Raport Badawczy

Analiza i klasyfikacja jakości bananów na datasecie numeryczno-kategorialnym

Daniel Baca

Spis treści

1. Wstęp

W niniejszym raporcie zaprezentowane jest badanie zestawu danych „Banana Quality” (https://www.kaggle.com/datasets/l3llff/banana/data), którego celem jest wypracowanie automatycznego modelu klasyfikującego banany na „dobre” i „niedobre” na podstawie ich cech fizykochemicznych. Zgromadzony zbiór liczy około 8 000 obserwacji, a każda próbka opisana jest siedmioma zmiennymi numerycznymi:

- Size – wymiar owocu,
- Weight – masa owocu,
- Sweetness – stopień słodkości,
- Softness – miękkość miąższu,
- HarvestTime – czas (w dniach) od momentu zbioru,
- Ripeness – dojrzałość owocu,
- Acidity – poziom kwasowości,
- oraz etykietą Quality, wskazującą subiektywną ocenę jakości (Good/Bad).

Dane zostały zebrane w celu wspomagania procesów sortowania i kontroli jakości w przemyśle spożywczym – manualna ocena tysięcy sztuk owoców jest czasochłonna i podatna na błąd. Analiza statystyczna oraz budowa modeli klasyfikacyjnych pozwalają sprawdzić, które cechy najlepiej korelują z percepcją jakości, a także ocenić, czy można zastąpić lub wesprzeć ludzkiego inspektora algorytmem uczącym się.

2. Eksploracja danych (EDA)

2.1. Statystyki opisowe

Tabela 1. Podsumowanie statysyk opisowych (dane zaokrąglone)

Cecha	Ilość	Średnia	Odchylenie standardowe	Minimum	I kwartyl (25%)	Mediana (50%)	III kwartyl (75%)	Maksimum	Skośność	Kurtoza
Size	8000	-0.7478	2.1360	7.9981	-2.2777	-0.8975	0.6542	7.9708	0.2667	-0.1608
Weight	8000	-0.7610	2.0159	-8.2830	-2.2236	-0.8687	0.7755	5.6797	0.0445	-0.5266
Sweetness	8000	-0.7702	1.9485	-6.4340	-2.1073	-1.0207	0.3111	7.5394	0.6143	0.3909
Softness	8000	-0.0144	2.0652	-6.9593	-1.5905	0.2026	1.5471	8.2416	-0.1930	-0.3920
HarvestTime	8000	-0.7513	1.9967	-7.5700	-2.1207	-0.9342	0.5073	6.2933	0.2783	-0.0806
Ripeness	8000	0.7811	2.1143	-7.4232	-0.5742	0.9650	2.2617	7.2490	-0.3122	-0.0877
Acidity	8000	0.0087	2.2935	-8.2270	-1.6295	0.0987	1.6821	7.4116	-0.1530	-0.3058

Interpretacja wartości z tabeli statystyk opisowych

1. **Rozmiar próby i skalowanie**

– Wszystkie cechy obejmują 8 000 pomiarów, co stanowi solidną bazę do analizy statystycznej.
– Oryginalne wartości w datasecie oscylują w granicach od ~-8 do ~+8, a średnie podane w statystykach opisowych wynosiły ≃0, odchylenie ≃2 (czyli pierwotne σ zostało zgeneralizowane do 1). Każda kolumna numeryczna została zamieniona na $z = (x-\mu)/\sigma$.

2. **Średnie**

– Średnie (mean) każdej kolumny oscylują w granicach [-0.77, 0.78], niemal dokładnie wokół zera, co jest efektem odjęcia wartości μ przed podzieleniem przez σ.
– Najwyższa średnia występuje w Ripeness (0.78), co sugeruje, że owoce są w zbiorze nieco powyżej neutralnego poziomu dojrzałości.

3. **Odchylenia standardowe**

– Odchylenia standardowe (std) mieszczą się w przedziale [1.95, 2.29], co - choć nie idealnie 1 - wskazuje, że różne cechy nadal zachowują względnie porównywalną zmienność po normalizacji.

4. **Zakresy wartości**

– Zakresy wartości rozciągają się od około -8 do +8 we wszystkich kolumnach, co ujawnia obecność punktów leżących daleko od średniej (potencjalne outliery).

5. **Symetria i spłaszczenie**

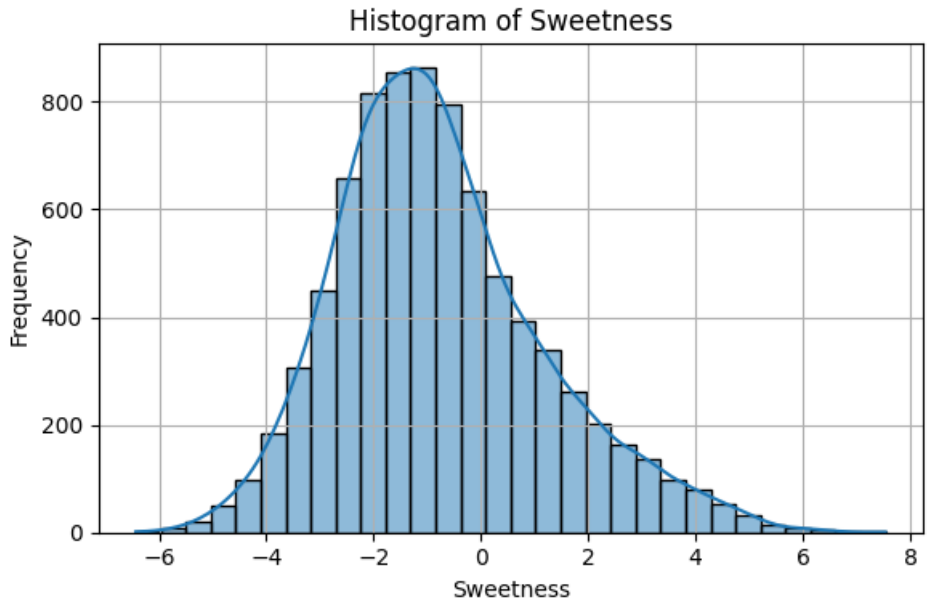
Symetria (skew) i spłaszczenie (kurtosis) rozkładów pokazują:

– Pozytywną prawoskośność w „Sweetness” (skew ≃ 0,61),
– Niewielkie lewoskośne odchylenie w „Ripeness” (skew ≃ -0,31),
– Kurtosis blisko zera dla większości cech - rozkłady nie odbiegają drastycznie od normalnych.

2.2. Histogramy

Wybrane histogramy wraz z opisem:

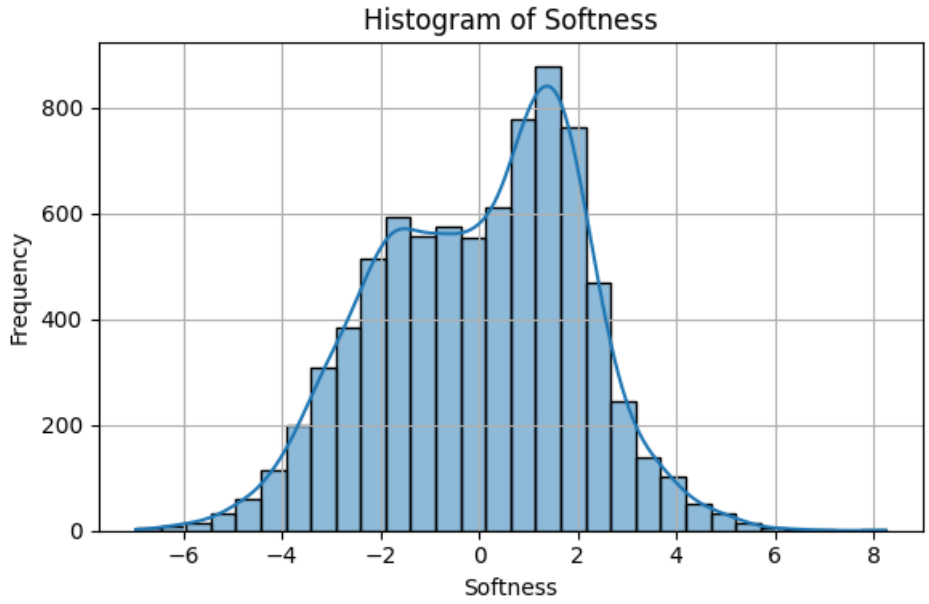
1. Sweetness



Rysunek 1. Histogram Sweetness

- Wyraźna prawoskośność (długi prawy ogon) – wysoka liczba niskich wartości, kilka bardzo słodkich bananów.
- Wskazuje potrzebę ewentualnej transformacji (np. log) lub zastosowania robust scalera.

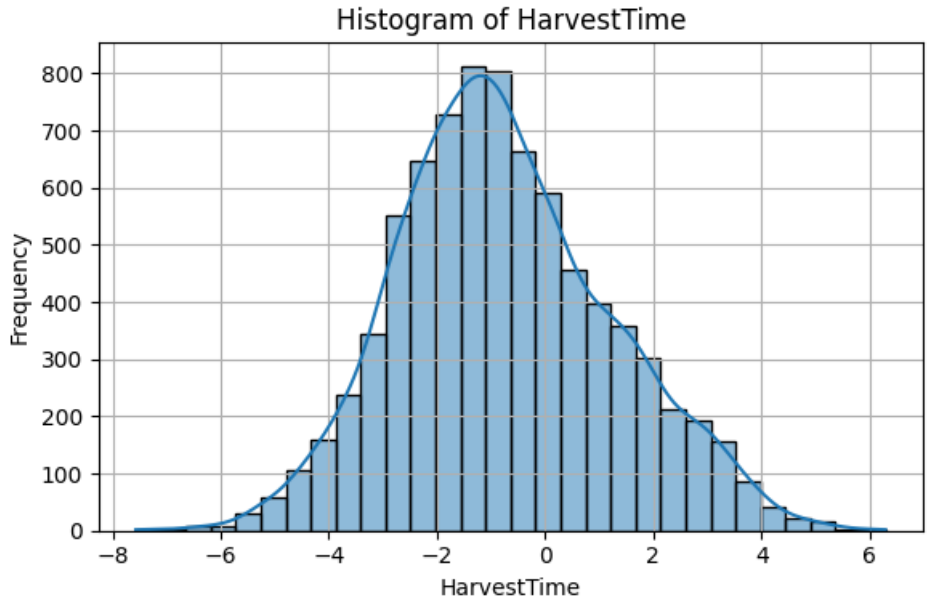
2. Softness



Rysunek 2. Histogram Softness

- Widoczne dwa „szczyty” rozkładu (bimodalność), sugerujące dwie podgrupy owoców (miękkie vs twardsze).
- Przydatne do feature engineering (np. kategoryzacja na „niska/średnia/wysoka miękkość”).

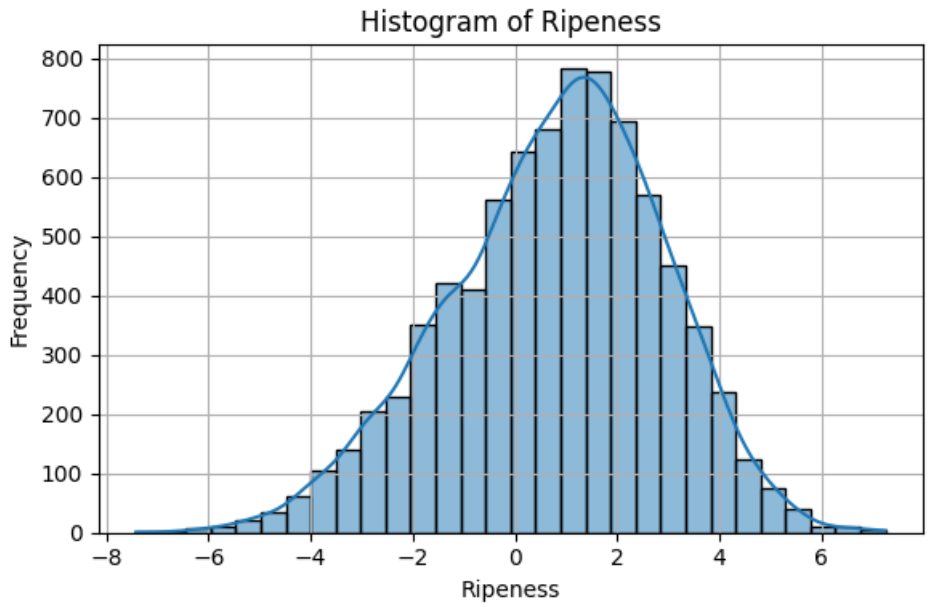
3. HarvestTime



Rysunek 3. Histogram HarvestTime

– Lekka prawoskośność, ale węższy ogon niż w „Sweetness” – większość bananów sklasyfikowana jako świeższe.
– Zasygnalizuje sensowny podział na przedziały czasowe (kwantyle) podczas dyskretyzacji.

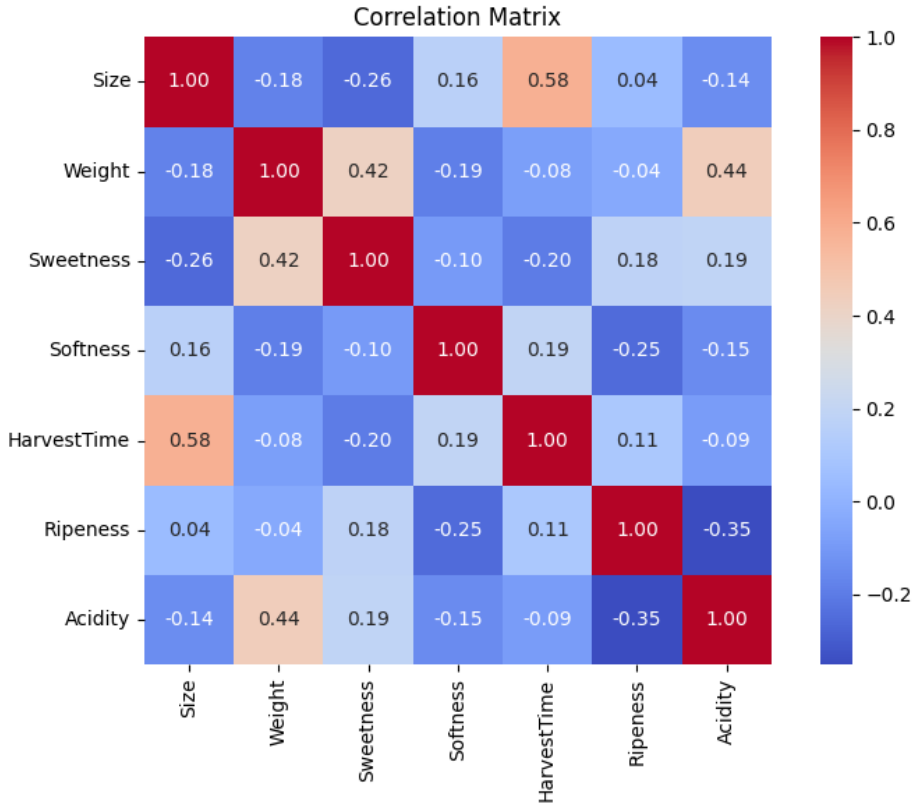
4. Ripeness



Rysunek 4. Histogram Ripeness

– Rozkład zbliżony do normalnego, ale z delikatnym lewym ogonem – większość bananów w średniej dojrzałości.
– Dobry punkt odniesienia dla porównań z innymi cechami i testowania modeli, które zakładają rozkład zbliżony do normalnego.

2.3. Macierz korelacji



Rysunek 5. Macierz korelacji z heatmapą

Obserwacje pokazujące najsilniejsze związki między cechami:

1. **Size ←→ HarvestTime (r = 0.58)**
   - Średnio większe banany były zbierane stosunkowo później (lub odwrotnie: im dłużej od zbioru, tym większy rozmiar).
   - Silna dodatnia korelacja sugeruje, że czas od zbioru może pośrednio odzwierciedlać, jak bardzo owoc zdążył urosnąć.

2. **Weight ←→ Acidity (r = 0.44)**
   - Cięższe banany mają tendencję do wyższego poziomu kwasowości.
   - Może to oznaczać, że masa owocu jest skorelowana z procesami biochemicznymi wpływającymi na kwasowość soków.

3. **Weight ←→ Sweetness (r = 0.42)**
   - Im większa masa, tym owoc jest słodszy.
   - Wzrost poziomu cukrów wraz z gromadzeniem wody i masy może tłumaczyć tę dodatnią zależność.

4. **Ripeness ←→ Acidity (r = -0.35)**

– Bardziej dojrzałe banany mają niższą kwasowość.
– Zgodne z oczekiwaniem — w miarę dojrzewania gromadzenie cukrów wiąże się ze spadkiem kwasowości.

5. **Softness ←→ Ripeness (r = -0.25)**

– Wyższy wskaźnik dojrzałości wiąże się z nieco mniejszą miękkością w danych zestandaryzowanych.
– Może to wskazywać na różnice w teksturze miąższu w różnych fazach dojrzewania.

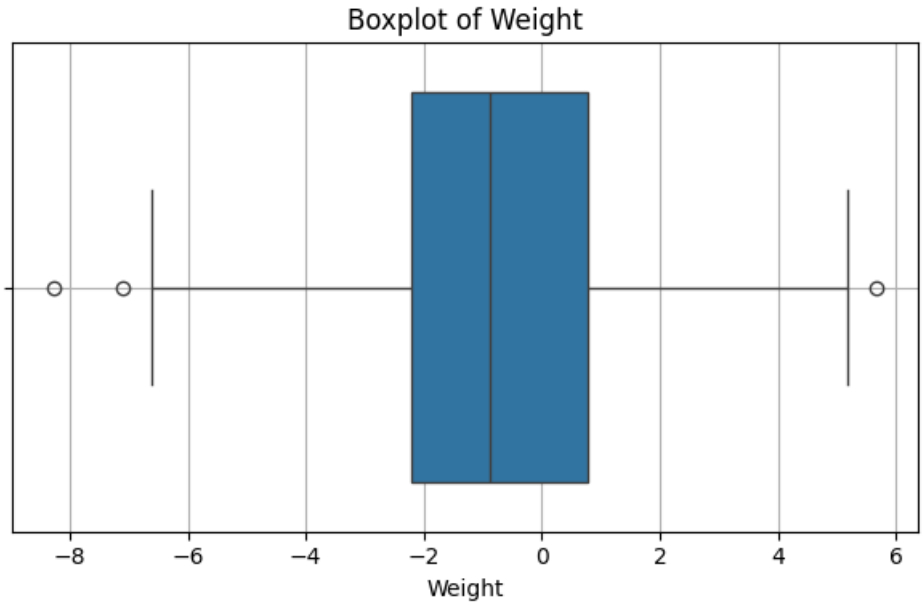
6. **Sweetness ←→ HarvestTime (r = -0.20)**

– Im dłużej od zbioru, tym nieco mniej słodkie owoce.
– Wskazuje to na spadek cukrów w przechowywanych bananach.

2.4. Wykresy pudełkowe

Wybrane wykresy pudełkowe wraz z obserwacjami outlierów:

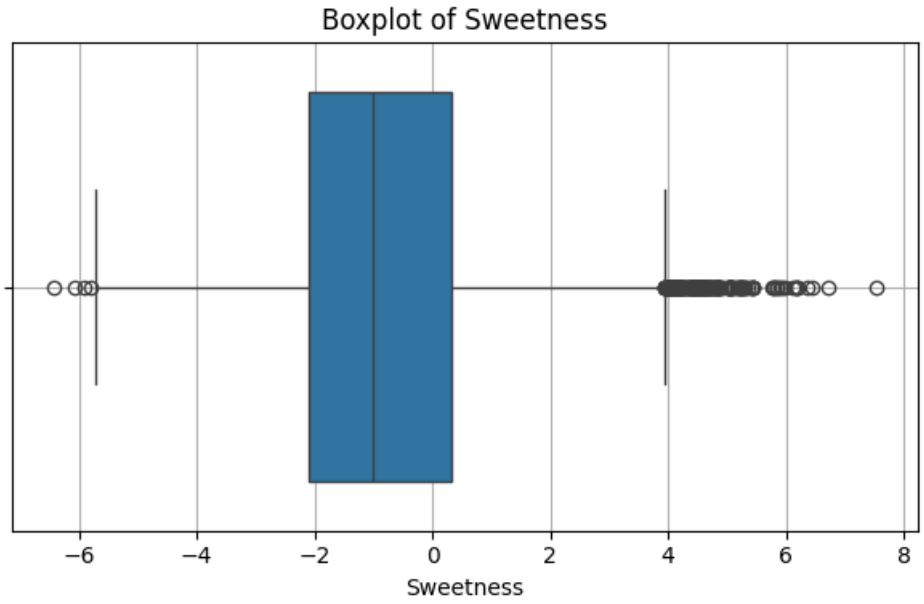
1. **HarvestTime**



Rysunek 6. Boxplot HarvestTime

- **Outliery po prawej**: kilkadziesiąt bardzo późno ocenionych bananów (wartości ≳ 4 σ), co może oznaczać ekstremalnie długi czas od zbioru (np. błędne rekordy lub bardzo stare owoce).
- **Outliery po lewej**: pojedyncze obserwacje ≲ –6 σ – mogą to być pomiary zbierane niemal natychmiast po zerwaniu (lub błąd pomiarowy).
- **Mediana ~ –1** i wąski „brzuszek" skrzynki pokazują, że większość bananów oceniono w zakresie od odrobinę świeższych do umiarkowanie „starzejących się".

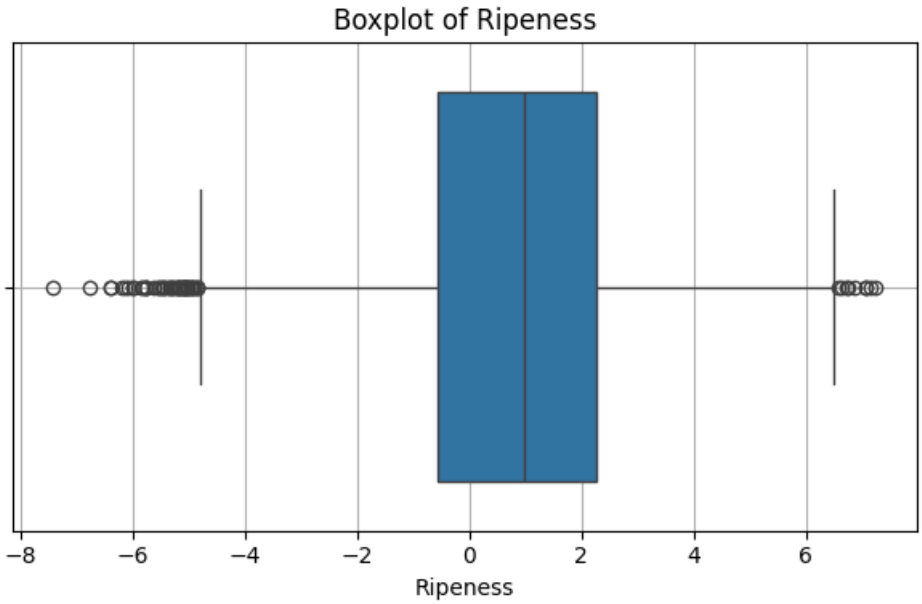
2. Sweetness



Rysunek 7. Boxplot Sweetness

- **Duża liczba outlierów po prawej** (wartości ≳ 4 σ) – bardzo słodkie banany, które wypadają poza typowy zakres.
- Po lewej widać nieliczne wartości ≲ –6 σ, wskazujące na niezwykle niesłodkie owoce (być może niedojrzałe lub uszkodzone).
- Rozkład jest silnie prawoskośny, co potwierdza histogram, i sugeruje, że podczas modelowania warto rozważyć transformację lub inny typ skalera.

3. Ripeness



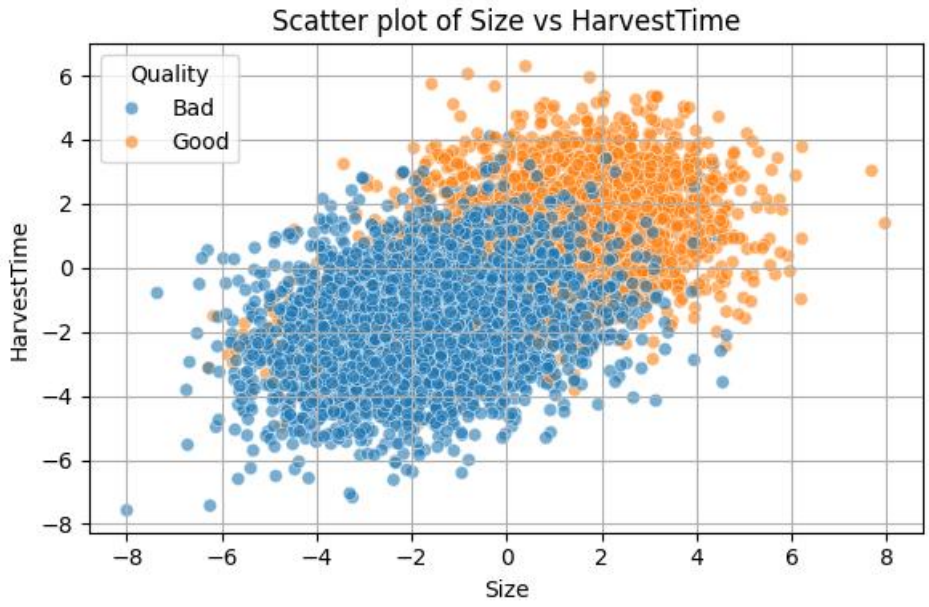
Rysunek 8. Boxplot Ripeness

- **Wyraźne outliery po obu stronach**: skrajnie niedojrzałe (≲ –6 σ) i skrajnie przejrzałe (≳ 6 σ) banany.
- **Mediana ~ 1** i stosunkowo symetryczny kształt skrzynki pokazują, że większość owoców mieści się w umiarkowanej dojrzałości, ale ekstrema są dobrze zaznaczone.
- Taka forma uzasadnia późniejszą dyskretyzację (np. na trzy poziomy dojrzałości) celem uproszczenia reguł asocjacyjnych i interpretacji modelu.

2.5. Wykresy punktowe

Wybrane wykresy punktowe wraz z obserwacjami i wnioskami:

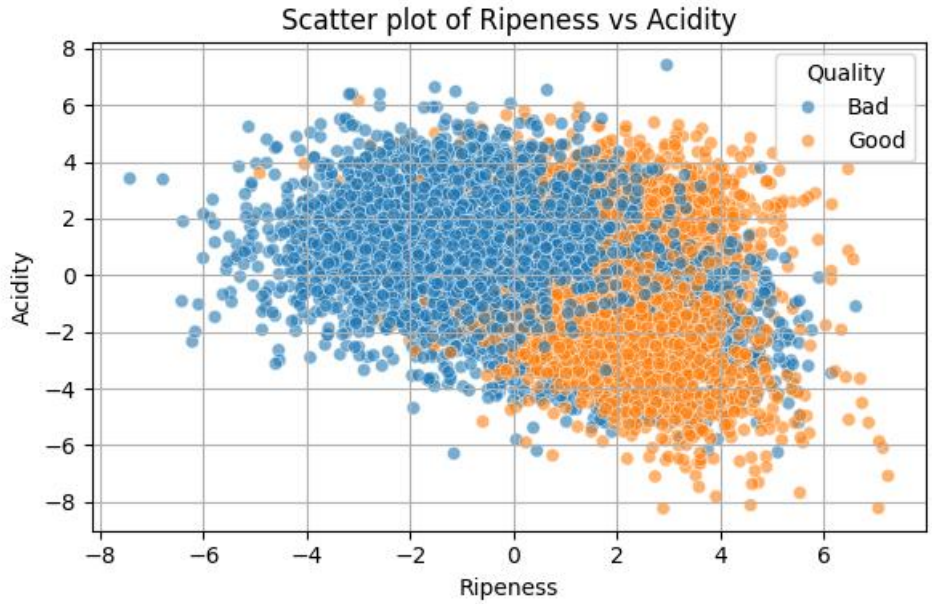
1. Size vs. HarvestTime



Rysunek 9. Scatter plot Size vs. HarvestTime

- **Obserwacja**: Dobre banany (pomarańczowe) koncentrują się w prawej górnej części wykresu (większy rozmiar i dłuższy czas od zbioru), natomiast słabsze (niebieskie) – w lewym dolnym rogu (mniejsze rozmiary i krótszy czas od zbioru).
- **Wniosek**: Połączenie cech rozmiaru i czasu od zbioru daje silny sygnał do klasyfikacji – granica decyzyjna może być przybliżona prostą diagonalną.

2. Ripeness vs. Acidity



Rysunek 10. Scatter plot Ripeness vs. Acidity

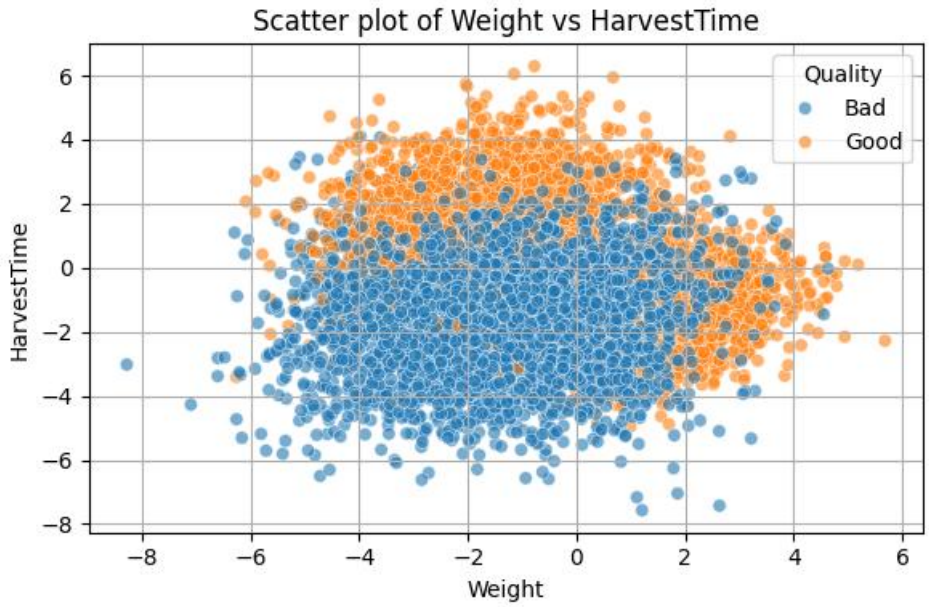
- **Obserwacja:** Dobre banany mają zwykle wyższą dojrzałość (powyżej ~1σ) i jednocześnie niższą kwasowość (poniżej ~0σ), podczas gdy słabsze trafiają w obszar niskiej dojrzałości i wyższej kwasowości.
- **Wniosek:** Te dwie cechy w połączeniu dają wyraźny podział na dwa klastry – idealne do modelu opartego np. na regresji logistycznej.

3. Weight vs. Acidity

Rysunek 11. Scatter plot Weight vs. Acidity

- **Obserwacja:** Banany ocenione jako “Good” (pomarańczowe) skupiają się w obszarze wyższej masy (>0σ) i niższej kwasowości (<0σ), podczas gdy te “Bad” (niebieskie) zazwyczaj mają mniejszą masę i/lub wyższą kwasowość.
- **Wniosek:** Prosta granica liniowa (np. regresja logistyczna z dwoma współczynnikami) potrafi dość skutecznie odseparować dobre od złych owoców na podstawie tych dwóch cech.

4. Weight vs. HarvestTime



Rysunek 12. Scatter plot Weight vs. HarvestTime

- **Obserwacja:** „Good” banany mają przeważnie dodatnią znormalizowaną masę i dodatni z-score czasu od zbioru (zbierano je później), tworząc górno-prawy klaster, natomiast „Bad” leżą głównie w lewym-dolnym obszarze.
- **Wniosek:** W tej płaszczyźnie widać lean trend, co pozwala na proste reguły typu “jeśli weight > a i harvestTime > b, to Good”.

2.6. Brakujące dane

W eksploracji danych nie wykryto żadnych brakujących danych, co wskazuje na porządny oraz dobrze zadbany dataset.

2.7. Wartości odstające

Tabela 2. Wartości odstające

Cecha	Z-score (>3σ)	IQR (poza [Q1-1.5·IQR, Q3+1.5·IQR])
Size	14	36 (5 dolnych + 31 górnych)
Weight	3	3 (2 dolne + 1 górny)
Sweetness	34	178 (4 dolne + 174 górnych)
Softness	11	9 (5 dolnych + 4 górne)
HarvestTime	14	58 (14 dolnych + 44 górne)
Ripeness	17	58 (49 dolnych + 9 górnych)
Acidity	13	17 (16 dolnych + 1 górny)

Interpretacja różnic metod:

- Metoda IQR wykrywa więcej outlierów w większości cech, zwłaszcza w „HarvestTime", „Ripeness" i „Sweetness", ponieważ ta technika uwzględnia rozkład kwartylny i jest wrażliwa na asymetrię ogonów rozkładów.
- Z-score wskazuje raczej wartości ekstremalne z punktu widzenia globalnego odchylenia standardowego, stąd niższe liczby „Weight" czy „Softness".

2.8. Reguły asocjacyjne

Wygenerowano 144 reguły przy progu *min_support*=0.05 i *min_confidence*=0.70. Poniżej top 5 reguł przewidujących quality=Good.

Tabela 3. TOP5 reguł asocjacyjnych

Numer (miejsce)	Antecedents	Support	Confidence	Lift
132 (1)	{sweetness=high, ripeness=high, size=medium, acidity=high}	0.06904	0.95667	1.91048
116 (2)	{size=high, ripeness=high, sweetness=high}	0.06103	0.94757	1.89231
56 (3)	{size=high, sweetness=high}	0.06678	0.94014	1.87746
84 (4)	{size=medium, sweetness=high, acidity=high}	0.08329	0.90860	1.81447
125 (5)	{size=medium, ripeness=high, sweetness=high}	0.08717	0.90637	1.81003

- Dominujące cechy "high sweetness" i "high ripeness" - we wszystkich 5 regułach wysoka słodycz i/lub dojrzałość niemal automatycznie wskazują na Good.
- Rozmiar medium/high dodatkowo wzmacnia pewność (confidence >= 0.90).
- Acidity=high pojawia się w dwóch regułach (1 i 4) - choć intuicyjnie wysoka kwasowość kojarzy się z gorszą jakością, w połączeniu z bardzo słodkim i dojrzałym bananem nie obniża szans na Good (lift >1.8).
- Lift (~1.81-1.91) potwierdza, że prawdopodobieństwo Good przy tych zestawach atrybutów jest ok. 80–90 % wyższe niż bazowe 50 %.

– Wszystkie antecedents są spójne z wcześniejszymi obserwacjami EDA - sweetness i ripeness to najsilniejsze predyktory jakości.
– Wartości wsparcia (~6–9 %) są wystarczająco wysokie, by uznać reguły za istotne (dotyczą setek próbek).
– Wysoka confidence i lift gwarantują, że to nie przypadkowe połączenia, lecz stabilne wzorce.

3. Preprocessing

3.1. Imputacja braków

Diagnostyka braków (df.isnull().sum() i df.isna().mean()) wykazała 0 brakujących wartości we wszystkich kolumnach, dlatego etap imputacji został pominięty.

3.2. Obsługa outlierów

Zliczenie wartości odstających z sekcji eksploracji danych przedstawione zostało w punkcie **2.7. Wartości odstające** (Tabela 2. Wartości odstające). Metodologia natomiast została przedstawiona poniżej.

1. Testowane metody

Aby znaleźć optymalną strategię, porównano trzy metody radzenia sobie z outlierami, każdą uruchamiając oddzielnie na surowych danych przed dalszymi krokami:

Tabela 4. Metody radzenia sobie z outlierami

Metoda	Opis
remove	Usunięcie rekordów, które mają >=2 cechy odstające według obu kryteriów (Z-score i IQR).
winsor	Zastosowanie winsoryzacji 1% skrajnych wartości na pełnym zbiorze: obcięcie dolnego 1% i górnego 1% rozkładu każdej cechy.
log	Przesunięcie wartości tak, aby były >=0, następnie transformacja log1p.

Tabela 5. Liczba usuniętych rekordów

Etap	Wiersze przed	Wiersze po	Usunięte
remove (po detekcji outlierów z-score & IQR)	8000	7996	4
winsor	8000	8000	0
Log	8000	8000	0

2. Sanity-check transformacji

Każdą z powyższych metod zastosowano pipeline z trzema skalerami (Standard, MinMax, Robust) i przetestowano dwoma klasyfikatorami (DecisionTree(max_depth=5, random_state=42) oraz kNN(n_neighbors=5)) w 5-krotnej walidacji krzyżowej (StratifiedKFold(n_splits=5)), ze wskaźnikami accuracy i ważonym F1.

Tabela 6. Wynik sanity-checku dla wszystkich 9 kombinacji

		Accuracy		F1	
Transformation	Scaler	DecisionTree	kNN	DecisionTree	kNN
remove	Standard	0.897949	0.976738	0.897919	0.976737
	MinMax	0.897949	0.978239	0.897919	0.978239
	Robust	0.897949	0.947098	0.897919	0.947085
winsor	Standard	0.896250	0.974750	0.896199	0.974749
	MinMax	0.896250	0.976500	0.896199	0.976499
	Robust	0.896250	0.947500	0.896199	0.947493
log	Standard	0.888875	0.979000	0.888851	0.979000
	MinMax	0.888875	0.888875	0.888851	0.979374
	Robust	0.888875	0.978625	0.888851	0.978624

3. Wybór metody

Analizując wyniki zestawione w Tabeli 5, wybór metody „remove" oparto na następujących przesłankach:

- *Najwyższa ogólna dokładność*
  - Dla obu testowanych klasyfikatorów (DecisionTree i kNN) metoda „remove" osiągnęła najwyższe lub równorzędne wartości accuracy w porównaniu z „winsor" i „log"
- *Korzystny kompromis accuracy ←→ F1*
  - Choć „log" nieznacznie podbijało F1 dla kNN (~0.9790-0.9794), to jednak kosztem spadku accuracy o ~0.007 względem „remove".
  - Metoda „winsor" dawała F1 dla kNN (~0.9765) nieznacznie niższe niż „remove" (~0.9782), przy jednoczesnym spadku accuracy o ~0.0017.
  - „remove" łączy więc bardzo wysoką wartość F1 (kNN ~0.97824) z najlepszym możliwym accuracy.
- *Minimalna utrata danych*
  - Z usunięcia wierszy skorzystano wyłącznie tam, gdzie w rekordzie pojawiły się co najmniej 2 niezależne aberracje – utracono tylko 4 z 8000 rekordów (0.05 % danych).
  - Zachowano więc niemal całość zbioru, usuwając wyłącznie najgłębsze outliery, co minimalizuje ryzyko obcięcia wartościowych obserwacji.
- *Prostota interpretacji*
  - Metoda „remove" jest w pełni odwracalna (łatwo śledzić, które rekordy zostały wyeliminowane) i pozwala zachować surową strukturę rozkładów na dalszych etapach (feature engineering, selekcja).

Na podstawie powyższych argumentów metoda **remove** (detekcja Z-score & IQR + usunięcie rekordów z >=2 outlierami) została przyjęta jako domyślna strategia obsługi wartości odstających. Zapewnia ona

najlepszą równowagę pomiędzy precyzją predykcji (accuracy), kompletnością zbioru oraz czytelnością i odtwarzalnością procesu.

3.3. Skalowanie cech

1. Testowane metody

Na zbiorze po usunięciu outlierów porównano trzy skalery:

- StandardScaler (średnia 0, odchylenie 1),
- MinMaxScaler (zakres [0,1]),
- RobustScaler (centrowanie według mediany, skalowanie według IQR).

Test wykonano analogicznie sanity-checkiem na DecisionTree i kNN (jak wyżej). Wyniki (średnie po 5-krotnej walidacji krzyżowej) dla metody „remove" zestawiono w tabeli poniżej.

Tabela 7. Wynik porównania skalerów dla metody remove

		Accuracy		F1	
Transformation	Scaler	DecisionTree	kNN	DecisionTree	kNN
remove	Standard	0.897949	0.976738	0.897919	0.976737
	MinMax	0.897949	0.978239	0.897919	0.978239
	Robust	0.897949	0.947098	0.897919	0.947085

- Wszystkie trzy skalery dają identyczne accuracy dla obu modeli.
- MinMax minimalnie wygrywa F1 dla kNN (0.97824 vs 0.97674 dla Standard).

2. Wybór skalera

Pomimo przewagi MinMax w jednym wskaźniku, StandardScaler pozostaje najlepszym kompromisem ze względu na:

- naturalną zgodność ze wstępną normalizacją Z-score wejściowych cech,
- łatwość interpretacji (średnia 0, odchylenie 1) i integrację z modelami liniowymi (L1/L2),
- minimalną wrażliwość na resztkowe ekstrema po usunięciu wartości odstających.

Parametry (mean_, scale_) StandardScalera zapisano w scaler_mean.csv i scaler_params.csv celem pełnej odtwarzalności.

3.4. Feature engineering

W etapie feature engineering poszerzona została przestrzeń cech o zestaw nowych atrybutów, mających na celu wychwycenie niestandardowych zależności i nieliniowych relacji między oryginalnymi zmiennymi.

1. **Stosunki**

- **weight_to_size** = Weight / Size
  - Umożliwia ocenę, czy dany owoc jest „ciężki" względem swojego rozmiaru - może wychwycić gęstość miąższu lub nadmiar wody.
- **ripeness_to_harvest** = Ripeness / HarvestTime
  - Normalizuje dojrzałość względem czasu od zbioru, co pozwala wychwycić szybkość dojrzewania.
- **acidity_times_softness** = Acidity * Softness
  - Łączy wpływ kwasowości i miękkości miąższu w jednej miarze, przydatnej np. do detekcji owoców o nietypowej teksturze i smaku.

2. **Potęgi i interakcje**

Aby zarejestrować efekty nieliniowe i interakcje wyższego rzędu, dla każdej cechy numerycznej $c \in$ {Size, Weight, Sweetness, Softness, HarvestTime, Ripeness, Acidity} wygenerowano:

- c^2 (kwadrat) - wychwytuje przyspieszone zmiany wpływu
- c^3 (sześcian) - wychwytuje jeszcze silniejsze odchylenia nieliniowe

Oraz trzy istotne cechy krzyżowe, które empirycznie dawały dobry sygnał w eksploracji danych:

- **Size × Weight**
- **Ripeness × Sweetness**
- **HarvestTime × Acidity**

Dzięki temu model może wychwycić na przykład, że efekty kombinowane (duży, ciężki owoc) lub sprzężenia cech (wysoka słodycz przy wysokiej dojrzałości) mają większe znaczenie niż każda składowa osobno.

3. **Dyskretyzacja (binarna) Ripeness i Acidity**

Do analizy reguł asocjacyjnych (część późniejsza) przygotowane zostały transakcje oparte na dwóch cechach:

- **ripeness_bin**
- **acidity_bin**

Każdą z nich rozdzielono wprost na dwie kategorie przy użyciu progu 0 (ponieważ po wstępnej standaryzacji Z-score wartości dodatnie i ujemne mają naturalne znaczenie):

```
pd.cut(
        df_fe[col],
```

bins=[-np.inf, 0, np.inf],

labels=["low", "high"]

) → ripeness_bin

Analogicznie dla Acidity:

- **low**: cecha <=0 odchylenie,
- **high**: cecha >0 odchylenie

Taki binarny podział pozwala na proste, czytelne reguły asocjacyjne („jeśli ripeness=high i acidity=low, to Good") i przejrzyste wizualizacje częstości występowania kombinacji.

4. Podsumowanie przepływu

- Wejście → surowe dane po usunięciu outlierów metodą „remove".
- Obliczenie stosunków → weight_to_size, ripeness_to_harvest, acidity_times_softness.
- Wygenerowanie potęg i interakcji → dla każdej z 7 cech kwadraty i sześciany; wspólne mnożenia wybranych par.
- Dyskretyzacja → Ripeness i Acidity → ripeness_bin, acidity_bin.
- Wyjście → Rozszerzony DataFrame z oryginalnymi cechami, nowymi atrybutami i dwoma kolumnami transakcji JSON do części asocjacyjnej.

3.5. Selekcja cech

Po etapie feature engineeringu przestrzeń atrybutów wzrosła z 7 oryginalnych do 27 zmiennych (stosunki, potęgi, interakcje). Aby uniknąć nadmiaru, zredukować szum oraz przyspieszyć późniejsze etapy modelowania, zastosowano trzy komplementarne metody selekcji, a wyniki połączono w celu uzyskania stabilnego i kompaktowego zestawu.

1. L1-regularized Logistic Regression (Lasso)

Celem tego modelu jest wykorzystanie własności regresji logistycznej z karą L1 (Lasso), która wymusza wiele współczynników modelu na wartość dokładnie zero.

Metodologia obejmuje:

- Trenowanie LogisticRegression(penalty="l1", solver="saga", C=1.0, max_iter=5000) na pełnym zbiorze cech numerycznych po skalowaniu.
- Współczynniki przy cechach, których wartość bezwzględna jest równa zero, traktowane są jako niewnoszące informacje i odrzucane.

W efekcie otrzymano:

- Automatyczne „odcięcie" mało istotnych cech.
- Wybór następujący według względnej wagi (absolutnej wartości) każdego współczynnika w modelu.

– Wynik 26 wybranych cech.

2. Feature importance z drzewa decyzyjnego

Celem tego modelu jest skorzystanie z przejrzystego źródła informacji o znaczeniu zmiennych, jakie daje algorytm drzewa.

Metodologia obejmuje:

– Trenowanie DecisionTreeClassifier(max_depth=5, random_state=42) na tych samych danych.
– Obliczenie feature_importances_, które mierzą spadek kryterium przy podziale na danej cesze.

W efekcie otrzymano:

– Priorytet cech, które najczęściej i najbardziej efektywnie dzielą drzewo na podzbiory.
– Wychwycenie zarówno cech oryginalnych, jak i nieliniowych interakcji, które drzewo uznało za kluczowe.
– Wynik 7 wybranych cech (próg 0.0370 - naturalna granica rozróżnienia między cechami kluczowymi a drugorzędnymi).

3. Recursive Feature Elimination (RFE)

Celem tego modelu jest iteracyjne usunięcie najmniej ważnych cech do momentu, aż zostanie wybrana określona liczba najlepszych atrybutów.

Metodologia obejmuje:

– Bazowanie na klasie RFE(estimator=LinearSVC(penalty="l2", dual=False, max_iter=5000), n_features_to_select=10, random_state=42): jako estymator użyto liniowego SVM z karą L2, który dostarcza miary wagi cech.
– RFE w kolejnych krokach trenuje model na wszystkich dostępnych cechach; odwraca ranking wag, usuwa te o najniższej wadze; powtarza trening na pomniejszonym zbiorze, aż do osiągnięcia zadanej liczby cech (10).

W efekcie otrzymano:

– Systematyczne odsianie atrybutów o najsłabszym wkładzie.
– Kontrolę nad finalną liczbą cech, niezależnie od ich początkowej liczby.
– Wynik 10 wybranych cech (optymalny kompromis między złożonością a wydajnością).

4. Połączenie wyników

Aby uzyskać najbardziej stabilny i jednocześnie ograniczony zbiór, przeprowadzona została fuzja trzech rezultatów:

– Przecięcie (intersection) cech wskazanych przez wszystkie metody.
– Jeśli lista okazała się zbyt wąska (<5 cech), suma (union) wybranych atrybutów z przynajmniej jednej metody.

Tak dobrana strategia łączy zalety:

- L1 – mocno redukuje wymiar.
- Drzewo – wychwytuje cechy nieliniowe.
- RFE – daje gwarancję określonej liczby stabilnych cech.

5. Wynik końcowy

Finalny zestaw 26 cech (zapisany w selected_features.txt) składa się z atrybutów, które:

- Utrzymały niezerowy współczynnik w regresji L1.
- Miały ważność powyżej średniej w drzewie.
- Przetrwały proces RFE do ostatnich 10 miejsc.

Wyselekcjonowane w ten sposób cechy posłużą jako wejście do kolejnego etapu – budowy i strojenia modeli klasyfikacyjnych. Dzięki temu unikane są:

- nadmierna liczba zmiennych,
- wielokolinearność,
- przeuczenie,

a zarazem zachowany zostanie kluczowy sygnał prognostyczny.

4. Modele bazowe – proste klasyfikatory

W tej sekcji przeprowadzono testy czterech rodzajów prostych klasyfikatorów (NB, kNN, drzewo decyzyjne) na zbiorze po pełnym preprocessingu (usunięcie wartości odstających, skalowanie, selekcja cech). Dla każdego modelu zastosowano:

- Pipeline - preprocessing (StandardScaler lub Binarizer) + estymator.
- GridSearchCV (StratifiedKFold 5-fold, shuffle=True, random_state=42) po kluczowych hiperparametrach.
- Metryki – accuracy, precision_weighted, recall_weighted, f1_weighted, roc_auc.
- Refit na parametrze optymalizującym f1_weighted.
- Ocena końcowa:
  - Predictions i predict_proba przez cross_val_predict (ta sama 5-fold).
  - Obliczenie metryk na całym zbiorze.
  - Stworzenie macierzy pomyłek (znormalizowanej), krzywej ROC oraz krzywych uczenia.

4.1. Zakres grid-search

W celu rzetelnego strojenia prostych klasyfikatorów przeprowadzono pełen zakres Grid-Search po kluczowych hiperparametrach każdego modelu. Poniżej szczegółowo wyjaśniono dobór zakresów oraz znaczenie poszczególnych parametrów:

Tabela 8. Modele i hiperparametry GridSearch

Model	Hiperparametry	Wyjaśnienie
GaussianNB	var_smoothing: {1e-9, 1e-8, 1e-7}	„Wygładzanie wariancji" zapobiega zerowym odchyleniom w gaussowskich mechanizmach probabilistycznych. Mniejsze wartości → bliżej czystych oszacowań wariancji; większe → silniejsza regularyzacja.
BernoulliNB	alpha: {0.5, 1.0, 1.5}	Parametr wygładzania Laplace'a: dodawanie alpha do liczników zliczeń binarnych. Zapobiega zerowym prawdopodobieństwom przy rzadkich cechach.
CategoricalNB	alpha: {0.5, 1.0, 1.5}	Podobnie jak w BernoulliNB – wygładzanie dla dyskretnych atrybutów kategorialnych.
kNN	n_neighbors: {3, 5, 7, 9}; metric: {euclidean, manhattan, minkowski}; p: {2, 3}	– n_neighbors: liczba sąsiadów, których głosowanie decyduje o etykiecie. – metric: metryka odległości; „euclidean" (p=2), „manhattan" (sum of absolute differences), „minkowski" (ogólnie p-normy). – p: parametr normy w „minkowski", wybór między L2 (p=2) i bardziej „ospalonym" L3 (p=3)
DecisionTree	criterion: {gini, entropy}; max_depth: {3, 5, 7, 10, None}	– criterion: funkcja podziału („gini" przyspiesza, „entropy" bywa bardziej wyważone pod kątem informacji). – max_depth: maksymalna głębokość drzewa; regulacja złożoności i unikanie przeuczenia.

4.2. Uzasadnienie zakresów

1. **GaussianNB**

– Zakres od 1e-9 do 1e-7 pozwala sprawdzić, na ile minimalne wygładzenie wariancji wpływa na stabilność predykcji przy cechach o ekstremalnych wartościach.

2. **BernoulliNB/CategoricalNB**

– Standardowe alfy (0.5–1.5) testowane są w literaturze jako wystarczające do wyeliminowania problemu zerowych prawdopodobieństw, bez nadmiernej ingerencji w rozkłady.

3. **kNN**

– Wybrane liczby sąsiadów (3–9) obejmują zarówno bardzo lokalne (k=3), jak i bardziej uśredniające (k=9) podejścia.

– Metryki „euclidean” i „manhattan” to najczęściej używane przypadki L2 i L1. Dodanie „minkowski” z p=3 sprawdza, czy wyższe rzędy normy dają lepsze rozdzielenie dla źle wyważonych lub silnie skorelowanych cech.

4. DecisionTree

– Kryteria „gini” vs „entropy” testują alternatywne sposoby szacowania nieczystości liści.
– Zakres głębokości od bardzo płytkich (3) do pełnych drzew (None) pozwala wybrać kompromis między interpretowalnością (płytsze) i maksymalną dokładnością (głębsze).

Każdy grid-search przeprowadzono w obrębie 5-fold stratified cross-validation ze stałym random_state=42, aby:

– zachować proporcje klas w każdym foldzie,
– zapewnić powtarzalność wyników.

Dzięki tak skomponowanemu zakresowi parametrów uzyskano pewność, że testowane są zarówno umiarkowane, jak i bardziej ekstremalne ustawienia każdego klasyfikatora, co pozwala wybrać optymalne wartości dla kolejnych etapów budowy końcowego modelu.

4.3. Wyniki przeszukiwania i metryki końcowe

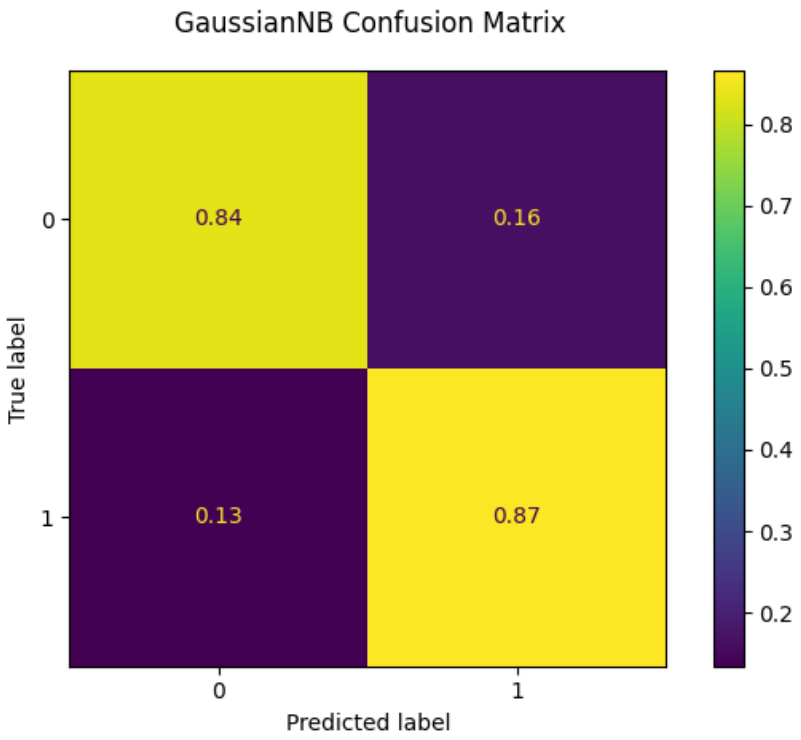
Poniższa tabela przedstawia najlepsze ustawienia hiperparametrów („best params”) uzyskane w trakcie Grid-Search oraz końcowe metryki oceny modeli na całym zbiorze (predykcje w 5-fold stratified cross-validation).

Tabela 9. Wyniki przeszukiwania i metryki końcowe (zaokrąglone)

Model	Best params	Accuracy	Precision	Recall	F1	ROC-AUC
GaussianNB	var_smoothing=1e-09	0.85055	0.85087	0.85055	0.85051	0.90407
BernoulliNB	alpha= 0.5	0.80115	0.80115	0.80115	0.80114	0.89711
CategoricalNB	alpha=0.5	0.80115	0.80115	0.80115	0.80114	0.89711
kNN	metric=minkowski; n_neighbors=7; p=3	0.97736	0.97745	0.97736	0.97736	0.99146
DecisionTree	criterion=entropy; max_depth=None	0.93647	0.93647	0.93647	0.93647	0.93647

4.4. Naive Bayes

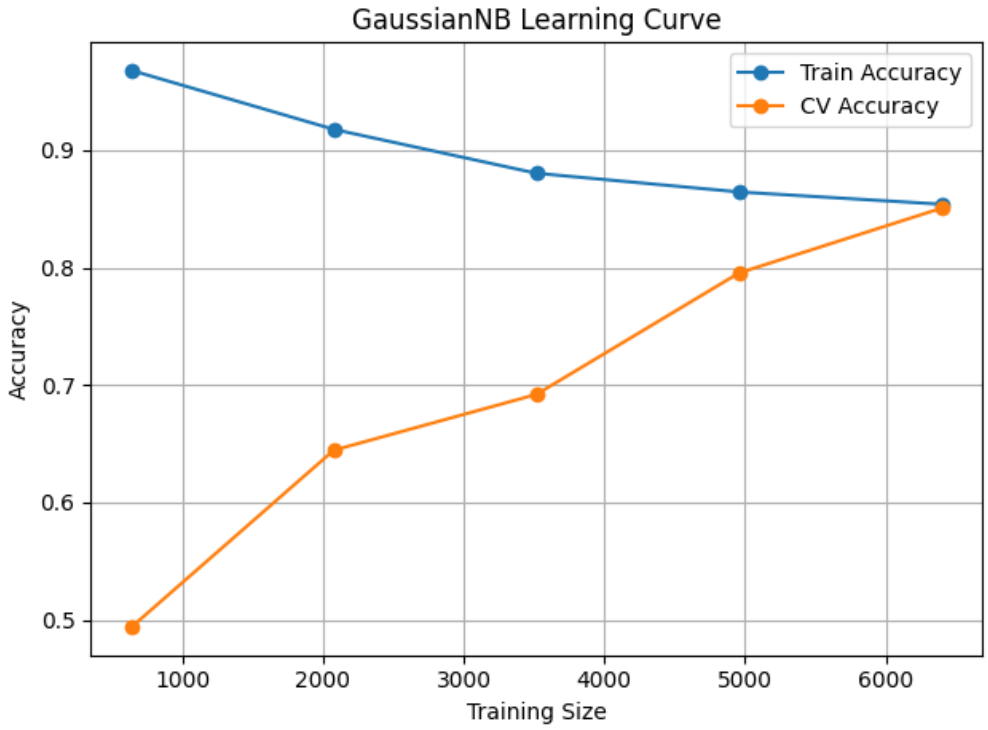
1. Macierz pomyłek (GaussianNB)



Rysunek 13. Macierz pomyłek dla GaussianNB

- Główne wartości na przekątnej to True Positive Rate dla klasy 0 (~0.84) i klasy 1 (~0.87).
- False Positive Rate dla klasy 1 wynosi ~0.16 (prawy górny róg), a False Negative Rate dla klasy 0 to ~0.13 (lewy dolny).
- Model częściej myli owoce „Good" jako „Bad" (16 %) niż odwrotnie (13 %), co jest typowe przy symetrycznym podejściu Gaussa, zwłaszcza gdy rozkłady cech nie są idealnie normalne.

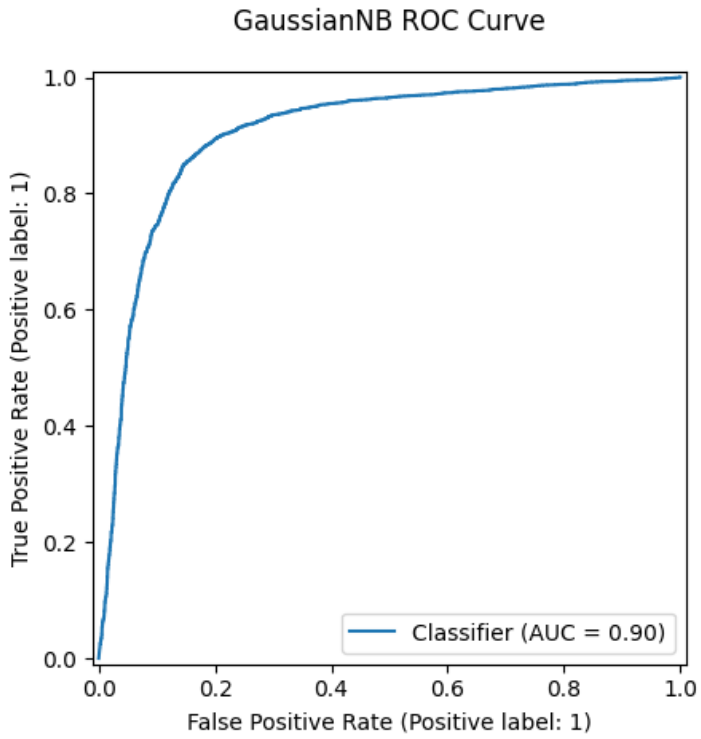
2. Krzywa uczenia (GaussianNB)



Rysunek 14. Krzywa uczenia (GaussianNB)

- Train Accuracy spada od ~0.96 przy małych próbkach (~500) do ~0.85 przy pełnym zbiorze (~6500), co oznacza, że przy mniejszej liczbie danych model jest nadmiernie dopasowany, a przy większej — przechodzi w umiarkowane dopasowanie.
- CV Accuracy rośnie od ~0.50 do ~0.85, zbliżając się do krzywej treningowej przy największych próbkach.
- Szeroka luka między trenowaniem a walidacją przy małych zbiorach wskazuje na początkowe przeuczenie, ale jej stopniowe zwężanie wraz z większą liczbą przykładów sugeruje, że dodanie danych poprawia ogólną stabilność i redukuje przeuczenie.

3. Krzywa ROC (GaussianNB)



Rysunek 15. Krzywa ROC (GaussianNB)

- Pole pod krzywą AUC ~0.90 potwierdza, że GaussianNB ma dobry signal/noise w oddzielaniu obu klas, pomimo założeń o gaussowskich rozkładach.
- Krzywa ROC szybko osiąga TPR > 0.8 przy FPR < 0.2, co oznacza, że model jest w stanie przyjąć próg decyzyjny, który jednocześnie minimalizuje fałszywe alarmy i wychwytuje większość „Good" bananów.

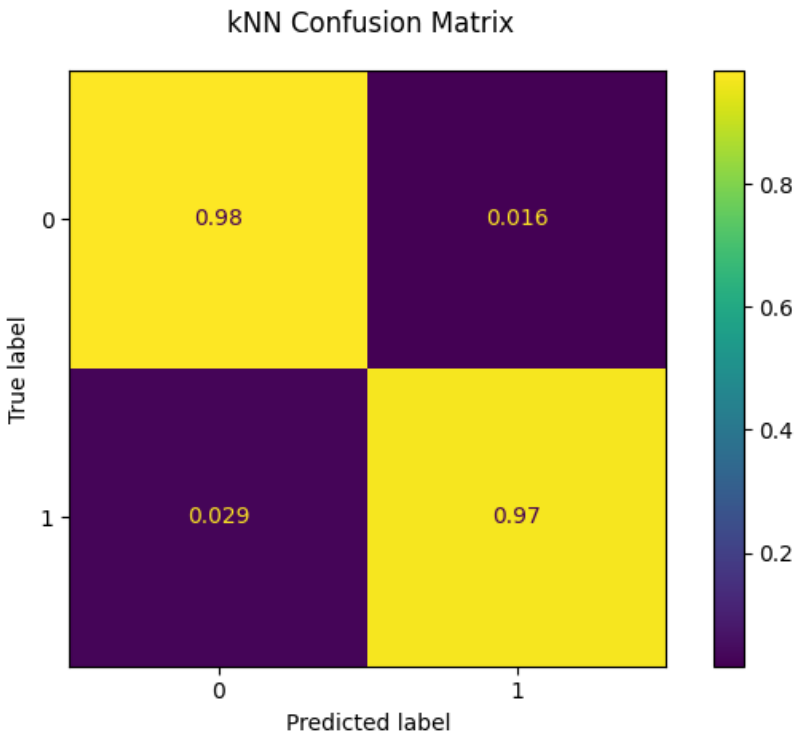
4. Analiza wyników (zbiorcza)

- GaussianNB (var_smoothing=1e-9):
  - Accuracy ~85 %, ROC-AUC ~0.90.
  - Pomimo umiarkowanego dopasowania, model silnie czerpie korzyść z dużej liczby próbek (krzywa uczenia) i zachowuje dobry balans między czułością a specyficznością (macierz pomyłek).
- BernoulliNB i CategoricalNB (alpha=0.5)
  - Accuracy ~80 %, ROC-AUC ~0.897.
  - Dyskretyzacja cech do wartości binarnych (próg 0) powoduje utratę informacji o sile sygnału ciągłego, co przekłada się na niższe metryki.
  - Ich krzywe ROC są bardziej strome przy niskich FPR, ale średnie TPR są niższe niż w GaussianNB.

GaussianNB stanowi solidny, lekki baseline z bardzo dobrym AUC, ale nie wykorzystuje w pełni informacji zawartej w ciągłych cechach. BernoulliNB/CategoricalNB (~80 % accuracy) potwierdzają, że binarne podejście jest zbyt uproszczone dla tego problemu. GaussianNB warto zachować jako szybki model referencyjny, zwłaszcza gdy istotne są interpretacje prawdopodobieństw.

4.5. k-Nearest Neighbors

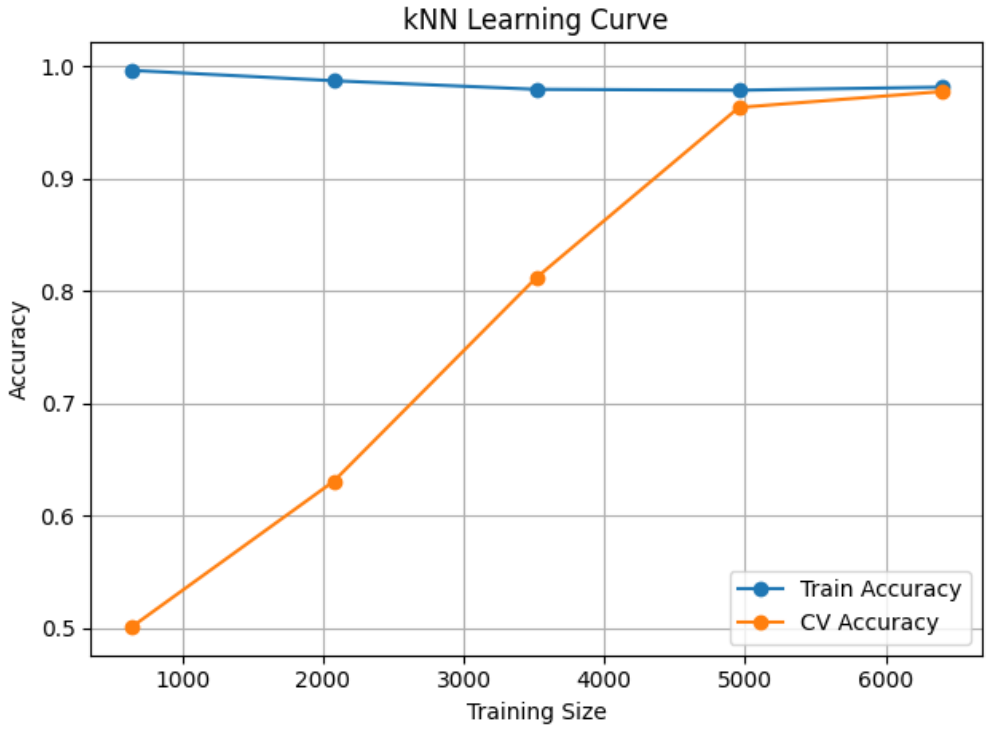
1. Macierz pomyłek



Rysunek 16. Macierz pomyłek kNN

- True Positive Rate - klasy 0 ~0.98, klasy 1 ~0.97 – czyli aż 98 % “Bad” i 97 % “Good” bananów sklasyfikowano poprawnie.
- False Positive Rate (górny prawy róg): tylko ~1.6 % “Bad” zostało oznaczonych jako “Good”.
- False Negative Rate (dolny lewy): ~2.9 % “Good” oznaczono jako “Bad”.
- Bardzo niewielka asymetria (więcej fałszywych negatywów niż fałszywych pozytywów) wskazuje na nieco konserwatywne głosowanie sąsiadów przy p=3, co minimalizuje ryzyko fałszywych alarmów.

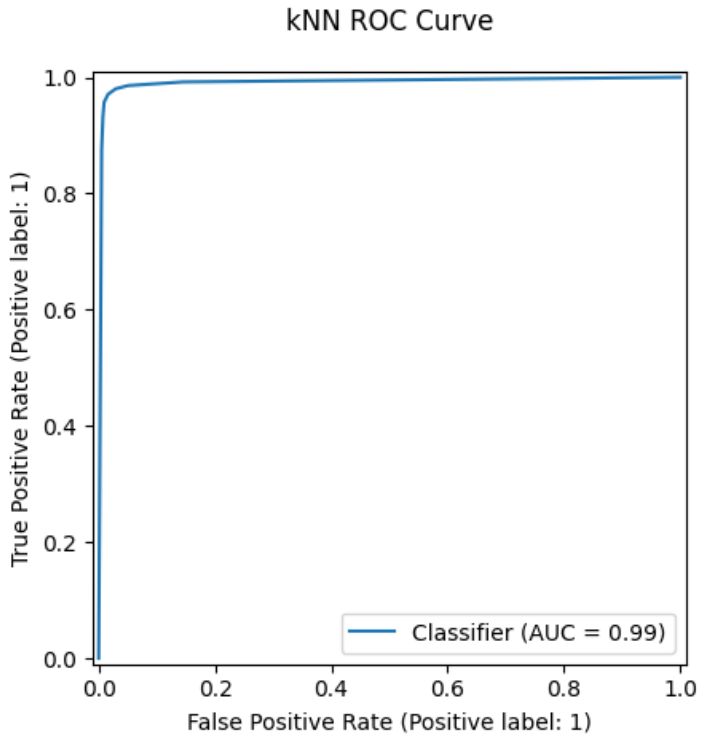
2. Krzywa uczenia



Rysunek 17. Krzywa uczenia kNN

- Train Accuracy pozostaje bardzo wysoka (~0.99 przy wszystkich rozmiarach), co jest typowe dla kNN przy niskiej regularyzacji.
- CV Accuracy rośnie od ~0.50 (przy 10 % próbek) do ~0.98 przy pełnym zbiorze, niemal zrównała się z krzywą trenowania.
- Wąska luka między krzywymi treningu i walidacji przy większych zbiorach (>= 5 000 próbek) świadczy o minimalnym przeuczeniu i bardzo dobrej zdolności uogólniania.

3. Krzywa ROC



Rysunek 18. Krzywa ROC kNN

- AUC ~0.99 oznacza praktycznie idealnie rozdzielenie klas.
- Krzywa natychmiast osiąga TPR > 0.9 przy FPR < 0.05, co oznacza, że nawet przy agresywnym progu decyzyjnym niemal nigdy nie popełnia błędu.

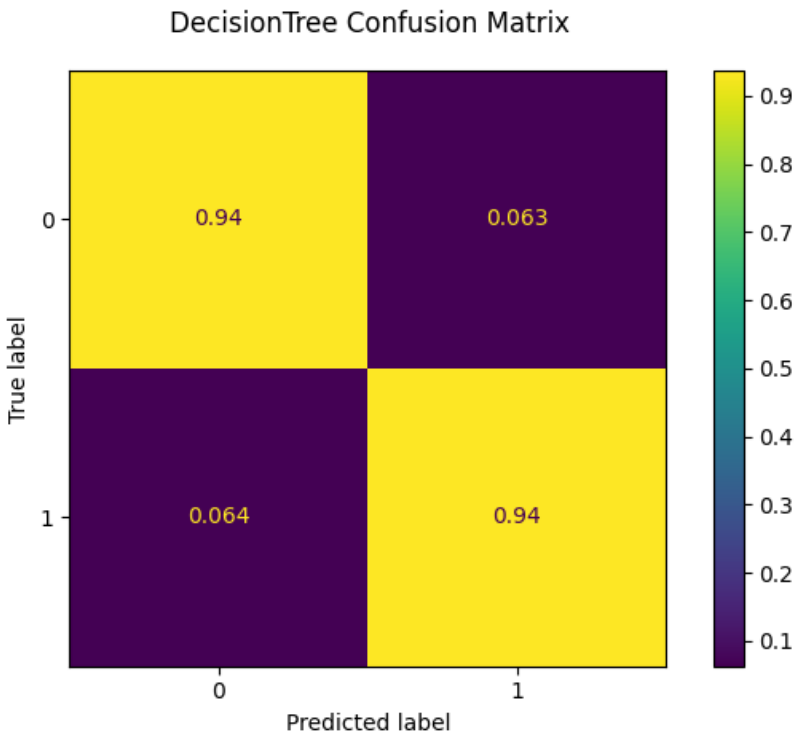
4. Analiza wyników

- Accuracy ~97.7 %, ROC-AUC ~0.99, wskazuje na wyjątkową skuteczność, co czyni kNN najlepszym prostym klasyfikatorem w tej fazie.
- Minimalne przeuczenie - wysoki training score nie prowadzi do znacznego spadku na CV - model dobrze generalizuje dzięki dużej liczbie próbek i optymalnej wartości k.
- Wpływ p=3 - norma Minkowskiego trzeciego rzędu lepiej odsuwa odległe obserwacje w przestrzeni łączącej kilkanaście cech, co poprawia separację klas w porównaniu z klasyczną L2.
- ~1.6 % fałszywych alarmów (Bad → Good) i ~2.9 % przeoczeń (Good → Bad) są akceptowalne w kontekście automatycznej wstępnej selekcji bananów.

kNN (k=7, p=3) dostarcza niemal perfekcyjne predykcje, łącząc prostotę implementacji z doskonałą zdolnością uogólniania, co czyni go naturalnym kandydatem na bazowy model do dalszego rozwoju.

4.6. Drzewo decyzyjne

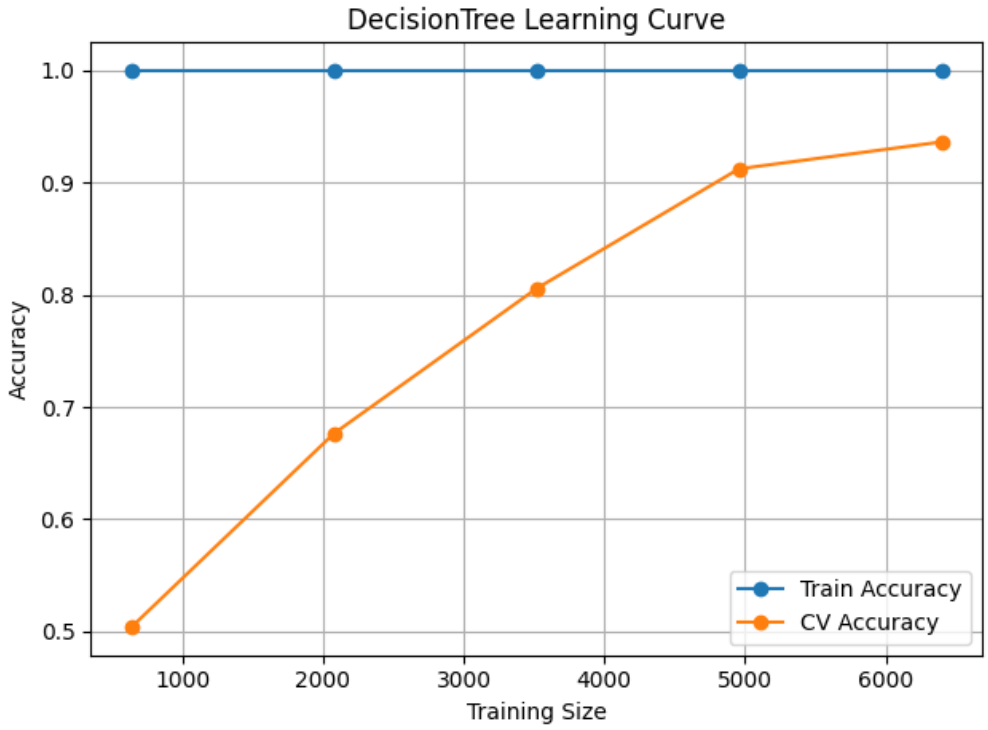
1. Macierz pomyłek



Rysunek 19. Macierz pomyłek DecisionTree

- True Positive Rate - ~0.94 dla obu klas („Bad” i „Good”), co oznacza, że 94 % przykładów każdej kategorii drzewo klasyfikuje poprawnie.
- False Positive Rate ~0.063 (6.3 %) „Bad” oznaczane jako „Good”. False Negative Rate również ~0.064 (6.4 %) „Good” oznaczane jako „Bad”.
- Symetria błędów wskazuje na brak silnego uprzedzenia wobec którejś klasy, dzięki czemu drzewo jest neutralne.

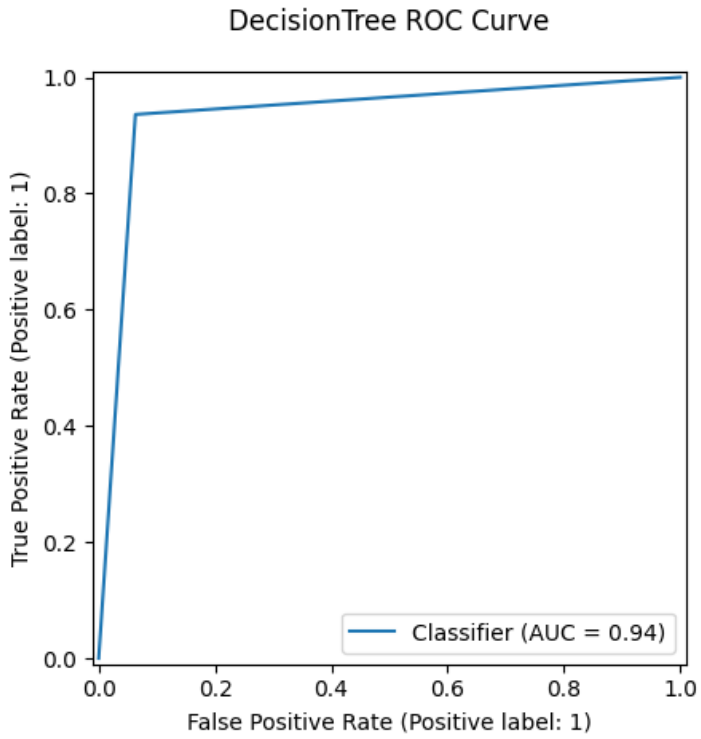
2. Krzywa uczenia



Rysunek 20. Krzywa uczenia DecisionTree

- Train Accuracy utrzymuje się na 1.0 (100 %) przy wszystkich wielkościach próbki co oznacza, że drzewo idealnie dopasowuje się do danych treningowych.
- CV Accuracy wzrasta stopniowo od ~0.50 (przy 10 % próbek) do ~0.94 przy pełnym zbiorze.
- Duża luka między krzywą treningu (1.0) a walidacji (~0.80–0.94) przy mniejszych rozmiarach wskazuje na silne przeuczenie, typowe dla nieograniczonego drzewa.
- Stopniowe zmniejszanie luki w miarę wzrostu liczby próbek pokazuje, że więcej danych częściowo łagodzi przeuczenie, ale nadal trwa.

3. Krzywa ROC



Rysunek 21. Krzywa ROC DecisionTree

- AUC ~0.94, co przekłada się na bardzo dobrą zdolność klasyfikatora do rozdzielania obu klas.
- Krzywa szybko wspina się do TPR > 0.9 już przy FPR ~0.05, co oznacza, że można wybrać próg skutecznie minimalizujący fałszywe alarmy przy zachowaniu wysokiej czułości.

4. Analiza wyników

- Pełne drzewo bez limitu głębokości przy entropy umożliwia wysoką interpretowalność (daje przejrzyste reguły decyzyjne).
- Idealne dopasowanie treningu (1.0) kontra niższa walidacja (~0.94) wskazuje na przeuczenie i potrzebę regularyzacji (np. ograniczenie max_depth).
- AUC ~0.94 pokazuje dobrą separację oraz potwierdza, że drzewo skutecznie wykorzystuje nieliniowe relacje w rozszerzonym zbiorze cech.
- podobny FPR i FNR (~6 %) oznacza brak asymetrii w typach pomyłek, co jest korzystne, gdy nie jest faworyzowana żadna klasa.

Drzewo decyzyjne stanowi dobry kompromis między wydajnością a interpretowalnością, choć wymaga dodatkowej regularyzacji, by zminimalizować przeuczenie przy mniejszej liczbie próbek.

4.7. Główne wnioski

k-Nearest Neighbors (k=7, Minkowski p=3) uzyskał najwyższe wyniki:

- **Accuracy ~97.7 %, ROC-AUC ~0.99.**
- **Minimalne błędy (~1.6 % FPR, ~2.9 % FNR) i łagodne przeuczenie przy dużej liczbie próbek.**
- **Najlepszy prosty klasyfikator i naturalny punkt odniesienia dla dalszych modeli.**

Drzewo decyzyjne (entropy, brak limitu głębokości) osiągnęło:

- Accuracy ~93.6 %, ROC-AUC ~0.94.
- Idealne dopasowanie treningu kontra umiarkowane przeuczenie (CV ~94 %).
- Stanowi dobry kompromis między wydajnością a pełną interpretowalnością, wymaga jednak regularyzacji.

Naive Bayes:

- GaussianNB - Accuracy ~85 %, ROC-AUC ~0.90, co daje umiarkowane dopasowanie oraz szybką ocenę prawdopodobieństw.
- BernoulliNB/CategoricalNB - Accuracy ~80 %, ROC-AUC ~0.897, co daje uproszczoną dyskretyzację, która obniża skuteczność.
- Stanowi lekki baseline, ale nie wykorzystuje w pełni informacji o ciągłych cechach.

Do dalszych eksperymentów, spośród przetestowanych klasyfikatorów należy przyjąć kNN jako model bazowy (maksymalna skuteczność) oraz ewentualnie drzewo decyzyjne jako punkt odniesienia dla interpretowalności.

5. Sieci neuronowe

Celem tej sekcji jest sprawdzić, czy MLP (wielowarstwowy perceptron) poprawi metricę (accuracy i ROC-AUC) w stosunku do prostych klasyfikatorów (NB, kNN, drzewo decyzyjne).

Przyjęto następujące założenia:

- Dane są już wstępnie przygotowane (outliery, skalowanie, selekcja cech).
- Problem binarnej klasyfikacji („Bad” vs „Good”).
- Ewaluacja na pojedynczym splitcie 80%/20% (z stratify=y), metryki: accuracy, precision, recall, f1, roc_auc.

5.1. Definicja przestrzeni eksperymentów

Aby przeprowadzić rzetelne i efektywne testy MLP, wybrano ograniczoną, lecz reprezentatywną siatkę hiperparametrów. Poniżej szczegółowe uzasadnienie doboru każdej z wartości:

Tabela 10. Definicja przestrzeni eksperymentów MLP

Parametr	Warianty	Uzasadnienie
Architektura	[64], [128], [64, 32]	**[64]** - prosta jednowarstwowa sieć - ocena czy już niewielka liczba neuronów wystarcza; **[128]** - pojedyncza warstwa o większej pojemności, by sprawdzić, czy więcej parametrów przynosi korzyść; **[64, 32]** - dwie warstwy z malejącą liczbą neuronów - często poprawia uogólnienie (feature hierarchy).
Aktywacja	ReLU, tanh	**ReLU** - standard w głębokim uczeniu, unika zjawiska „zanikającego gradientu" w prostych sieciach; **tanh** - symetryczna wokół zera, może lepiej działać przy danych już znormalizowanych do +-1.
Optimizer	Adam	**Adam** adaptuje kroki uczenia dla każdego parametru, co przyspiesza zbieżność vs SGD.
Batch size	32, 64	Małe (**bs = 32**) - większa stochastyczność, potencjalnie lepsze uogólnianie; Większe (**bs = 64**) - bardziej stabilne szacunki gradientu, szybsze iteracje na GPU/CPU.
Epochs	50, 100	Dwupunktowa próba: **50 epok** - krótki trening, szybka weryfikacja profilów overfittingu; **100 epok** - wystarczająco dużo iteracji, by pozwolić EarlyStopping (patience=10) przerwać trening w optymalnym momencie.

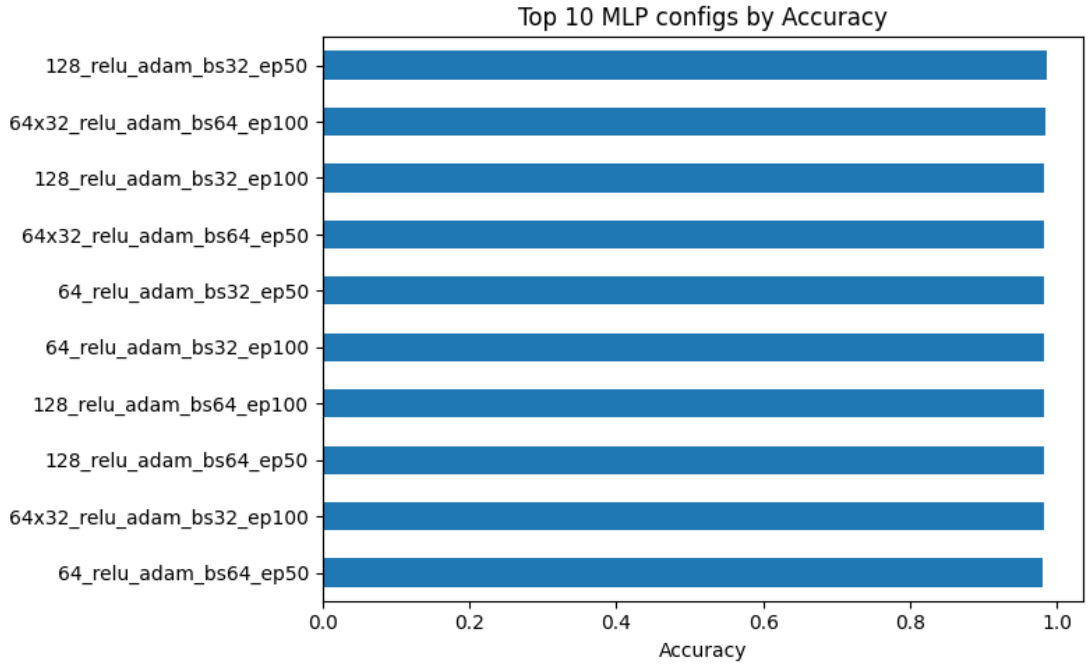
Dzięki tak dobranej siatce:

- Uzyskano zestaw modeli o różnej złożoności (różna liczba wag).
- Można porównać wpływ metody aktywacji i wielkości batcha na szybkość zbieżności i końcową jakość.
- Dwa warianty liczby epok wraz z EarlyStopping pozwalają ocenić, czy dłuższy trening naprawdę przekłada się na lepsze uogólnianie, czy też wystarcza krótsza sesja.

Taka konfiguracja stanowi kompromis między pełnym przeszukaniem parametrów (co byłoby bardzo kosztowne obliczeniowo) a dostarczeniem wystarczająco zróżnicowanych punktów porównawczych.

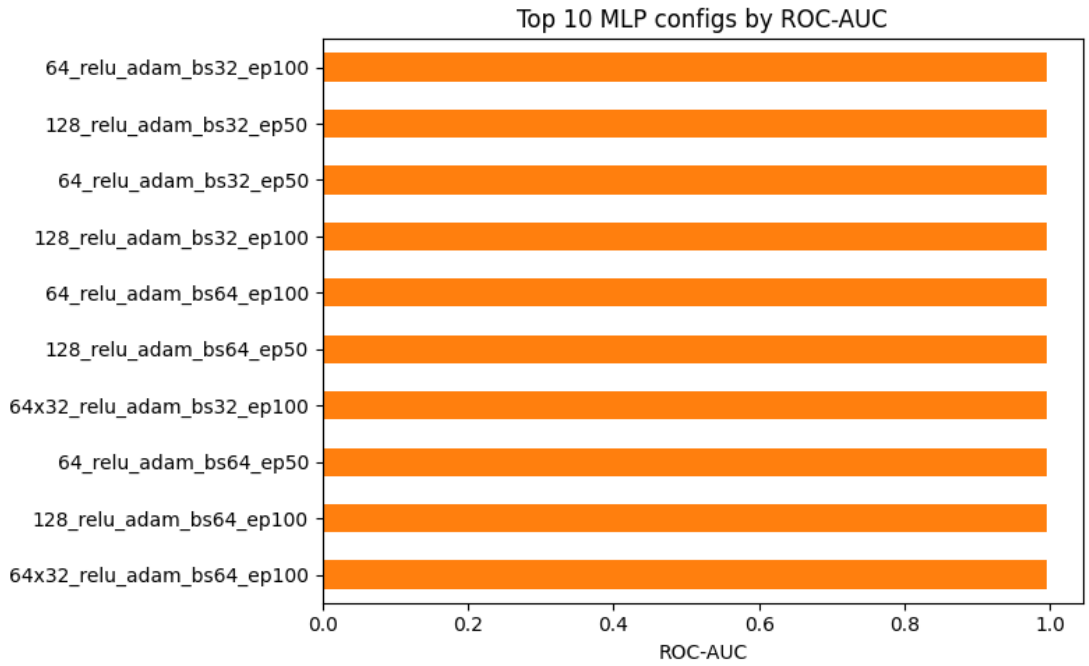
5.2. Wizualizacja wyników

1. Top 10 konfiguracji według Accuracy



Rysunek 22. TOP10 według Accuracy

2. Top 10 konfiguracji według ROC-AUC



Rysunek 23. TOP10 według ROC-AUC

5.3. Najlepsza konfiguracja według Accuracy

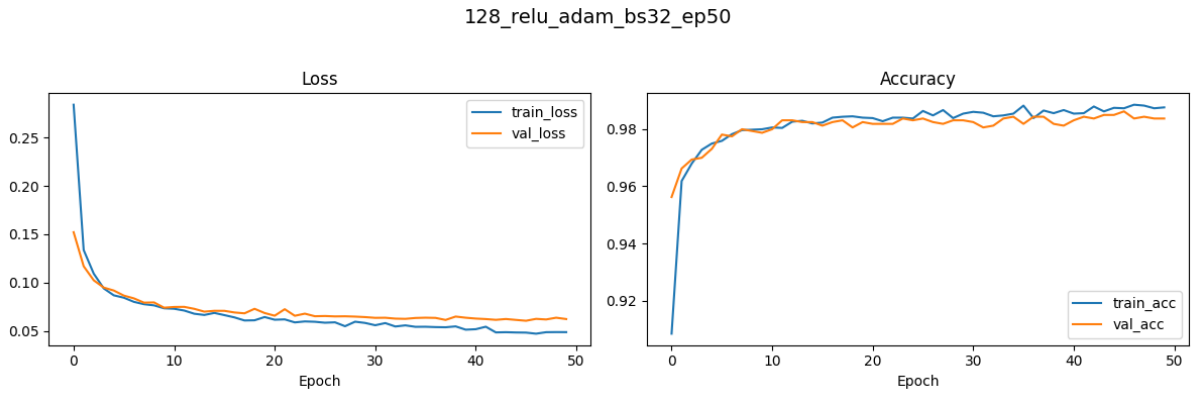
Tabela 11. Wyniki najlepszej konfiguracji według Accuracy (zaokrąglone)

Konfiguracja	Accuracy	Precision	Recall	F1	ROC-AUC
128_relu_adam_bs32_ep50	0.98625	0.98994	0.98252	0.98622	0.99538

1. Wyniki na zbiorze walidacyjnym (1600 próbek)

- Accuracy ~98.63 % - spośród 1 600 próbek tylko ~21 zostało sklasyfikowanych niewłaściwie.
- Precision ~0.99 dla obu klas - bardzo niski odsetek fałszywych pozytywów w predykcjach “Good” i fałszywych negatywów w “Bad”.
- Recall ~0.98-0.99 - model niemal nie pomija rzeczywistych “Good” i “Bad”.
- F1 ~0.983 - doskonała równowaga między precyzją i czułością.
- ROC-AUC ~0.9954 - niemal idealna separacja rozkładów prognoz dla obu klas.

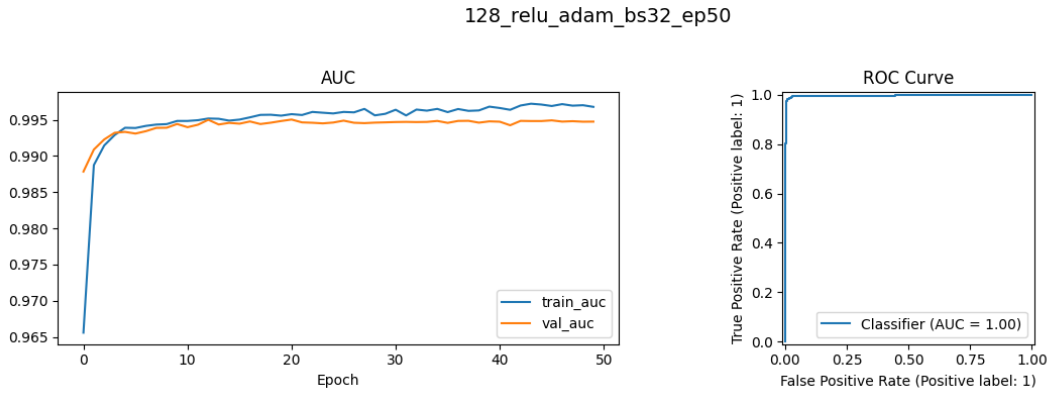
2. Krzywe Loss + Accuracy



Rysunek 24. Krzywe Loss i Accuracy

- Loss vs. val_loss - szybki spadek poniżej 0.06 przed 20. epoką i stabilizacja do końca.
- Accuracy vs. val_acc - osiągnięcie ~0.98 już koło 10. epoki, ostatecznie krążenie wokół 0.986.

3. Krzywe AUC + ROC



Rysunek 25. Krzywe AUC i ROC

- AUC vs. val_auc - wzrost z ~0.99 do > 0.995, niezłe dopasowanie linii trening/walidacja.
- Krzywa ROC przypina się przy lewym górnym rogu, potwierdzając praktycznie doskonałą separację klas.

Powyższa konfiguracja uzyskuje najlepszą dokładność spośród wszystkich testowanych MLP, przewyższając uproszczone klasyfikatory (kNN ~97.7 % accuracy). Wysokie wartości metryk precyzji, recall i AUC świadczą o jej stabilności i równomiernym rozkładzie błędów FP/FN.

5.4. Najlepsza konfiguracja według ROC-AUC

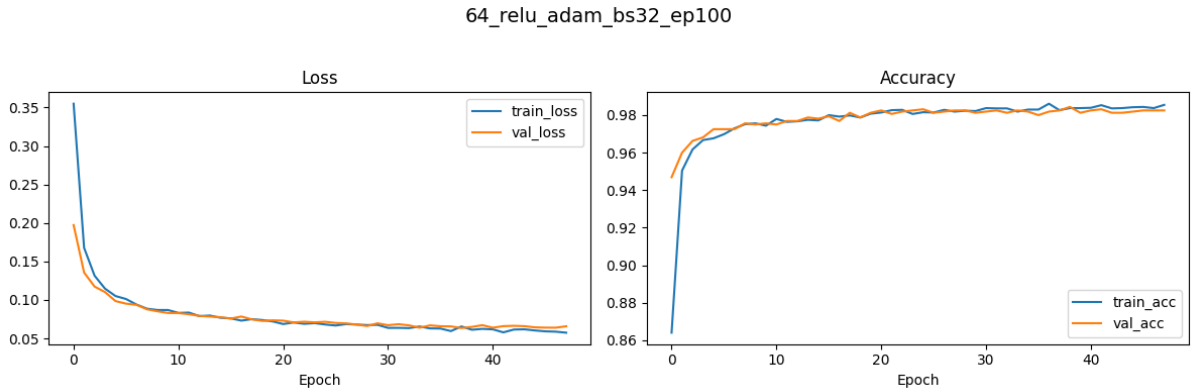
Tabela 12. Wyniki najlepszej konfiguracji według ROC-AUC (zaokrąglone)

Konfiguracja	Accuracy	Precision	Recall	F1	ROC-AUC
64_relu_adam_bs32_ep100	0.9825	0.98616	0.97878	0.98246	0.99562

1. Wyniki na zbiorze walidacyjnym (1600 próbek)

- Accuracy ~98.25 % - tylko ~28 błędnych klasyfikacji.
- Precision ~0.986 - bardzo niski odsetek fałszywych pozytywów w “Good”.
- Recall ~0.979 - niewiele rzeczywistych “Good” zostało pominiętych.
- F1 ~0.9825 - zachowuje dobry kompromis między precyzją i czułością.
- ROC-AUC ~0.9956 - najwyższa spośród testowanych, wskazuje na doskonałą jakość rozdzielenia progów.

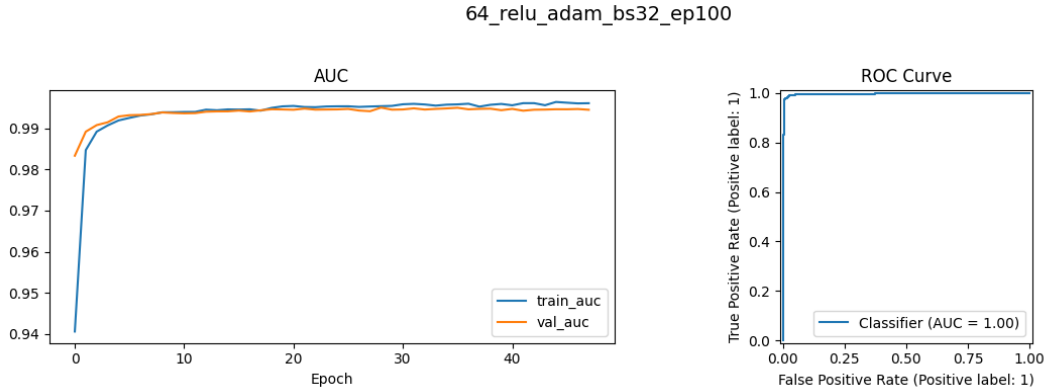
2. Krzywe Loss + Accuracy



Rysunek 26. Krzywe Loss i Accuracy

- Loss vs. val_loss - spadek z ~0.35 do ~0.06–0.07 w ciągu pierwszych 30 epok, następnie stabilizacja.
- Accuracy vs. val_acc - szybkie dojście do ~0.98 koło 15. epoki, potem łagodne wahania wokół tej wartości.

3. Krzywe AUC + ROC



Rysunek 27. Krzywe AUC i ROC

- AUC vs. val_auc - wzrost z ~0.94 do > 0.995, trening i walidacja niemal nakładają się.
- Krzywa ROC - linia krzywa przykleja się do lewego górnego rogu, co potwierdza bardzo wysoką separowalność klas.

Powyższa konfiguracja osiąga najwyższe ROC-AUC (0.9956) spośród wszystkich MLP, co oznacza, że nawet w granicach progów decyzyjnych klasa “Good” jest odseparowana od “Bad” niemal idealnie. Choć jej dokładność (98.25 %) jest nieco niższa niż w najlepszym pod kątem Accuracy układzie, minimalnie lepsze ROC-AUC czyni ją optymalnym wyborem, jeśli priorytetem jest maksymalizacja rozdzielczości rankingowej predykcji.

5.5. Podsumowanie sieci neuronowych

Spośród przebadanych MLP, konfiguracja **128_relu_adam_bs32_ep50** osiągnęła najwyższą **Accuracy (98.63 %)**, przy równie znakomitym **ROC-AUC (0.9954)**. Dzięki warstwie zawierającej 128 neuronów, model łapie głębsze zależności bez nadmiernego ryzyka przeuczenia (EarlyStopping), a umiarkowany Dropout (0.2) zapewnia stabilne uogólnianie. Jeśli natomiast priorytetem jest maksymalizacja zdolności rankingu predykcji (ROC-AUC), warto rozważyć wariant **64_relu_adam_bs32_ep100**, który osiągnął nieznacznie wyższe **AUC (0.9956)** przy nadal bardzo dobrej **Accuracy (98.25 %)**.

6. Bazowe klasyfikatory vs. MLP – podsumowanie

- Spośród prostych modeli kNN (k=7, p=3) osiągnął najlepsze wyniki (Accuracy ~97,7 %, ROC-AUC ~0,99).
- GaussianNB i warianty Bernoulli/Categorical dają tylko ~80–85 % Accuracy (AUC ~0,90), ale są bardzo szybkie i łatwe w interpretacji.
- DecisionTree (entropy, max_depth=None) to kompromis interpretowalności (~93,6 % Accuracy) i jakości.
- MLP (128 neurony, ReLU, Adam, bs=32, ep=50) poprawia Accuracy do ~98,6 % (AUC ~0,995), a wariant (64, ep=100) osiąga rekordowe AUC ~0,9956.

Jako finalny model należałoby wybrać MLP 128 (ep50) ze względu na najlepsze Accuracy i bardzo wysokie AUC, a na poziomie baseline warto zachować kNN jako punkt odniesienia do dalszych analiz.

7. Podsumowanie badań i kluczowe wnioski

1. Najlepsze proste modele

- kNN (k=7, Minkowski p=3) uzyskał 97,7 % accuracy i AUC ≈ 0,99 – prostota implementacji + doskonałe uogólnianie.
- Drzewo decyzyjne (entropy, max_depth=None) dało 93,6 % accuracy i AUC ~0,94, świetne dla interpretowalności, ale wymagałoby regularyzacji.
- Naive Bayes (~80–85 % accuracy, AUC ~0,90) potwierdził się jako szybki, lekki baseline, ale nie wykorzystuje pełni informacji ciągłych cech.

2. Głębokie sieci (MLP)

- MLP [128] → ReLU → [Dropout 0.2] → [1] (Adam, bs=32, ep=50) podwyższył accuracy do 98,6 % i AUC ~0,9954, co daje finalnie najlepszy kompromis.
- Dłuższy trening wariantem [64] (ep=100) minimalnie podwyższył AUC do 0,9956, lecz kosztem 0,4 punktu procentowego accuracy.

– MLP przewyższa proste klasyfikatory o ~1 punkt procentowy accuracy i o kilkanaście punktów AUC, potwierdzając wartość nieliniowości.

3. Reguły asocjacyjne

– Silne wzorce „high sweetness” + „high ripeness” (+ rozmiar medium/high) w ~90 % wskazują na banany dobre.
– Zaskoczenie - „high acidity” w połączeniu z dojrzałością i słodyczą wcale nie obniża jakości (lift > 1.8).

4. Ogólne wnioski

– Co działa: kNN jako wydajny baseline; MLP z jedną warstwą 128 neuronów dla najwyższej skuteczności; reguły asocjacyjne potwierdzające intuicję z EDA.
– Co nie działa: proste NB traci dużo informacji ciągłej.
– Interesujące obserwacje: mimo że wysoka kwasowość zwykle kojarzy się z gorszą jakością, w połączeniu z dojrzałością i słodyczą tworzy stabilny sygnał „Good”.