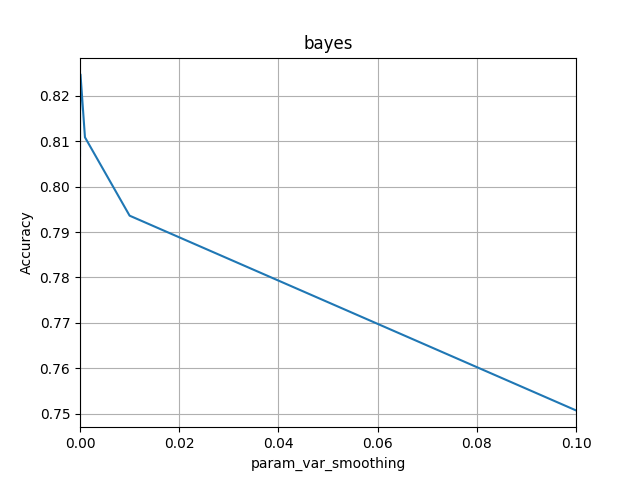
Projekt: rozpoznawanie liści – Etap II

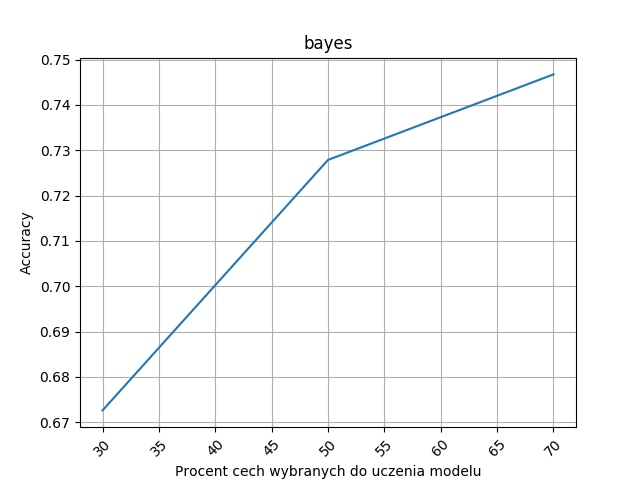
Katarzyna Markowska, Jędrzej Kuczyński

Eksperymenty obliczeniowe



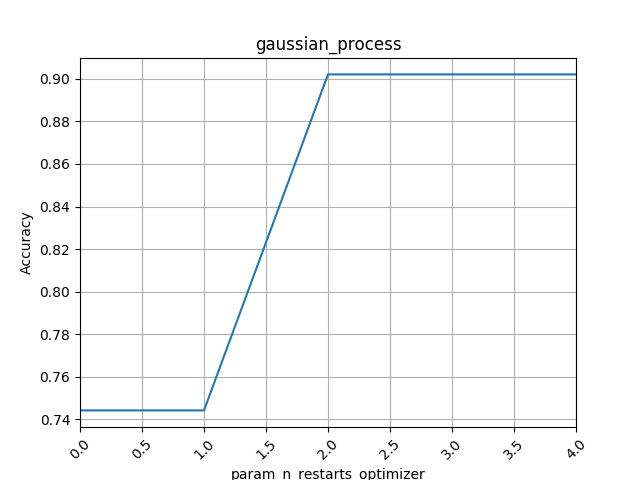
## Gaussian Naive Bayes

Pierwszy algorytm jaki testowaliśmy wykazuje spadającą skuteczność wraz ze wzrostem parametru var\_smoothing



Jeśli natomiast chodzi o jego efektywność w zależności od ilości cech, które były brane pod uwagę podczas uczenia, to można zauważyć, że im było ich więcej, tym przewidywania algorytmu były na wyższym poziomie.

## Gaussian Process Classification

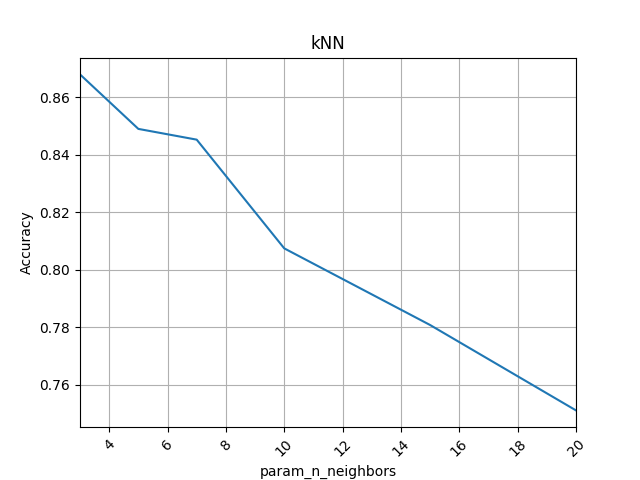


W przypadku kolejnego algorytmu parametr n\_restarts\_optimizer ma bardzo duże znaczenie na efektywność modelu. Im wartość tego argumentu jest większa tym lepsze są przewidywania, jednak warto zauważyć, że przy przejściu z wartości 1 do 2 następuje gwałtowny wzrost jakości tych przewidywań.

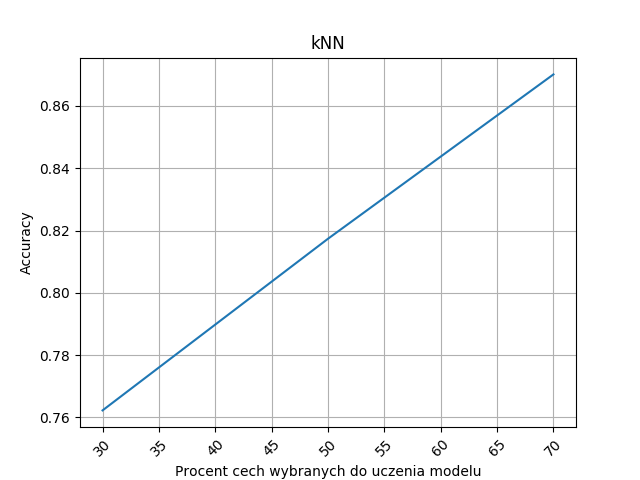


Tym razem również im więcej cech było branych pod uwagę, tym model przewidywał gatunki z większą trafnością.

## Nearest Neighbors Classification

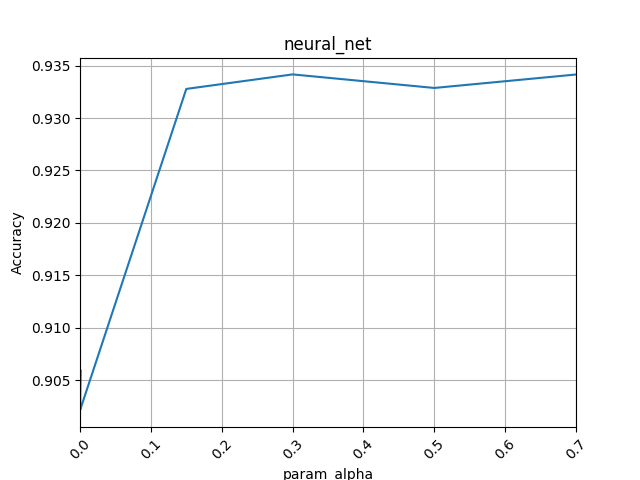


Trzeci testowany przez nas algorytm cechował się coraz gorszą skutecznością wraz ze wzrostem liczby sąsiadów biorących udział w klasyfikacji.



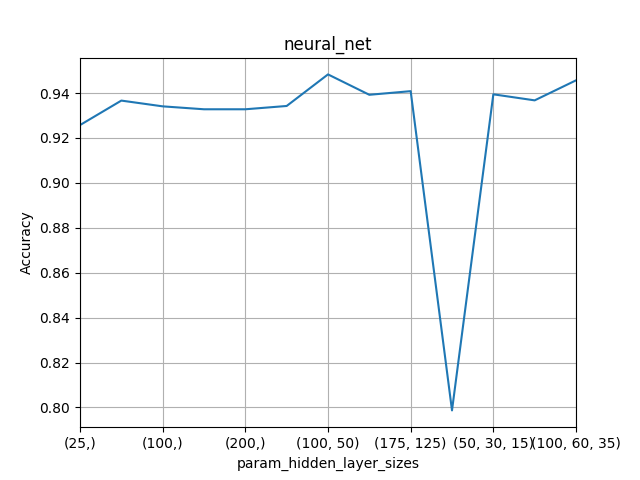
(wykres był rysowany dla parametru *weights =* uniform)

Jeśli zaś spojrzeć na zależność skuteczności od ilości cech to wygląda ona dokładnie tak samo jak w przypadku poprzednich algorytmów, czyli im więcej cech, tym efektywność większa.



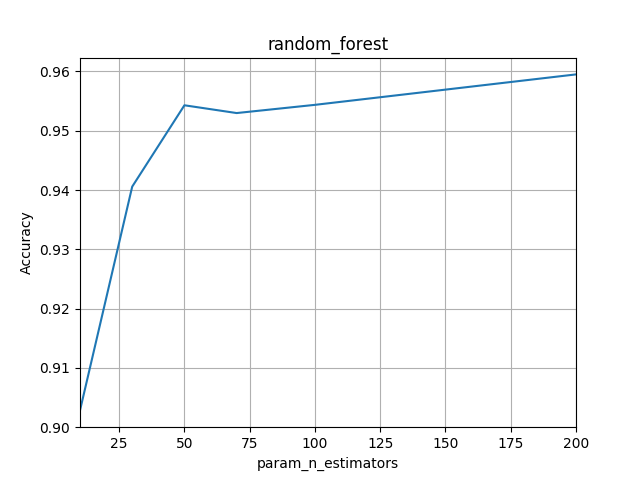
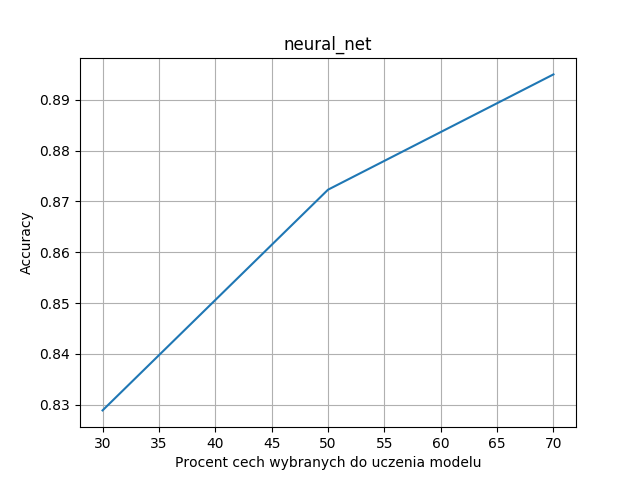
## Neural network models

Kolejny rozpatrywany algorytm osiąga w ogólności lepsze wyniki wraz ze wzrostem wartości parametru alpha. Skokowy wzrost skuteczności następuje przy przejściu wartości od 0 do 0,15. (*hidden\_layer\_sizes* = (150,), *activation* = tanh, *max\_iter* = 300)



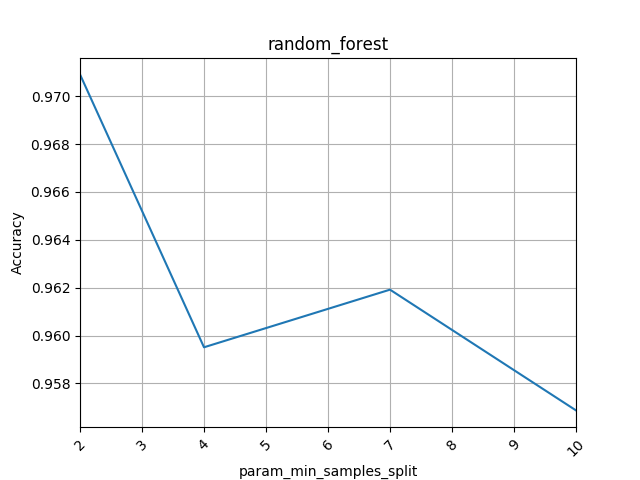
W przypadku parametru hidden\_layer\_sizes najgorsza jego wartość to (25, 15, 5) – skuteczność spada do około 80%, natomiast najlepsza to (100, 50) – około 94,5% skuteczności. (*alpha* = 0.15, *activation* = tanh, *max\_iter* = 300)

Rozpatrując zależność skuteczności sieci neuronowych w zależności od ilości cech, które brały udział w uczeniu można zauważyć tą samą prawidłowość co przy poprzednich eksperymentach – im więcej cech, tym lepsze wyniki.



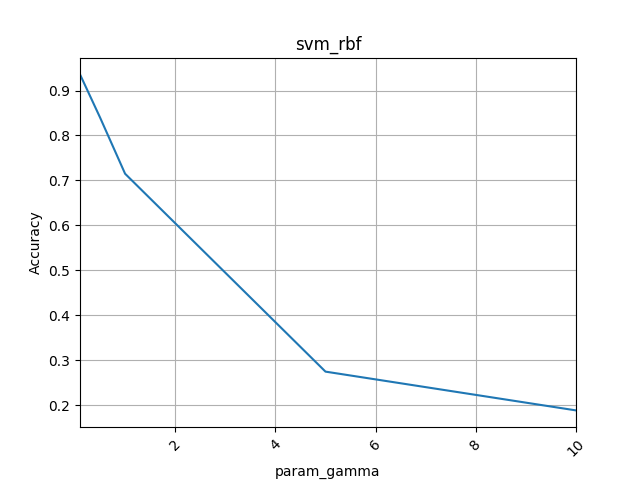
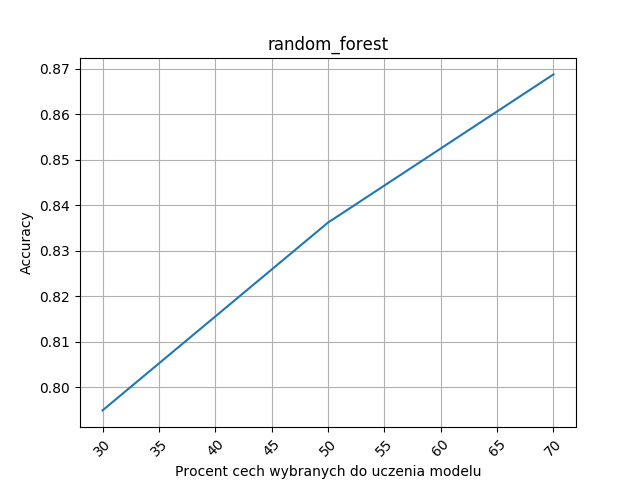
## Random forest

Im wyższa wartość parametru n\_estimators tym lepsza skuteczność algorytmu. Skokowy jej wzrost następuje w przejściu wartości parametru od 0 do 50. (*max\_depth* = 15, *min\_samples\_split* = 4)



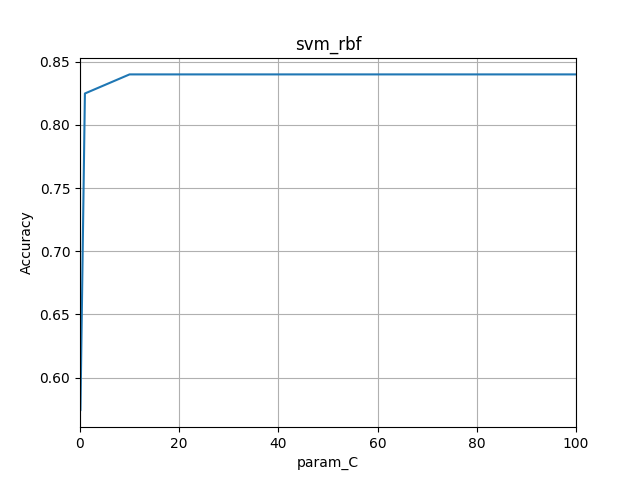
W przypadku parametru min\_sample\_split można powiedzieć że im mniejsza jego wartość, tym lepsze są przewidywania algorytmu. Następuje jednak nieznaczny wzrost skuteczności w przypadku wartości 7. (*max\_depth* = 15, *n\_estimators* = 200)

Zależność skuteczności przewidywań w zależności od ilości użytych cech przyjmuje dokładnie taki sam kształt jak w przypadku poprzednich algorytmów – im więcej cech, tym lepsza skuteczność.



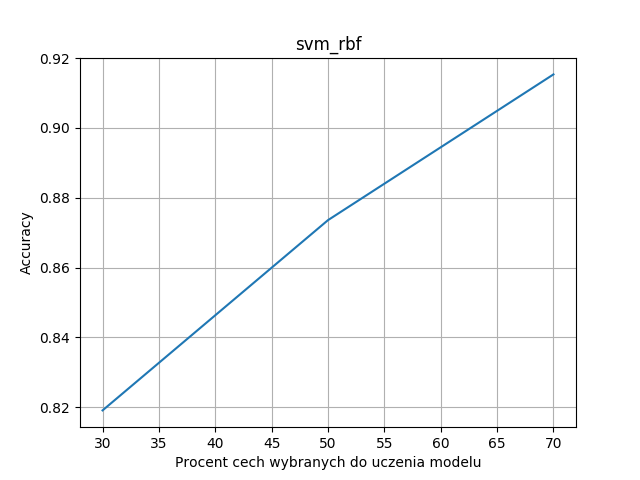
## Support Vector Machines

Ostatni badany przez nas algorytm osiąga tym lepsze wartości, im niższa wartość parametru gamma. (*C* = 100, *max\_iter* = 300)



Parametr C nie wpływa za bardzo na efektywność przewidywań jeśli jest powyżej wartości 10. Poniżej tego progu wraz ze spadkiem wartości parametru C można zaobserwować gwałtowny spadek trafności przewidywań. (*gamma* = 0.5, *max\_iter* = 300)

Jeśli natomiast chodzi o zależność skuteczności tego algorytmu w zależności od ilości użytych cech to kolejny raz powtarza się zależność, że im więcej cech, tym lepsze przewidywania.



# Najlepszy klasyfikator

Najlepszym klasyfikatorem okazał się algorytm random forest z poniższymi parametrami:

max\_depth: 15,

min\_samples\_split: 2,

n\_estimators: 200.

Jego skuteczność wynosiła 97%.