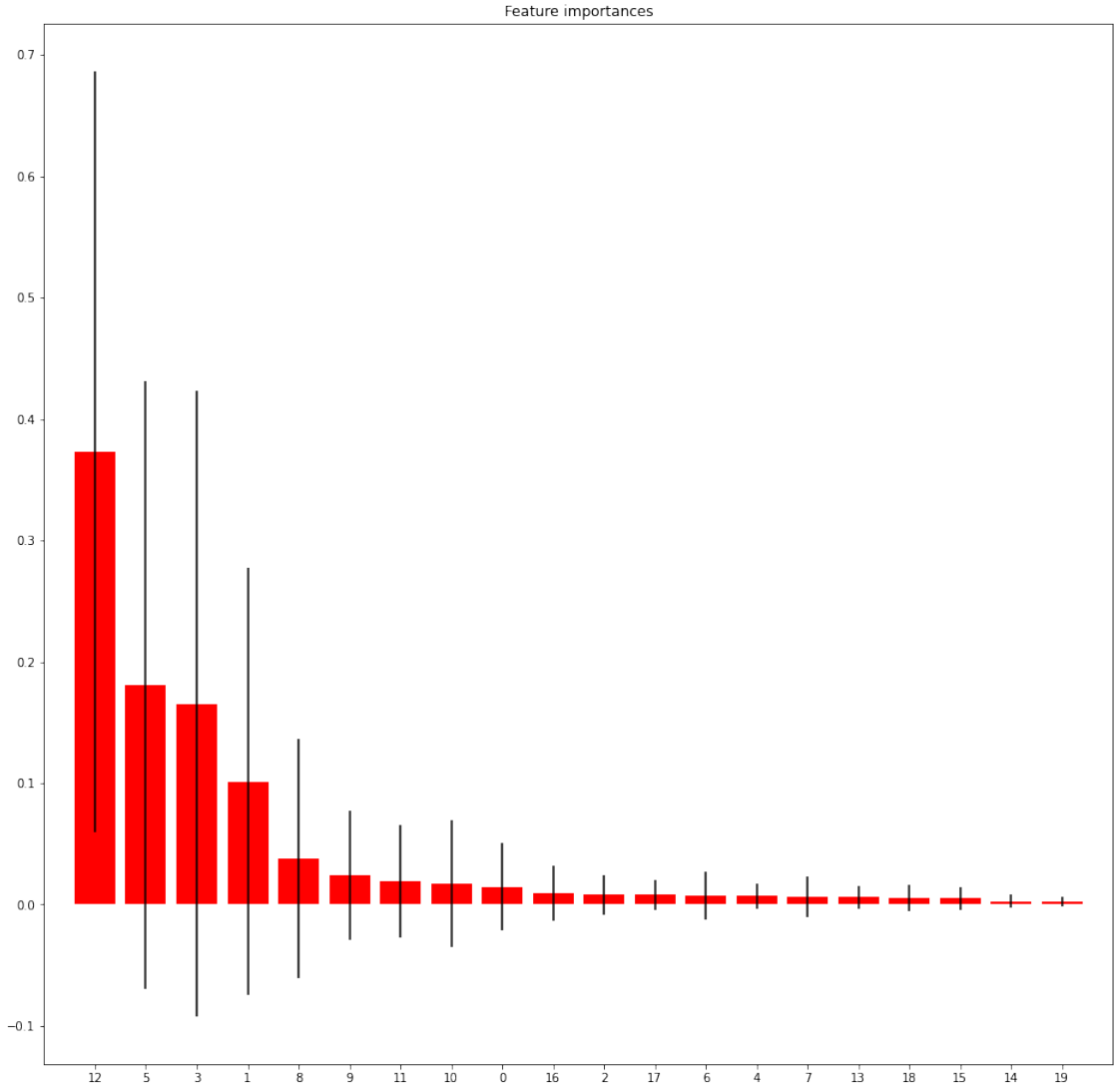
[8]:
```
def feature_importance(X_prim, y_prim):
  X_train, X_val, X_test, y_train, y_val, y_test = split(X_prim, y_prim)
  forest = RandomForestClassifier(n_estimators=100, random_state=3, max_depth=5)
  forest.fit(X_train, y_train)
  print(f"Forest score: {forest.score(X_val, y_val)}")
  importances = forest.feature_importances_
  std = np.std([tree.feature_importances_ for tree in forest.estimators_],
               axis=0)
  indices = np.argsort(importances)[::-1]

  # Print the feature ranking
  print("Feature ranking:")
  z = []
  for f in range(X_prim.shape[1]):
      z.append(X_prim.columns[indices[f]])
      print(f"{f}. Feature: {X_prim.columns[indices[f]]}␣
 ↪({importances[indices[f]]})" )
  print("Features in order of decreasing importance:")
  print(z)
  plot_features(X_prim, importances, indices, std)
  return
```

[9]:
```
X_train, X_val, X_test, y_train, y_val, y_test = split(X, y)
```

[10]:
```
feature_importance(X, y)
```

```
Forest score: 0.976775956284153
Feature ranking:
0. Feature: meanfun (0.3730480606041995)
1. Feature: IQR (0.18068260670112893)
2. Feature: Q25 (0.1654106623519519)
3. Feature: sd (0.10138369560814825)
4. Feature: sp.ent (0.03764084212078971)
5. Feature: sfm (0.024211412677322453)
6. Feature: centroid (0.01925311587804616)
7. Feature: mode (0.017082554571492412)
8. Feature: meanfreq (0.014620150930463418)
```

```
9. Feature: mindom (0.009127135173735814)
10. Feature: median (0.008139557390557285)
11. Feature: maxdom (0.007797658429642278)
12. Feature: skew (0.007142328092043546)
13. Feature: Q75 (0.006881542475392338)
14. Feature: kurt (0.00625604272119457)
15. Feature: minfun (0.0059435120337111205)
16. Feature: dfrange (0.005569110241437829)
17. Feature: meandom (0.004914936254316448)
18. Feature: maxfun (0.002564453405151461)
19. Feature: modindx (0.0023306223392745877)
Features in order of decreasing importance:
['meanfun', 'IQR', 'Q25', 'sd', 'sp.ent', 'sfm', 'centroid', 'mode', 'meanfreq',
'mindom', 'median', 'maxdom', 'skew', 'Q75', 'kurt', 'minfun', 'dfrange',
'meandom', 'maxfun', 'modindx']
```



Teraz sprawdzamy rozszerzając dane o wymnożenia wszystkich kombinacji dwóch cech. (Polynomial features)

[11]:
```
poly = PolynomialFeatures(degree=2)
X2 = poly.fit_transform(X)
X2 = pd.DataFrame(X2, columns = poly.get_feature_names(X.columns))

feature_importance(X2, y)
```

```
Forest score: 0.9808743169398907
Feature ranking:
0. Feature: meanfun^2 (0.11206663528082693)
1. Feature: meanfun (0.08173151241406311)
2. Feature: sp.ent meanfun (0.07870896943321555)
3. Feature: meanfreq IQR (0.05784046327309138)
4. Feature: meanfun maxfun (0.052545351744354596)
5. Feature: IQR^2 (0.04033255748737141)
6. Feature: Q75 IQR (0.03764606423774803)
7. Feature: IQR (0.03685819983725974)
8. Feature: Q25 meanfun (0.035103655791838675)
9. Feature: IQR sp.ent (0.03489305955044352)
10. Feature: sd IQR (0.033578007721619424)
11. Feature: median IQR (0.026553387010480812)
12. Feature: sd meanfun (0.023318836647589797)
13. Feature: Q25^2 (0.021162297288636053)
14. Feature: meanfreq Q25 (0.020744521575336145)
15. Feature: IQR centroid (0.02029828752135873)
16. Feature: centroid meanfun (0.01792774321739263)
17. Feature: sd Q75 (0.015738668482412364)
18. Feature: sfm meanfun (0.015730877945356905)
19. Feature: IQR maxfun (0.014567079162893414)
20. Feature: IQR sfm (0.014547210321033312)
21. Feature: Q75 meanfun (0.014353554096713768)
22. Feature: meanfreq meanfun (0.013994816283802707)
23. Feature: Q25 centroid (0.013895193056687636)
24. Feature: Q25 (0.013759594910701535)
25. Feature: Q25 maxfun (0.011267464640811388)
26. Feature: median meanfun (0.009661040877450055)
27. Feature: Q25 sp.ent (0.00885175273612518)
28. Feature: sd^2 (0.006027916594098504)
29. Feature: sd sp.ent (0.005261822419667841)
30. Feature: sd (0.005234510480801986)
31. Feature: IQR meanfun (0.0048987135294551246)
32. Feature: Q25 IQR (0.0045111369940500886)
33. Feature: meanfun minfun (0.004368942894248351)
34. Feature: median Q25 (0.004117010626104901)
```

35. Feature: IQR skew (0.0036238187177308505)
36. Feature: IQR modindx (0.003468038716114985)
37. Feature: Q25 sfm (0.002791826361985843)
38. Feature: Q25 skew (0.002576761878160061)
39. Feature: Q25 Q75 (0.0024401396495517904)
40. Feature: IQR minfun (0.0021956845006441154)
41. Feature: meanfun mindom (0.0019657450928548315)
42. Feature: meanfreq^2 (0.0018474006233018846)
43. Feature: Q25 kurt (0.001841438416612175)
44. Feature: meanfreq sd (0.0014154328305713369)
45. Feature: sd maxfun (0.0014140883175226282)
46. Feature: sd Q25 (0.0013955286220398541)
47. Feature: centroid^2 (0.0013270187724792135)
48. Feature: skew meanfun (0.0012747372150088117)
49. Feature: IQR mindom (0.0010183978317474056)
50. Feature: sfm maxfun (0.0010135672839953814)
51. Feature: Q75 sp.ent (0.0010108104466420172)
52. Feature: Q25 meandom (0.0010011052813253382)
53. Feature: sfm^2 (0.0009192886131967497)
54. Feature: IQR mode (0.0008116876962905084)
55. Feature: sfm (0.0007943134856401267)
56. Feature: Q75 (0.000773199763635525)
57. Feature: IQR dfrange (0.0007636194152981328)
58. Feature: meanfreq (0.0007571685104617355)
59. Feature: median sp.ent (0.0007571020648001772)
60. Feature: median minfun (0.0007523897535792101)
61. Feature: sd median (0.0007521433719692118)
62. Feature: sfm modindx (0.0007499383434554584)
63. Feature: sp.ent maxdom (0.0007443194189513148)
64. Feature: IQR maxdom (0.000692077974035435)
65. Feature: mode^2 (0.0006508526805164004)
66. Feature: Q75 sfm (0.0006501027967442159)
67. Feature: median sfm (0.0006433918129247483)
68. Feature: Q25 mindom (0.0006379983537332326)
69. Feature: Q25 maxdom (0.0006267385911353285)
70. Feature: median maxfun (0.000616289520861694)
71. Feature: meanfreq maxdom (0.000608622138804346)
72. Feature: sfm mindom (0.0005883826018896103)
73. Feature: Q75^2 (0.0005824092663690461)
74. Feature: mode (0.000574231476327858)
75. Feature: median centroid (0.000547665033149378)
76. Feature: mindom^2 (0.0005422344467891154)
77. Feature: Q75 maxdom (0.0005366458431241033)
78. Feature: sp.ent sfm (0.000534634060812765)
79. Feature: skew modindx (0.0005339077892922519)
80. Feature: centroid dfrange (0.0005217517118331545)
81. Feature: median modindx (0.0005153658906491125)
82. Feature: minfun mindom (0.0005107530679255686)

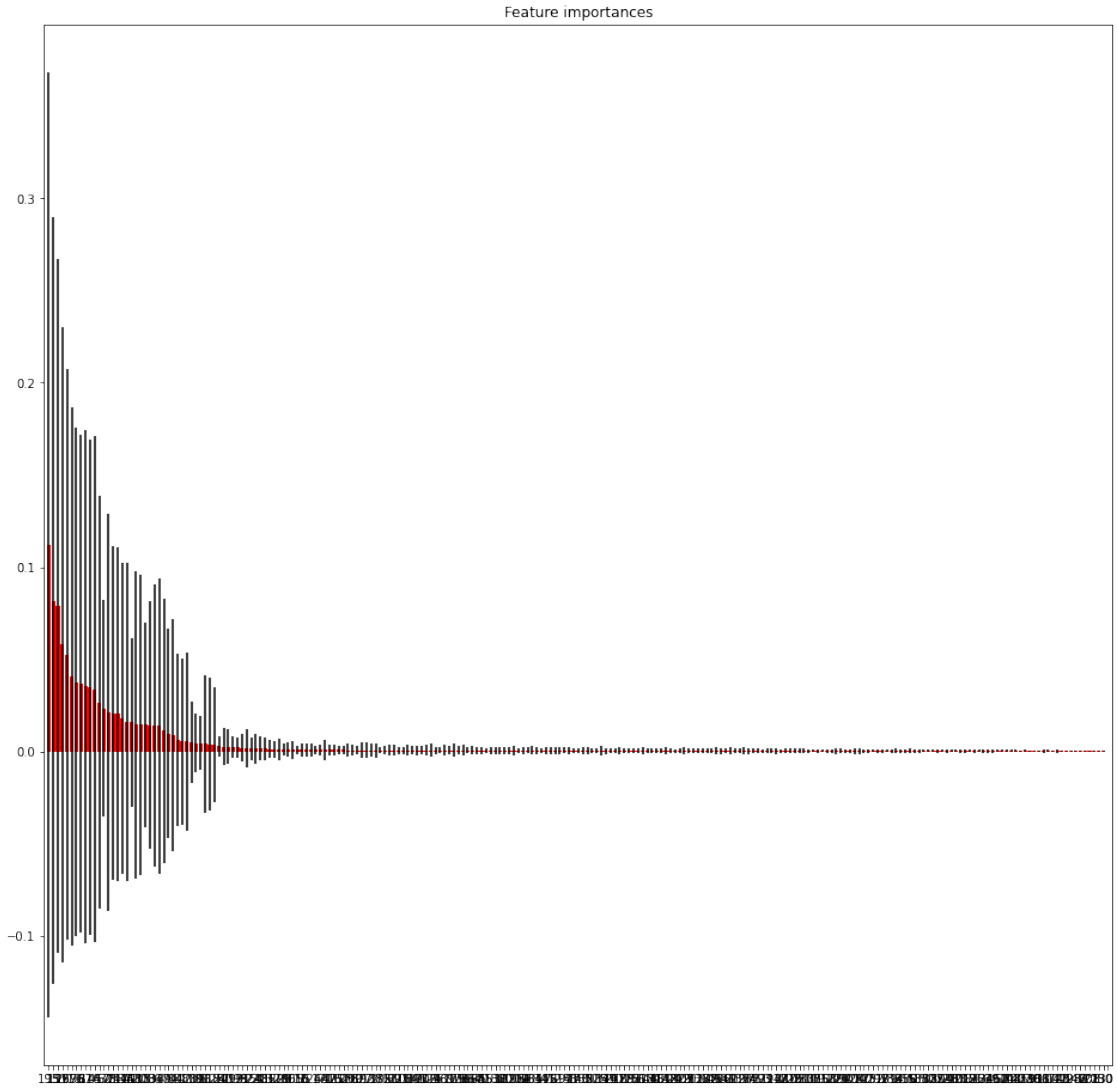
83. Feature: skew sfm (0.0005006152932043885)
84. Feature: meanfreq Q75 (0.0004916639863313636)
85. Feature: sd skew (0.0004761430223040 6024)
86. Feature: skew centroid (0.0004688961358933702)
87. Feature: sp.ent (0.0004686453947048572)
88. Feature: mode minfun (0.0004666298530929233)
89. Feature: mode meanfun (0.0004465616603583564)
90. Feature: sp.ent centroid (0.00044484341919639374)
91. Feature: sfm minfun (0.0004375418056340132)
92. Feature: sfm centroid (0.00043456880959191155)
93. Feature: median mode (0.0004336771973790728)
94. Feature: meanfreq modindx (0.0004333171616511944)
95. Feature: sd meandom (0.00042140462601995317)
96. Feature: sp.ent^2 (0.000420796429323829)
97. Feature: centroid minfun (0.0004036935553786201)
98. Feature: meanfreq sfm (0.00040322472000096856)
99. Feature: IQR meandom (0.00040261558590879917)
100. Feature: meanfun maxdom (0.00039533815899788416)
101. Feature: meanfun meandom (0.00039277667538018804)
102. Feature: meanfun modindx (0.0003840626664957682)
103. Feature: median skew (0.00037672573167766025)
104. Feature: Q25 dfrange (0.0003743325087170432)
105. Feature: sp.ent modindx (0.00037340215942623036)
106. Feature: mode dfrange (0.0003711061569050759)
107. Feature: sd centroid (0.0003688328532402381)
108. Feature: kurt meanfun (0.0003675916622469631)
109. Feature: median dfrange (0.0003644561657936423)
110. Feature: centroid (0.00036130593950533293)
111. Feature: maxfun dfrange (0.0003541785805430 5793)
112. Feature: Q75 skew (0.0003538191683771468)
113. Feature: sd minfun (0.00035375263116378474)
114. Feature: mindom (0.0003497068308842242)
115. Feature: kurt^2 (0.00034968039273762156)
116. Feature: mode modindx (0.000342846015164257)
117. Feature: meanfreq dfrange (0.0003423546815758409)
118. Feature: meanfreq mode (0.0003380831504165445)
119. Feature: maxdom^2 (0.00033637335451257473)
120. Feature: meanfreq minfun (0.0003334095110718464)
121. Feature: skew meandom (0.000321024830874526)
122. Feature: minfun (0.0003185869634348458)
123. Feature: meandom mindom (0.00031676355520988763)
124. Feature: Q75 mode (0.00029958267004461696)
125. Feature: centroid maxfun (0.00029791126947849197)
126. Feature: sfm maxdom (0.00029553850616686914)
127. Feature: Q25 mode (0.00029517970972332103)
128. Feature: sd mindom (0.00029171074367077403)
129. Feature: maxfun maxdom (0.00028847791961888287)
130. Feature: kurt centroid (0.00028237262870480 73)

131. Feature: mode meandom (0.00027845336212738907)
132. Feature: Q25 minfun (0.0002784148001449111)
133. Feature: sp.ent mindom (0.0002784021770959134)
134. Feature: kurt sfm (0.00027713722596778225)
135. Feature: mindom modindx (0.0002760083728443188)
136. Feature: mode mindom (0.0002748339912279751)
137. Feature: meandom dfrange (0.00027285475017134047)
138. Feature: Q75 dfrange (0.0002669013141628841)
139. Feature: skew (0.000256019866568791)
140. Feature: meanfun dfrange (0.00025597540208291907)
141. Feature: skew mindom (0.0002554155479047083)
142. Feature: meanfreq skew (0.000255050513421883362)
143. Feature: skew maxdom (0.0002531895685346137)
144. Feature: kurt minfun (0.0002519665036043748)
145. Feature: centroid meandom (0.0002506968525834974)
146. Feature: meandom modindx (0.0002496908751660209)
147. Feature: kurt mindom (0.0002481320627371673)
148. Feature: median Q75 (0.0002471016362234265)
149. Feature: minfun modindx (0.0002468737392787987)
150. Feature: skew minfun (0.0002454267934984441)
151. Feature: meanfreq kurt (0.00024139943127308734)
152. Feature: median mindom (0.0002394514985394146)
153. Feature: median maxdom (0.00023671066184433402)
154. Feature: sd modindx (0.0002338102343739289)
155. Feature: dfrange modindx (0.00023344525011445818)
156. Feature: median (0.00022350453456180218)
157. Feature: IQR kurt (0.00022312337974931217)
158. Feature: meandom maxdom (0.00021911565078818526)
159. Feature: sd kurt (0.00021657774412320034)
160. Feature: maxdom modindx (0.0002144056613306248)
161. Feature: mindom maxdom (0.00021344129031456786)
162. Feature: Q75 maxfun (0.00020570147680657854)
163. Feature: kurt dfrange (0.00020517500655333303)
164. Feature: mode maxfun (0.00020219153490655575)
165. Feature: maxfun meandom (0.00019995408764367958)
166. Feature: meanfreq mindom (0.0001973747884492374)
167. Feature: sp.ent mode (0.00019727160794169119)
168. Feature: centroid mindom (0.00019461330648521297)
169. Feature: sp.ent minfun (0.0001939963271364644)
170. Feature: mode centroid (0.00018915871055538778)
171. Feature: mindom dfrange (0.0001875223215368718)
172. Feature: centroid modindx (0.00018490329516510022)
173. Feature: maxfun mindom (0.00018444126761589345)
174. Feature: Q75 mindom (0.00018328107222439314)
175. Feature: minfun maxfun (0.0001789561786227486)
176. Feature: kurt (0.00017649295545836284)
177. Feature: Q75 centroid (0.00016803871923470 7)
178. Feature: median meandom (0.00016748986803289006)

179. Feature: kurt modindx (0.00016698051596990466)
180. Feature: centroid maxdom (0.0001668238726469773)
181. Feature: meandom (0.00016642858259917744)
182. Feature: sfm meandom (0.00016276831557022998)
183. Feature: mode maxdom (0.00016183005676522242)
184. Feature: Q25 modindx (0.00015906495896655527)
185. Feature: median kurt (0.0001557142503567967)
186. Feature: maxfun modindx (0.0001518432549483834)
187. Feature: meanfreq meandom (0.00014869385430137734)
188. Feature: sd dfrange (0.00014695910278023793)
189. Feature: skew dfrange (0.00014648061486825098)
190. Feature: kurt maxdom (0.00014226561274217125)
191. Feature: sd mode (0.00013738464047557944)
192. Feature: skew kurt (0.00013733199446382152)
193. Feature: meanfreq maxfun (0.0001355329487889972)
194. Feature: Q75 minfun (0.00013539314168440256)
195. Feature: dfrange^2 (0.00013155827162873357)
196. Feature: maxfun^2 (0.00012865046253514884)
197. Feature: sp.ent dfrange (0.00012598877912742723)
198. Feature: Q75 meandom (0.00012458724961463105)
199. Feature: minfun meandom (0.0001244271019205891)
200. Feature: dfrange (0.00012144916245475881)
201. Feature: skew^2 (0.00012126583356648472)
202. Feature: meandom^2 (0.00012056996764960434)
203. Feature: skew maxfun (0.00011586545048518696)
204. Feature: maxdom dfrange (0.00011195625950159703)
205. Feature: sd sfm (0.00010884397721227349)
206. Feature: sp.ent meandom (0.00010861351394526194)
207. Feature: sd maxdom (0.00010651644244324978)
208. Feature: modindx (0.00010608226314518603)
209. Feature: kurt meandom (0.0001045563881587145)
210. Feature: kurt mode (9.868359042338895e-05)
211. Feature: minfun maxdom (9.370154080630282e-05)
212. Feature: sp.ent maxfun (9.297065069639522e-05)
213. Feature: modindx^2 (9.04030724050802e-05)
214. Feature: Q75 kurt (8.918159792291425e-05)
215. Feature: sfm mode (8.18165541853939e-05)
216. Feature: maxdom (8.117624195121266e-05)
217. Feature: Q75 modindx (7.817465740575024e-05)
218. Feature: sfm dfrange (7.747921911359829e-05)
219. Feature: kurt sp.ent (7.351421092370216e-05)
220. Feature: meanfreq median (6.662273415719219e-05)
221. Feature: meanfreq sp.ent (6.11763846925552e-05)
222. Feature: skew sp.ent (5.857687682874431e-05)
223. Feature: kurt maxfun (5.777556488539093e-05)
224. Feature: meanfreq centroid (5.066399943370645e-05)
225. Feature: median^2 (4.5379721593206554e-05)
226. Feature: minfun dfrange (3.892552857881645e-05)

```
227. Feature: minfun^2 (3.846892061356815e-05)
228. Feature: maxfun (3.2113125698659905e-05)
229. Feature: skew mode (2.693033561221176e-05)
230. Feature: 1 (0.0)
Features in order of decreasing importance:
['meanfun^2', 'meanfun', 'sp.ent meanfun', 'meanfreq IQR', 'meanfun maxfun',
'IQR^2', 'Q75 IQR', 'IQR', 'Q25 meanfun', 'IQR sp.ent', 'sd IQR', 'median IQR',
'sd meanfun', 'Q25^2', 'meanfreq Q25', 'IQR centroid', 'centroid meanfun', 'sd
Q75', 'sfm meanfun', 'IQR maxfun', 'IQR sfm', 'Q75 meanfun', 'meanfreq meanfun',
'Q25 centroid', 'Q25', 'Q25 maxfun', 'median meanfun', 'Q25 sp.ent', 'sd^2', 'sd
sp.ent', 'sd', 'IQR meanfun', 'Q25 IQR', 'meanfun minfun', 'median Q25', 'IQR
skew', 'IQR modindx', 'Q25 sfm', 'Q25 skew', 'Q25 Q75', 'IQR minfun', 'meanfun
mindom', 'meanfreq^2', 'Q25 kurt', 'meanfreq sd', 'sd maxfun', 'sd Q25',
'centroid^2', 'skew meanfun', 'IQR mindom', 'sfm maxfun', 'Q75 sp.ent', 'Q25
meandom', 'sfm^2', 'IQR mode', 'sfm', 'Q75', 'IQR dfrange', 'meanfreq', 'median
sp.ent', 'median minfun', 'sd median', 'sfm modindx', 'sp.ent maxdom', 'IQR
maxdom', 'mode^2', 'Q75 sfm', 'median sfm', 'Q25 mindom', 'Q25 maxdom', 'median
maxfun', 'meanfreq maxdom', 'sfm mindom', 'Q75^2', 'mode', 'median centroid',
'mindom^2', 'Q75 maxdom', 'sp.ent sfm', 'skew modindx', 'centroid dfrange',
'median modindx', 'minfun mindom', 'skew sfm', 'meanfreq Q75', 'sd skew', 'skew
centroid', 'sp.ent', 'mode minfun', 'mode meanfun', 'sp.ent centroid', 'sfm
minfun', 'sfm centroid', 'median mode', 'meanfreq modindx', 'sd meandom',
'sp.ent^2', 'centroid minfun', 'meanfreq sfm', 'IQR meandom', 'meanfun maxdom',
'meanfun meandom', 'meanfun modindx', 'median skew', 'Q25 dfrange', 'sp.ent
modindx', 'mode dfrange', 'sd centroid', 'kurt meanfun', 'median dfrange',
'centroid', 'maxfun dfrange', 'Q75 skew', 'sd minfun', 'mindom', 'kurt^2', 'mode
modindx', 'meanfreq dfrange', 'meanfreq mode', 'maxdom^2', 'meanfreq minfun',
'skew meandom', 'minfun', 'meandom mindom', 'Q75 mode', 'centroid maxfun', 'sfm
maxdom', 'Q25 mode', 'sd mindom', 'maxfun maxdom', 'kurt centroid', 'mode
meandom', 'Q25 minfun', 'sp.ent mindom', 'kurt sfm', 'mindom modindx', 'mode
mindom', 'meandom dfrange', 'Q75 dfrange', 'skew', 'meanfun dfrange', 'skew
mindom', 'meanfreq skew', 'skew maxdom', 'kurt minfun', 'centroid meandom',
'meandom modindx', 'kurt mindom', 'median Q75', 'minfun modindx', 'skew minfun',
'meanfreq kurt', 'median mindom', 'median maxdom', 'sd modindx', 'dfrange
modindx', 'median', 'IQR kurt', 'meandom maxdom', 'sd kurt', 'maxdom modindx',
'mindom maxdom', 'Q75 maxfun', 'kurt dfrange', 'mode maxfun', 'maxfun meandom',
'meanfreq mindom', 'sp.ent mode', 'centroid mindom', 'sp.ent minfun', 'mode
centroid', 'mindom dfrange', 'centroid modindx', 'maxfun mindom', 'Q75 mindom',
'minfun maxfun', 'kurt', 'Q75 centroid', 'median meandom', 'kurt modindx',
'centroid maxdom', 'meandom', 'sfm meandom', 'mode maxdom', 'Q25 modindx',
'median kurt', 'maxfun modindx', 'meanfreq meandom', 'sd dfrange', 'skew
dfrange', 'kurt maxdom', 'sd mode', 'skew kurt', 'meanfreq maxfun', 'Q75
minfun', 'dfrange^2', 'maxfun^2', 'sp.ent dfrange', 'Q75 meandom', 'minfun
meandom', 'dfrange', 'skew^2', 'meandom^2', 'skew maxfun', 'maxdom dfrange', 'sd
sfm', 'sp.ent meandom', 'sd maxdom', 'modindx', 'kurt meandom', 'kurt mode',
'minfun maxdom', 'sp.ent maxfun', 'modindx^2', 'Q75 kurt', 'sfm mode', 'maxdom',
'Q75 modindx', 'sfm dfrange', 'kurt sp.ent', 'meanfreq median', 'meanfreq
sp.ent', 'skew sp.ent', 'kurt maxfun', 'meanfreq centroid', 'median^2', 'minfun
```

```
dfrange', 'minfun^2', 'maxfun', 'skew mode', '1']
```



Tutaj sprawdzaliśmy jakie rezultaty uzyskamy po odrzuceniu konkretnych zestawów kolumn. Zestawy tworzyliśmy na podstawie wcześniejszej analizy ważności cech. Finalny zestaw kolumn jeszcze do sprecyzowania.

[12]:
```
drops = ['meanfreq', 'mindom', 'median', 'maxdom', 'skew', 'Q75', 'kurt',␣
 ↪'minfun', 'dfrange', 'meandom', 'maxfun', 'modindx']
#Columns in importance order
#drops = ['meanfun', 'IQR', 'Q25', 'sd', 'sp.ent', 'sfm', 'centroid', 'mode',␣
 ↪'meanfreq', 'mindom', 'median', 'maxdom', 'skew', 'Q75', 'kurt', 'minfun',␣
 ↪'dfrange', 'meandom', 'maxfun', 'modindx']
```

```
#['meanfun^2', 'meanfun', 'sp.ent meanfun', 'meanfreq IQR', 'meanfun maxfun',␣
 ↪'IQR^2', 'Q75 IQR', 'IQR', 'Q25 meanfun', 'IQR sp.ent', 'sd IQR', 'median␣
 ↪IQR', 'sd meanfun', 'Q25^2', 'meanfreq Q25', 'IQR centroid', 'centroid␣
 ↪meanfun', 'sd Q75', 'sfm meanfun', 'IQR maxfun', 'IQR sfm', 'Q75 meanfun',␣
 ↪'meanfreq meanfun', 'Q25 centroid', 'Q25', 'Q25 maxfun', 'median meanfun',␣
 ↪'Q25 sp.ent', 'sd^2', 'sd sp.ent', 'sd', 'IQR meanfun', 'Q25 IQR', 'meanfun␣
 ↪minfun', 'median Q25', 'IQR skew', 'IQR modindx', 'Q25 sfm', 'Q25 skew',␣
 ↪'Q25 Q75', 'IQR minfun', 'meanfun mindom', 'meanfreq^2', 'Q25 kurt',␣
 ↪'meanfreq sd', 'sd maxfun', 'sd Q25', 'centroid^2', 'skew meanfun', 'IQR␣
 ↪mindom', 'sfm maxfun', 'Q75 sp.ent', 'Q25 meandom', 'sfm^2', 'IQR mode',␣
 ↪'sfm', 'Q75', 'IQR dfrange', 'meanfreq', 'median sp.ent', 'median minfun',␣
 ↪'sd median', 'sfm modindx', 'sp.ent maxdom', 'IQR maxdom', 'mode^2', 'Q75␣
 ↪sfm', 'median sfm', 'Q25 mindom', 'Q25 maxdom', 'median maxfun', 'meanfreq␣
 ↪maxdom', 'sfm mindom', 'Q75^2', 'mode', 'median centroid', 'mindom^2', 'Q75␣
 ↪maxdom', 'sp.ent sfm', 'skew modindx', 'centroid dfrange', 'median modindx',␣
 ↪'minfun mindom', 'skew sfm', 'meanfreq Q75', 'sd skew', 'skew centroid', 'sp.
 ↪ent', 'mode minfun', 'mode meanfun', 'sp.ent centroid', 'sfm minfun', 'sfm␣
 ↪centroid', 'median mode', 'meanfreq modindx', 'sd meandom', 'sp.ent^2',␣
 ↪'centroid minfun', 'meanfreq sfm', 'IQR meandom', 'meanfun maxdom', 'meanfun␣
 ↪meandom', 'meanfun modindx', 'median skew', 'Q25 dfrange', 'sp.ent modindx',␣
 ↪'mode dfrange', 'sd centroid', 'kurt meanfun', 'median dfrange', 'centroid',␣
 ↪'maxfun dfrange', 'Q75 skew', 'sd minfun', 'mindom', 'kurt^2', 'mode␣
 ↪modindx', 'meanfreq dfrange', 'meanfreq mode', 'maxdom^2', 'meanfreq␣
 ↪minfun', 'skew meandom', 'minfun', 'meandom mindom', 'Q75 mode', 'centroid␣
 ↪maxfun', 'sfm maxdom', 'Q25 mode', 'sd mindom', 'maxfun maxdom', 'kurt␣
 ↪centroid', 'mode meandom', 'Q25 minfun', 'sp.ent mindom', 'kurt sfm',␣
 ↪'mindom modindx', 'mode mindom', 'meandom dfrange', 'Q75 dfrange', 'skew',␣
 ↪'meanfun dfrange', 'skew mindom', 'meanfreq skew', 'skew maxdom', 'kurt␣
 ↪minfun', 'centroid meandom', 'meandom modindx', 'kurt mindom', 'median Q75',␣
 ↪'minfun modindx', 'skew minfun', 'meanfreq kurt', 'median mindom', 'median␣
 ↪maxdom', 'sd modindx', 'dfrange modindx', 'median', 'IQR kurt', 'meandom␣
 ↪maxdom', 'sd kurt', 'maxdom modindx', 'mindom maxdom', 'Q75 maxfun', 'kurt␣
 ↪dfrange', 'mode maxfun', 'maxfun meandom', 'meanfreq mindom', 'sp.ent mode',␣
 ↪'centroid mindom', 'sp.ent minfun', 'mode centroid', 'mindom dfrange',␣
 ↪'centroid modindx', 'maxfun mindom', 'Q75 mindom', 'minfun maxfun', 'kurt',␣
 ↪'Q75 centroid', 'median meandom', 'kurt modindx', 'centroid maxdom',␣
 ↪'meandom', 'sfm meandom', 'mode maxdom', 'Q25 modindx', 'median kurt',␣
 ↪'maxfun modindx', 'meanfreq meandom', 'sd dfrange', 'skew dfrange', 'kurt␣
 ↪maxdom', 'sd mode', 'skew kurt', 'meanfreq maxfun', 'Q75 minfun',␣
 ↪'dfrange^2', 'maxfun^2', 'sp.ent dfrange', 'Q75 meandom', 'minfun meandom',␣
 ↪'dfrange', 'skew^2', 'meandom^2', 'skew maxfun', 'maxdom dfrange', 'sd sfm',␣
 ↪'sp.ent meandom', 'sd maxdom', 'modindx', 'kurt meandom', 'kurt mode',␣
 ↪'minfun maxdom', 'sp.ent maxfun', 'modindx^2', 'Q75 kurt', 'sfm mode',␣
 ↪'maxdom', 'Q75 modindx', 'sfm dfrange', 'kurt sp.ent', 'meanfreq median',␣
 ↪'meanfreq sp.ent', 'skew sp.ent', 'kurt maxfun', 'meanfreq centroid',␣
 ↪'median^2', 'minfun dfrange', 'minfun^2', 'maxfun', 'skew mode', '1']
```

```
#X = X2[['sp.ent', 'meanfreq IQR', 'meanfun maxfun', 'IQR', 'sd',]]
#X = X2[['meanfun^2', 'sp.ent', 'meanfreq IQR', 'meanfun maxfun', 'IQR sp.ent',␣
↪'sd']]
X = X.drop(drops, axis=1)
X.columns
```

[12]:
```
Index(['sd', 'Q25', 'IQR', 'sp.ent', 'sfm', 'mode', 'centroid', 'meanfun'],
dtype='object')
```

Tutaj testowaliśmy lasy z innymi hiperparametrami.

[13]:
```
X_train, X_val, X_test, y_train, y_val, y_test = split(X, y)
```

[14]:
```
forest2 = RandomForestClassifier(n_estimators=100, random_state=10, max_depth=6)
forest2.fit(X_train, y_train)
print(forest2.score(X_val, y_val))
```

```
0.9808743169398907
```

[15]:
```
X_train, X_val, X_test, y_train, y_val, y_test = split(X, y)
```

[16]:
```
forest = RandomForestClassifier(n_estimators=5, random_state=10)
forest.fit(X_train, y_train)
print("Validation score: {:.6f}".format(forest.score(X_val, y_val)))
print("Test score: {:.6f}".format(forest.score(X_test,y_test)))
```

```
Validation score: 0.979508
Test score: 0.987261
```

Teraz sprawdzaliśmy jaki wynik mają podstawowe modele w różnych metrykach.

[17]:
```
lr = LogisticRegression(max_iter=1000)
tree = DecisionTreeClassifier()
SVM = SVC(kernel='linear', C=1.0)
nb = GaussianNB()
forest1 = RandomForestClassifier(n_estimators=100, random_state=10, max_depth=6)
forest2 = RandomForestClassifier(n_estimators=5, random_state=10)
estimators = [('Logistic Regression', lr), ('Decision Tree', tree), ('SVM',␣
↪SVM), ('Naive Bayes', nb), ('Random Forest 1', forest1), ('Random Forest 2',␣
↪forest2)]
scoring = ['accuracy', 'precision', 'recall', 'f1', 'roc_auc']
matrix = []
for name, model in estimators:
    row = [name]
    for score in scoring:
        result = cross_val_score(model, X, y, cv=5, scoring=score).mean()
        row.append(result)
    matrix.append(row)
```

```
result = pd.DataFrame(matrix, columns=["Model", "Accuracy", "Precision",␣
↪"Recall", "F1", "Roc-Auc"])
```

[18]:
```
result
```

[18]:

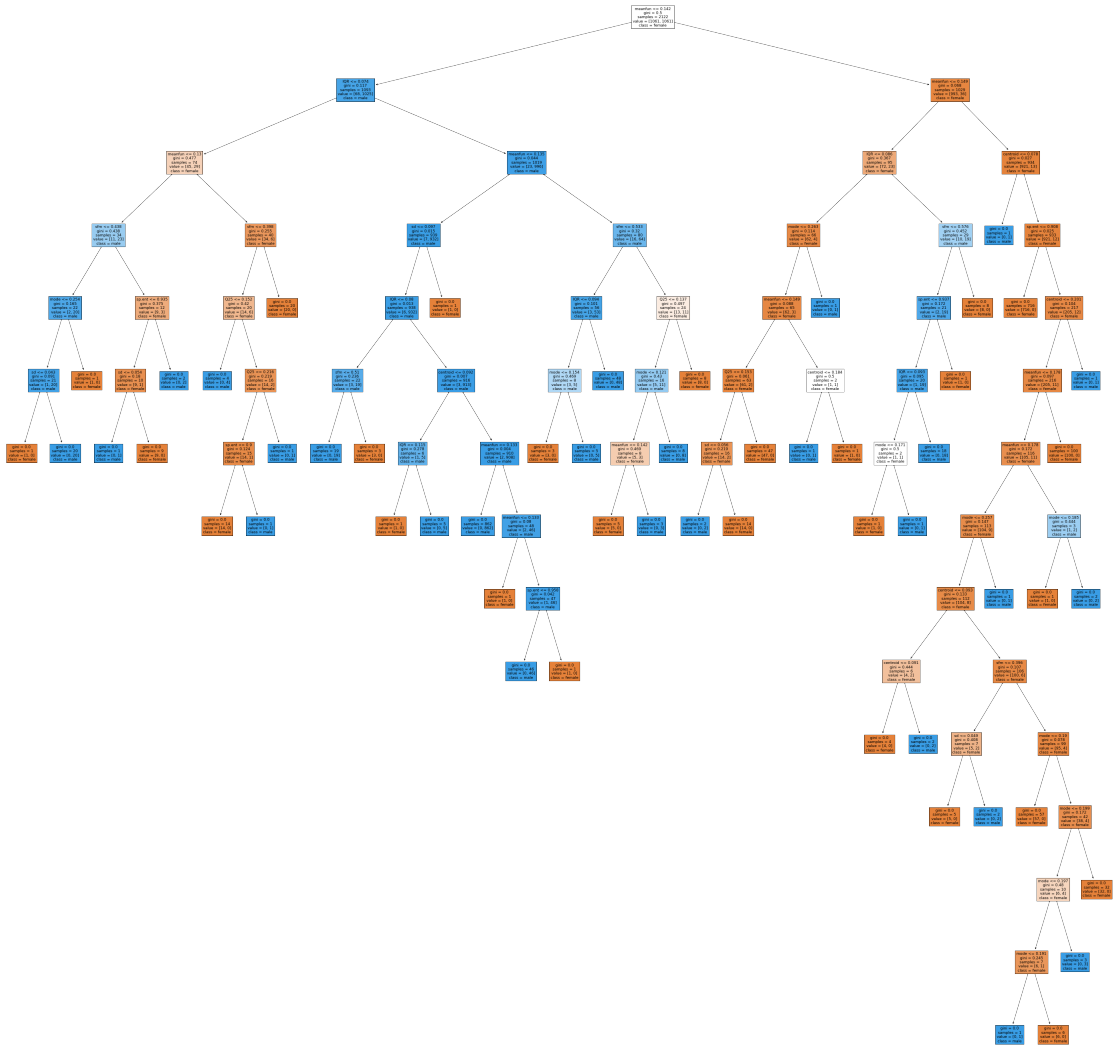
	Model	Accuracy	Precision	Recall	F1	Roc-Auc
0	Logistic Regression	0.903435	0.862941	0.971601	0.912226	0.941155
1	Decision Tree	0.956760	0.954945	0.953927	0.952598	0.955503
2	SVM	0.916055	0.875755	0.978549	0.922897	0.970967
3	Naive Bayes	0.929612	0.909160	0.960236	0.932718	0.979323
4	Random Forest 1	0.965279	0.963386	0.967813	0.965344	0.993128
5	Random Forest 2	0.963700	0.963939	0.964028	0.963635	0.985301

W powyższej tabelce zauważamy, że wyniki są bardzo obiecujące. Widać, że najlepsze rezultaty osiąga RandomForest, jednak musi być jeszcze przeprowadzona analiza dla różnych hiperparametrów przed wyborem finalnego modelu.

Poniżej wygenerowaliśmy rysunek reprezentujący jakie cechy bierze pod uwagę losowe drzewo decyzyjne.

[19]:
```
tree2 = DecisionTreeClassifier()
tree2.fit(X_train, y_train)
print(f"Accuracy score: {tree2.score(X_test, y_test)}")
plt.figure(figsize=(50,50))
splits=plot_tree(tree2, feature_names=X.columns, class_names=['female',␣
↪'male'], filled=True)
```

```
Accuracy score: 0.9713375796178344
```



[20]:
```
pip freeze > requirements.txt
```

```
Note: you may need to restart the kernel to use updated packages.
```