Politechnika Wrocławska

Politechnika Wrocławska
Wydział Mechaniczny

Zarządzanie i Inżynieria Produkcji (ZIP)
Organizacja Produkcji (OP)

Uczenie maszynowe jako narzędzie doskonalenia procesów wytwórczych

Machine Learning as a Tool for Manufacturing Process Improvement

Praca dyplomowa magisterska

Autor: Mateusz A. Tabor
Promotor: Dr inż. Kamil Musiał

Wrocław, 2026

Spis treści

1

Charakterystyka algorytmów uczenia maszynowego

1.1 Uczenie nadzorowane (Supervised Learning)

Uczenie nadzorowane to technika, w której algorytm otrzymuje gotowy zestaw danych z wcześniej określonymi etykietami i uczy się na ich podstawie predyktować etykiety dla nowych nieznanych danych. Dane uczące zawierają zmienne wejściowe oraz zmienną predykowaną (etykietę). Standardowy proces uczenia nadzorowanego obejmuje podział zbioru danych na zbiór treningowy, walidacyjny oraz testowy. Trening polega na dopasowaniu parametrów modelu do danych treningowych poprzez minimalizację funkcji straty, która mierzy różnicę między przewidywaniami modelu a rzeczywistymi etykietami. Po zakończeniu treningu danych model zostaje poddany ocenie na zbiorze walidacyjnym w celu doboru hiperparametrów oraz monitorowaniu uczenia maszynowego, aby zapobiec przeuczeniu modelu. Ostateczna ocena odbywa się na zbiorze testowym na danych nieznanych dla modelu i na podstawie tego modelu określana wartość dokładności, precyzji, recall, F1 (dla klasyfikacji) lub MSE, MAE (dla regresji)(Bishop, 2006; Hastie et al., 2009).

1.1.1 Regresja liniowa (Linear Regression)

Regresja liniowa to podstawowa technika statystyczna i uczenia maszynowego stosowana do modelowania zależności między zmienną zależną (predykowaną) a jedną lub więcej zmiennymi niezależnymi (cechami). Celem regresji liniowej jest znalezienie liniowej funkcji, która najlepiej opisuje związek między zmiennymi, umożliwiając przewidywanie wartości zmiennej zależnej na podstawie wartości zmiennych niezależnych.

Regresja liniowa prosta. W przypadku pojedynczej zmiennej niezależnej model można opisać równaniem:

$$y = \alpha + \beta x + \varepsilon, \tag{1.1}$$

gdzie y to zmienna zależna, x to zmienna niezależna, α to wyraz wolny, β to współczynnik nachylenia linii regresji, a ε to składnik losowy (błąd modelu).

Regresja wieloraka. Regresja wieloraka jest rozszerzeniem regresji prostej, poprzez używanie wielu zmiennych niezależnych. Zakłada się liniową zależność między zmienną zależną a kombinacją liniową zmiennych niezależnych:

$$y = \alpha + \sum_{i=1}^{p} \beta_i x_i + \varepsilon, \tag{1.2}$$

gdzie y to zmienna zależna, α to wyraz wolny, x_i to zmienne niezależne, β_i to współczynniki regresji określające wpływ danej zmiennej na zmienną zależną, a ε to składnik losowy (błąd modelu).

Celem algorytmu regresji (prostej i wielorakiej) jest znalezienie optymalnych wartości współczynników β_i, które minimalizują błąd predykcji. Współczynniki można dobrać za pomocą różnych metod, jednak najczęściej stosuje się metodę najmniejszych kwadratów (OLS — Ordinary Least Squares), która minimalizuje sumę kwadratów różnic między rzeczywistymi a przewidywanymi wartościami zmiennej zależnej.

Estymator OLS (macierzowy zapis).

$$\widehat{\boldsymbol{\beta}} = (X^\top X)^{-1} X^\top \mathbf{y} \tag{1.3}$$

Wzór (1.3) to macierzowy zapis estymatora OLS. Wyjaśnienie:

- X — macierz projektująca (design matrix) o wymiarach $n \times p$ (lub $n \times (p+1)$, jeśli dodano kolumnę jedynek dla wyrazu wolnego),
- $\mathbf{y}$ — wektor obserwacji o wymiarze $n \times 1$,
- $\widehat{\boldsymbol{\beta}}$ — wektor estymowanych współczynników o wymiarze $p \times 1$.

Aby wyrażenie było poprawne, macierz $X^\top X$ musi być odwracalna (brak doskonałej multikolinearności). Przy klasycznych założeniach (m.in. $E[\varepsilon] = 0$, $\mathrm{Var}(\varepsilon) = \sigma^2 I$) estymator jest nieobciążony, a jego wariancja wynosi $\mathrm{Var}(\widehat{\boldsymbol{\beta}}) = \sigma^2 (X^\top X)^{-1}$. Gdy $X^\top X$ jest źle uwarunkowana lub nieodwracalna, stosuje się regularyzację (np. Ridge) lub metody numeryczne (gradient descent, SVD)(James et al., 2013).

Inne metody estymacji współczynników regresji liniowej:

- Metoda gradientu prostego (Gradient Descent) — iteracyjna metoda optymalizacji minimalizująca funkcję straty poprzez aktualizację współczynników w kierunku przeciwnym do gradientu (Bishop, 2006; Murphy, 2012).
- Metoda najmniejszych modułów (Least Absolute Deviations) — minimalizuje sumę bezwzględnych różnic między rzeczywistymi a przewidywanymi wartościami, co czyni ją bardziej odporną na wartości odstające (Birkes and Dodge, 1993).
- Metoda Ridge Regression — wprowadza regularyzację L2, karę za duże wartości współczynników, co pomaga w radzeniu sobie z problemem multikolinearności i przeuczenia modelu (Hoerl and Kennard, 1970; Hastie et al., 2009).
- Metoda Lasso Regression — regularyzacja L1, która może prowadzić do zerowania niektórych współczynników, skutkując modelem o mniejszej liczbie cech (automatyczny wybór cech) (Tibshirani, 1996).
- Metoda Elastic Net — łączy regularyzację L1 i L2, co pozwala na lepsze dostosowanie modelu do danych (Zou and Hastie, 2005).
- Metoda SVD (Singular Value Decomposition) — rozkłada macierz projektującą na składniki, umożliwiając stabilne obliczenie współczynników regresji nawet w przypadku kolinearności cech (Golub and Loan, 1996; Press et al., 2007).
- Metoda Bayesian Regression — wykorzystuje podejście bayesowskie do estymacji współczynników, uwzględniając niepewność i priorytety w modelu (Gelman and Hill, 2006; Bishop, 2006).

- QR Decomposition — rozkłada macierz projektującą na iloczyn macierzy ortogonalnej i górnotrójkątnej, co umożliwia efektywne rozwiązanie układu równań regresji (Golub and Loan, 1996; Press et al., 2007).

1.1.2 Regresja logistyczna (Logistic Regression)

Regresja logistyczna to technika statystyczna i uczenia maszynowego stosowana do modelowania zależności między zmienną zależną a jedną lub więcej zmiennymi niezależnymi, gdy zmienna zależna przyjmuję wartości binarne (np. 0 lub 1, tak lub nie). Celem regresji logistycznej jest przewidywanie prawdopodobieństwa przynależności do jednej z dwóch klas na podstawie wartości zmiennych niezależnych.

Model regresji logistycznej. Model regresji logistycznej można zapisać jako:

$$P(Y = 1|X) = \sigma(\boldsymbol{\beta}^\top X) = \frac{1}{1 + e^{-\boldsymbol{\beta}^\top X}} \tag{1.4}$$

gdzie:

- $P(Y = 1|X)$ — prawdopodobieństwo, że zmienna zależna Y przyjmuje wartość 1, biorąc pod uwagę zmienne niezależne X,
- $\sigma(z)$ — funkcja sigmoidalna, która przekształca dowolną wartość rzeczywistą z w przedział (0, 1).

Estymacja parametrów. Parametry modelu $\boldsymbol{\beta}$ są estymowane za pomocą metody największej wiarygodności (Maximum Likelihood Estimation, MLE). Celem jest maksymalizacja funkcji wiarygodności:

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} P(Y_i|X_i; \boldsymbol{\beta}) \tag{1.5}$$

co jest równoważne minimalizacji funkcji straty:

$$J(\boldsymbol{\beta}) = -\sum_{i=1}^{n} [Y_i \log(P(Y_i|X_i; \boldsymbol{\beta})) + (1 - Y_i) \log(1 - P(Y_i|X_i; \boldsymbol{\beta}))] \tag{1.6}$$

Właściwości modelu. Model regresji logistycznej ma kilka istotnych właściwości:

- Wynikiem modelu jest prawdopodobieństwo, co czyni go odpowiednim narzędziem do klasyfikacji binarnej.
- Model jest odporny na wartości odstające, ponieważ wykorzystuje funkcję sigmoidalną.
- Można go łatwo rozszerzyć na problemy wieloklasowe (np. regresja wielomianowa).

1.1.3 k-Najbliższych Sąsiadów (k-Nearest Neighbors)

Algorytm k-Najbliższych Sąsiadów umieszcza dane wejściowe w przestrzeni wielowymiarowej i klasyfikuje je na podstawie etykiet najbliższych sąsiadów w tej przestrzeni. Przestrzeń jest definiowana przez cechy danych, zbiór danych posiadający x cech jest reprezentowany w x-wymiarowej przestrzeni. Algorytm klasyfikując dany obiekt oblicza odległości między nim a wszystkimi innymi obiektami w przestrzeni, a następnie wybiera k najblizszych sąsiadów. Wartość k jest ustalana ustalana przed rozpoczęciem działania algorytmu. Niska wartość parametru k jest bardziej podatna na szumy w danych, podczas gdy wysoka wartość k może prowadzić do nadmiernego uogólnienia modelu (Cover and Hart, 1967).

Algorytm k-najbliższych sąsiadów może wykorzystywać różne metryki do obliczania odległości m.in:

Metryka Euklidesowa: Najpowszechniejsza metryka używana do obliczania odległości między dwoma punktami w przestrzeni wielowymiarowej. Definiowana jest jako:

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{1.7}$$

gdzie p i q to dwa punkty w przestrzeni, a n to liczba wymiarów. Wyliczanie odległości metryką euklidesową polega na policzeniu różnicy ogległości w każdym wymiarze dla dwóch punktów, zsumowaniu kwadratów tych różnic, a następnie wyciągnięciu pierwiastka kwadratowego z tej sumy (Duda et al., 2001; Bishop, 2006).

Metryka Manhattan: Metryka Manhattan, nazywana również metryką taksówkową lub L1, mierzy odległość miedzy dwoma punktami jako sumę watości bezwzględnych z różnic współżędnych. Definiowana jest jako:

$$d(p, q) = \sum_{i=1}^{n} |p_i - q_i| \tag{1.8}$$

gdzie p i q to dwa punkty w przestrzeni, a n to liczba wymiarów. Obliczana jest jest różnica wartości p i q dla każdego wymiaru, a następnie sumowane są wartości bezwzględne tych różnic (Duda et al., 2001).

Metryka Kosinusowa: Metryka Kosinusowa mierzy wyznacza odległość między dwoma punktami na podstawie wyliczonego kąta między nimi. Definiowana jest jako:

$$d(p, q) = 1 - \frac{p \cdot q}{\|p\| \|q\|} \tag{1.9}$$

gdzie p i q to dwa punkty w przestrzeni, $p \cdot q$ to iloczyn skalarny wektorów p i q, a $\|p\|$ i $\|q\|$ to normy (długości) tych wektorów. Normy długości wektorów są obliczane jako pierwiastki kwadratowe z sumy kwadratów ich współrzędnych (Manning et al., 2008; Bishop, 2006).

Metryka Minkowskiego: Metryka Minkowskiego jest uogólnieniem metryk Euklidesowej i Manhattan. Umożliwa regulowanie sposobu obliczania odległości poprzez parametr p. Definiowana jest jako:

$$d(p,q) = \left(\sum_{i=1}^{n} |p_i - q_i|^p \right)^{\frac{1}{p}} \tag{1.10}$$

gdzie p i q to dwa punkty w przestrzeni, n to liczba wymiarów, a p to parametr regulujący sposób obliczania odległości. Dla $p = 1$ metryka Minkowskiego jest równoważna metryce Manhattan, a dla $p = 2$ jest równoważna metryce Euklidesowej (Hastie et al., 2009).

1.1.4 Drzewa decyzyjne (Decision Trees)

Drzewa decyzyjne to algorytm uczenia maszynowego, która służy do podejmowania decyzji na podstawie zestawu reguł, które są reprezentowane w formie drzewa. Drzewo decyzyjne składa się z korzenia, które jest cechą dzielącą dane na grupy, węzłów wewnętrznych, które reprezentują pytania dotyczące cech danych, oraz liści, które reprezentują ostateczne decyzje lub klasyfikacje. Algorytm budowy drzewa decyzyjnego polega na iteracyjnym dzieleniu danych na podzbiory na podstawie cech, które najlepiej rozdzielają dane według określonego kryterium. Drzewo decyzyjne jest budowane, aż wszystkie elementy w podzbiorze należą do tej samej klasy lub nie ma już cech do podziału, osiągnie maksymalną głębokość lub kiedy dalszy podział nie poprawia jakości klasyfikacji (Quinlan, 1986; Breiman et al., 1984).

Drzewo decyzyjne do wyboru najlepszego podziału danych może wykorzystywać różne kryteria, m.in:

Entropia: Entropia jest miarą niepewności lub nieuporządkowania w zbiorze danych. Entropia jest wykorzystywana do oceny jakości podziału danych na podstaiwe cechy. Definiowana jest jako:

$$H(S) = -\sum_{i=1}^{c} p_i \log_2(p_i) \tag{1.11}$$

gdzie S to zbiór danych, c to liczba klas, a p_i to proporcja elementów należących do klasy i w zbiorze S. Entropia obliczana jest jako suma iloczynów proporcji klas i logarytmów tych proporcji, a następnie mnożona przez -1 (Shannon, 1948; Quinlan, 1986).

Wskaźnik Gini (Gini Impurity): Wskaźnik Gini mierzy prawdopodobieństwo błędnej klasyfikacji losowo wybranego elementu, gdyby został on oznaczony losowo według rozkładu etykiet w węźle. Im niższy wskaźnik Gini, tym bardziej jednorodny jest węzeł. Wskaźnik Gini dla zbioru S jest definiowany jako:

$$\text{Gini}(S) = 1 - \sum_{i=1}^{c} p_i^2 \tag{1.12}$$

gdzie c to liczba klas, a p_i to proporcja przykładów w klasie i. Wskaźnik Gini jest używany w algorytmie CART (Classification and Regression Trees) ze względu na swoją prostotę obliczeniową i dobrą wydajność (Breiman et al., 1984).

Zysk informacji (Information Gain): Zysk informacji mierzy redukcję entropii osiągniętą przez podział zbioru danych według danego atrybutu. Jest to różnica między entropią zbioru nadrzędnego a ważoną sumą entropii zbiorów potomnych. Zysk informacji dla atrybutu A w zbiorze S definiuje się jako:

$$\mathrm{IG}(S, A) = H(S) - \sum_{v \in \mathrm{Values}(A)} \frac{|S_v|}{|S|} H(S_v) \tag{1.13}$$

gdzie Values(A) to zbiór wszystkich możliwych wartości atrybutu A, S_v to podzbiór S, dla którego atrybut A ma wartość v, a $H(S)$ to entropia zbioru. Algorytm ID3 wybiera atrybut o największym zysku informacji (Quinlan, 1986).

Współczynnik zysku (Gain Ratio) Współczynnik zysku jest modyfikacją zysku informacji, która koryguje tendencję do faworyzowania atrybutów o wielu wartościach. Normalizuje zysk informacji przez podzielenie go przez tzw. split information, która mierzy szerokość i jednolitość podziału. Współczynnik zysku dla atrybutu A w zbiorze S definiuje się jako:

$$\mathrm{GainRatio}(S, A) = \frac{\mathrm{IG}(S, A)}{\mathrm{SplitInfo}(S, A)} \tag{1.14}$$

gdzie $\mathrm{IG}(S, A)$ to zysk informacji dla atrybutu A, a $\mathrm{SplitInfo}(S, A)$ to informacja o podziale, definiowana jako:

$$\mathrm{SplitInfo}(S, A) = -\sum_{v \in \mathrm{Values}(A)} \frac{|S_v|}{|S|} \log_2 \left(\frac{|S_v|}{|S|} \right) \tag{1.15}$$

gdzie Values(A) to zbiór wszystkich możliwych wartości atrybutu A, S_v to podzbiór S zawierający elementy, dla których atrybut A ma wartość v, $|S_v|$ to liczba elementów w podzbiorze S_v, a $|S|$ to całkowita liczba elementów w zbiorze S. Informacja o podziale mierzy, jak bardzo atrybut dzieli dane. Im bardziej równomierny podział, tym wyższa wartość SplitInfo, co zmniejsza współczynnik zysku i zapobiega faworyzowaniu atrybutów o wielu unikalnych wartościach. Współczynnik zysku jest używany w algorytmie C4.5 jako ulepszona wersja ID3 (Quinlan, 1993).

Redukcja wariancji (Variance Reduction) Redukcja wariancji jest kryterium stosowanym w drzewach regresyjnych, gdzie celem jest przewidywanie wartości ciągłych, a nie kategorii. Kryterium to wybiera podział, który maksymalnie redukuje wariancję wartości docelowych w węzłach potomnych. Redukcja wariancji dla podziału zbioru S na podzbiory S_{left} i S_{right} definiuje się jako:

$$\mathrm{VarReduction}(S) = \mathrm{Var}(S) - \left(\frac{|S_{\text{left}}|}{|S|} \mathrm{Var}(S_{\text{left}}) + \frac{|S_{\text{right}}|}{|S|} \mathrm{Var}(S_{\text{right}}) \right) \tag{1.16}$$

gdzie $\mathrm{Var}(S)$ oznacza wariancję wartości docelowych w zbiorze S. Algorytm CART dla regresji wykorzystuje to kryterium do budowy drzew regresyjnych (Breiman et al., 1984).

1.1.5 Las losowy (Random Forest)

Las losowy to algorytm uczenia maszynowego, który łączy wiele drzew decyzyjnych w celu poprawy dokładności przewidywań i redukcji przeuczenia. Algorytm ten działa

na zasadzie tworzenia wielu niezależnych drzew decyzyjnych, z których każde jest trenowane na losowym podzbiorze danych i losowym podzbiorze cech. Ostateczna predykcja lasu losowego jest uzyskiwana przez agregację wyników wszystkich drzew. Dla klasyfikacji stosuje się głosowanie większościowe, a dla regresji średnią arytmetyczną (Breiman, 2001).

Las losowy uczy się poprzez losowanie i budowanie wielu drzew decyzyjnych. Każde drzewo dostaje losowy podzbiór danych treningowych oraz losowy podzbiór cech do rozważenia przy każdym podziale węzła. Ten proces losowania danych i cech wprowadza różnorodność między drzewami, co przekłada się na lepszą ogólną wydajność modelu. Poszczególne drzewa dzielone są na podstawie kryteriów omówionych w sekcji dotyczącej drzew decyzyjnych, takich jak entropia, wskaźnik Gini czy zysk informacji (Breiman, 2001).

Dodatkowo przy budownie lasu losowego dobiera się parametry:

Tabela 1.1: Podstawowe hiperparametry lasu losowego

Parametr	**Opis**
Liczba drzew (B)	Liczba drzew decyzyjnych w lesie. Większa liczba zazwyczaj poprawia wydajność, ale zwiększa czas obliczeń.
Maksymalna głębokość	Maksymalna głębokość każdego drzewa. Ograniczenie głębokości może zapobiec przeuczeniu.
Liczba cech (m)	Liczba losowo wybranych cech rozważanych przy każdym podziale. Typowo $m = \sqrt{p}$ dla klasyfikacji lub $m = p/3$ dla regresji. (p to liczba wszystkich cech)
Minimalna liczba próbek	Minimalna liczba próbek wymagana do podziału węzła wewnętrznego lub do utworzenia liścia.
Bootstrap	Czy stosować bootstrap sampling (losowanie ze zwracaniem) do tworzenia podzbiorów treningowych.

Strojenie hiperparametrów lasu losowego, jest zadaniem człowieka, ale najczęściej wykorzystuje się metody automatyczne do testowania.

Grid Search (przeszukiwanie siatki) Grid Search polega na przeszukiwaniu wszystkich możliwych kombinacji hiperparametrów z wcześniej zdefiniowanej siatki wartości. Dla każdej kombinacji algorytm trenuje model i ocenia jego wydajność za pomocą walidacji krzyżowej. Następnie wybiera zestaw parametrów dający najlepsze wyniki według ustalonej metryki (np. accuracy, F1).

Randomized Search (przeszukiwanie losowe) Randomized Search jest wariantem Grid Search, który zamiast sprawdzać wszystkie kombinacje, losowo próbkuje określoną liczbę zestawów hiperparametrów z zadanych rozkładów. Użytkownik definiuje budżet iteracji (liczbę kombinacji do przetestowania).

Optuna Optuna to framework do optymalizacji hiperparametrów wykorzystujący zaawansowane algorytmy, takie jak Tree-structured Parzen Estimator (TPE). W przeciwieństwie do metod grid i random search, Optuna adaptacyjnie wybiera kolejne kom-

binacje hiperparametrów na podstawie wyników wcześniejszych prób — uczy się, które obszary przestrzeni są obiecujące i koncentruje tam przeszukiwanie. Framework oferuje elastyczność w definiowaniu przestrzeni parametrów, wbudowane wizualizacje oraz możliwość przycinania nieobiecujących prób (pruning). Optuna wykazuje szybszą zbieżność niż metody losowe, ale wymaga instalacji dodatkowej biblioteki (Akiba et al., 2019).

Bayesian Optimization (optymalizacja bayesowska) Bayesian Optimization buduje probabilistyczny model zastępczy (surrogate model), zazwyczaj Gaussian Process, który aproksymuje funkcję celu (np. dokładność modelu jako funkcję hiperparametrów). Na podstawie tego modelu algorytm wybiera kolejne punkty do próbkowania za pomocą funkcji akwizycji (acquisition function), takiej jak Expected Improvement (EI), która balansuje eksplorację nowych obszarów przestrzeni i eksploatację obiecujących regionów. Metoda jest szczególnie efektywna, gdy ewaluacja funkcji celu jest kosztowna (np. trening dużych modeli), ponieważ wymaga stosunkowo małej liczby iteracji. Wadą jest złożoność implementacji oraz spowolnienie dla bardzo dużych przestrzeni hiperparametrów (Snoek et al., 2012).

AutoML (Automated Machine Learning) AutoML automatyzuje cały proces budowy modelu uczenia maszynowego, w tym wybór algorytmu, inżynierię cech, dobór hiperparametrów oraz tworzenie modeli ensemble. Narzędzia takie jak Auto-sklearn, TPOT czy H2O AutoML wykorzystują kombinację technik optymalizacji (bayesian optimization, evolutionary algorithms) oraz meta-learning (wiedza z wcześniejszych eksperymentów na podobnych zbiorach). Główną zaletą AutoML jest minimalna interwencja użytkownika — wystarczy dostarczyć dane i określić metrykę, a system samodzielnie buduje i optymalizuje model. Wadą jest ograniczona kontrola nad procesem, długie czasy obliczeń oraz potencjalne tworzenie skomplikowanych, trudnych do interpretacji modeli typu „czarna skrzynka"(Feurer et al., 2015; Olson et al., 2016).

Bibliografia

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2623–2631, 2019. doi: 10.1145/3292500.3330701.

David Birkes and Yadolah Dodge. *Alternative Methods of Regression*. Wiley, New York, 1993.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324.

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, CA, 1984.

Thomas M. Cover and Peter E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967. doi: 10.1109/TIT.1967.1053964.

Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley, New York, 2nd edition, 2001.

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Tobias Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Andrew Gelman and Jennifer Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, Cambridge, 2006.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 3rd edition, 1996.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2nd edition, 2009.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. doi: 10.1080/00401706.1970.10488634.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer, New York, 1st edition, 2013.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, 2008.

Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, Cambridge, MA, 2012.

Randal S. Olson, Nathan Bartley, Ryan J. Urbanowicz, and Jason H. Moore. Evaluation of a tree-based pipeline optimization tool for automating data science. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 485–492, 2016. doi: 10.1145/2908812.2908918.

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge, 3rd edition, 2007.

J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986. doi: 10.1007/BF00116251.

J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 25, 2012.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. doi: 10.1111/j.2517-6161.1996.tb02080.x.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005. doi: 10.1111/j.1467-9868.2005.00503.x.