

1. Why Machine Learning Strategy?

- **English:** Machine learning underpins applications like web search, spam filters, and recommendation systems. Developing a strategy is essential for selecting effective approaches. This section introduces the idea of learning clues from failed implementations to save time.
 - **Türkçe:** Makine öğrenimi, web arama, spam filtreleme ve öneri sistemleri gibi uygulamaların temelidir. Etkili yaklaşımları seçmek için bir strateji geliştirmek önemlidir. Bu bölüm, başarısız uygulamalardan ipuçları öğrenerek zaman kazanma fikrini tanıtır.
-

2. How to Use This Book to Help Your Team?

- **English:** This book aims to help readers set technical directions for ML projects and effectively communicate them to their teams through short, actionable chapters.
 - **Türkçe:** Bu kitap, okuyucuların makine öğrenimi projelerinde teknik yön belirlemesine ve bunları ekiplerine kısa ve uygulanabilir bölümlerle etkili bir şekilde aktarmasına yardımcı olmayı amaçlamaktadır.
-

3. Prerequisites and Notation

- **English:** A basic understanding of supervised learning, such as linear regression or neural networks, is required. Those with Andrew Ng's Coursera ML course will find this content easier to follow.
 - **Türkçe:** Lineer regresyon veya sinir ağları gibi gözetimli öğrenim hakkında temel bilgi gereklidir. Andrew Ng'in Coursera ML kursunu alanlar bu içeriği daha kolay takip edebilir.
-

4. Scale Drives Machine Learning Progress

- **English:** Availability of data and computational power has been key to deep learning's progress. Bigger datasets and larger neural networks often lead to better results.
 - **Türkçe:** Veri ve hesaplama gücünün mevcudiyeti, derin öğrenmedeki ilerlemenin anahtarıdır. Daha büyük veri setleri ve daha büyük sinir ağları genellikle daha iyi sonuçlara yol açar.
-

5. Development and Test Sets

- **English:** Properly splitting datasets into training, dev (development), and test sets ensures ML systems generalize well to real-world data. Dev/test sets should match the target distribution.

- **Türkçe:** Veri setlerini eğitim, geliştirme (dev) ve test olarak doğru bir şekilde ayırmak, ML sistemlerinin gerçek dünya verilerine iyi genelleme yapmasını sağlar. Geliştirme/test setleri hedef dağılımla eşleşmelidir.
-

6. Dev and Test Sets from the Same Distribution

- **English:** Mismatched distributions between dev and test sets can lead to wasted effort. Ensure they come from the same source to avoid unnecessary errors.
 - **Türkçe:** Geliştirme ve test setleri arasındaki uyumsuz dağılımlar, boşa harcanan çabalara yol açabilir. Gereksiz hatalardan kaçınmak için aynı kaynaktan gelmelerini sağlayın.
-

7. How Large Should Dev/Test Sets Be?

- **English:** Dev sets should be large enough to detect meaningful differences between models, typically 1,000–10,000 examples.
 - **Türkçe:** Geliştirme setleri, modeller arasındaki anlamlı farklılıkları tespit edebilecek kadar büyük olmalıdır; genellikle 1.000-10.000 örnek yeterlidir.
-

8. Single-Number Evaluation Metric

- **English:** Use a single evaluation metric, like accuracy or F1-score, to easily compare model performance.
 - **Türkçe:** Model performansını kolayca karşılaştırmak için doğruluk veya F1 skoru gibi tek bir değerlendirme metriği kullanın.
-

9. Optimizing and Satisficing Metrics

- **English:** Combine metrics effectively, using constraints on secondary metrics (e.g., running time) while optimizing the primary metric (e.g., accuracy).
 - **Türkçe:** Metotları etkili bir şekilde birleştirin; birincil metriği (ör. doğruluk) optimize ederken ikincil metriklerde (ör. çalışma süresi) kısıtlamalar kullanın.
-

Devam eden bölümler için aynı detaylı formatı hazırlıyorum. Her bir başlık tamamlandığında, bölümleri parça parça paylaşacağım. Bir sonraki aşamaya geçiyorum.

Devam Eden Özet: Machine Learning Yearning

10. Having a Dev Set and Metric Speeds Up Iterations

- **English:** A well-defined dev set and evaluation metric accelerate iteration cycles, helping teams quickly refine models based on performance.

- **Türkçe:** İyi tanımlanmış bir geliştirme seti ve değerlendirme metriği, iterasyon döngülerini hızlandırır ve ekiplerin performansla dayalı modelleri hızlı bir şekilde iyileştirmesine yardımcı olur.
-

11. When to Change Dev/Test Sets and Metrics?

- **English:** Change dev/test sets or metrics if they no longer align with the actual distribution or if they fail to reflect the application's needs.
 - **Türkçe:** Geliştirme/test setlerini veya metrikleri, gerçek dağılımla artık uyumlu değilse ya da uygulamanın ihtiyaçlarını yansıtmıyorsa değiştirin.
-

12. Setting Up Development and Test Sets: Key Takeaways

- **English:** Use dev/test sets that mirror expected future data distributions and ensure the dev set is large enough to detect meaningful improvements.
 - **Türkçe:** Beklenen gelecekteki veri dağılımlarını yansıtan geliştirme/test setleri kullanın ve geliştirme setinin anlamlı iyileştirmeleri tespit edecek kadar büyük olduğundan emin olun.
-

13. Build Your First System Quickly, Then Iterate

- **English:** Focus on building a basic system rapidly to gain insights from errors, which will guide iterative improvements.
 - **Türkçe:** Hatalardan elde edilen bilgilerle yönlendirilecek yinelemeli iyileştirmeler için temel bir sistemi hızla inşa etmeye odaklanın.
-

14. Error Analysis: Evaluating Misclassified Examples

- **English:** Review misclassified dev set examples to identify error categories and prioritize improvement efforts.
 - **Türkçe:** Geliştirme setinde yanlış sınıflandırılan örnekleri gözden geçirerek hata kategorilerini belirleyin ve iyileştirme çabalarına öncelik verin.
-

15. Evaluating Multiple Ideas in Parallel

- **English:** Analyze errors using spreadsheets to categorize misclassifications and assess the impact of various solutions simultaneously.
 - **Türkçe:** Yanlış sınıflandırmaları kategorize etmek ve çeşitli çözümlerin etkisini aynı anda değerlendirmek için hata analizinde elektronik tablolar kullanın.
-

16. Cleaning Up Mislabeled Examples

- **English:** Fix mislabeled examples in dev/test sets only if they significantly affect the system's ability to compare algorithms.
 - **Türkçe:** Geliştirme/test setlerindeki yanlış etiketlenmiş örnekleri yalnızca algoritmaları karşılaştırma yeteneğini önemli ölçüde etkiliyorsa düzeltin.
-

17. Splitting Large Dev Sets into Subsets

- **English:** Divide large dev sets into "Eyeball" (manually analyzed) and "Blackbox" (automatically evaluated) subsets to manage bias.
 - **Türkçe:** Büyük geliştirme setlerini "Eyeball" (manuel analiz edilen) ve "Blackbox" (otomatik değerlendirilen) alt setlere bölerek yanlılığı yönetin.
-

18. Size of Eyeball and Blackbox Dev Sets

- **English:** Ensure the Eyeball dev set has enough misclassifications to analyze (~100 examples), while the Blackbox dev set should be 1,000–10,000 examples.
 - **Türkçe:** Eyeball geliştirme setinde analiz için yeterli yanlış sınıflandırma (~100 örnek) olduğundan emin olun; Blackbox geliştirme seti ise 1.000-10.000 örnek olmalıdır.
-

19. Basic Error Analysis: Key Takeaways

- **English:** Start projects quickly, perform error analysis early, and iterate based on major error categories identified.
 - **Türkçe:** Projelere hızlı başlayın, erken hata analizi yapın ve belirlenen temel hata kategorilerine dayalı olarak yinelemeler gerçekleştirin.
-

20. Bias and Variance: The Two Big Sources of Error

- **English:** Bias refers to errors on the training set, while variance is the gap between training and dev/test set performance. Analyzing these can guide model improvements.
 - **Türkçe:** Bias, eğitim setindeki hatalara, variance ise eğitim ile geliştirme/test seti performansı arasındaki farka işaret eder. Bunları analiz etmek, model iyileştirmelerine rehberlik edebilir.
-

21. Examples of Bias and Variance

- **English:** High bias models underfit the training data, while high variance models overfit. Both issues need distinct approaches for resolution.
- **Türkçe:** Yüksek bias modeller eğitim verilerini yeterince öğrenemez (underfitting), yüksek variance modeller ise aşırı öğrenme (overfitting) yapar. Her iki sorun da farklı çözüm yaklaşımları gerektirir.

22. Comparing to the Optimal Error Rate

- **English:** Compare your model's performance to an optimal baseline, often based on human-level performance, to identify avoidable bias and variance.
- **Türkçe:** Modelinizin performansını, genellikle insan düzeyinde performansa dayalı olan, optimal bir referansla karşılaştırarak önlenebilir bias ve variance'ı belirleyin.

23. Addressing Bias and Variance

- **English:** To address bias, increase model size or add features. To address variance, add training data or use regularization techniques like L2 or dropout.
- **Türkçe:** Bias sorununu çözmek için model boyutunu artırın veya özellik ekleyin. Variance sorununu çözmek için eğitim verisi ekleyin veya L2 ya da dropout gibi düzenleme tekniklerini kullanın.

24. Bias vs. Variance Tradeoff

- **English:** In traditional ML, improving one often worsens the other. However, with modern deep learning, techniques like adding data or tuning regularization can address both.
- **Türkçe:** Geleneksel makine öğreniminde birini iyileştirmek genellikle diğerini kötüleştirir. Ancak modern derin öğrenimde, veri ekleme veya düzenleme ayarı gibi tekniklerle her ikisini de ele almak mümkündür.

25. Techniques for Reducing Avoidable Bias

- **English:** Use larger models, add relevant features, or tune architectures to reduce avoidable bias. Adding more training data typically won't help bias issues.
- **Türkçe:** Daha büyük modeller kullanın, ilgili özellikler ekleyin veya mimarileri ayarlayın. Bias sorunları için eğitim verisi eklemek genellikle yardımcı olmaz.

26. Error Analysis on the Training Set

- **English:** If the model has high bias, conduct error analysis on the training set to identify systematic issues, like specific data categories causing errors.
- **Türkçe:** Modelde yüksek bias varsa, eğitim seti üzerinde hata analizi yaparak hatalara neden olan sistematik sorunları belirleyin.

27. Techniques for Reducing Variance

- **English:** Adding training data, regularization, or early stopping helps reduce variance. Feature selection and decreasing model size may also work but require caution.
 - **Türkçe:** Eğitim verisi eklemek, düzenleme kullanmak veya erken durdurma yapmak variance'ı azaltır. Özellik seçimi ve model boyutunu küçültme de işe yarayabilir, ancak dikkatli uygulanmalıdır.
-

28. Diagnosing Bias and Variance with Learning Curves

- **English:** Learning curves (plots of error vs. training set size) help diagnose whether a model has high bias, high variance, or both.
- **Türkçe:** Öğrenme eğrileri (hata ile eğitim seti boyutu arasındaki grafikleri) bir modelin yüksek bias, yüksek variance veya her ikisine sahip olup olmadığını teşhis etmeye yardımcı olur.

29. Plotting Training Error

- **English:** Plot training error and dev set error against the size of the training set to better understand the learning curve and identify bias or variance issues.
 - **Türkçe:** Eğitim hatası ve geliştirme seti hatasını, eğitim seti boyutuna karşı grafikleyerek öğrenme eğrisini daha iyi anlayın ve bias ya da variance sorunlarını tespit edin.
-

30. Interpreting Learning Curves: High Bias

- **English:** If the training error is high and doesn't improve with more data, the model likely has high bias, indicating underfitting.
 - **Türkçe:** Eğitim hatası yüksekse ve daha fazla veriyle düzelmiyorsa, model muhtemelen yüksek bias'a sahiptir ve bu, yetersiz öğrenme (underfitting) anlamına gelir.
-

31. Interpreting Learning Curves: Other Cases

- **English:** Low training error but high dev set error suggests high variance. A flat dev set error with increasing data might point to dataset limitations.
 - **Türkçe:** Düşük eğitim hatası ancak yüksek geliştirme seti hatası, yüksek variance'a işaret eder. Artan veriyle birlikte sabit kalan geliştirme seti hatası, veri seti sınırlamalarına işaret edebilir.
-

32. Why Compare to Human-Level Performance?

- **English:** Human-level performance provides a benchmark to determine optimal error rates and prioritize bias or variance improvements.
 - **Türkçe:** İnsan düzeyi performans, optimal hata oranlarını belirlemek ve bias ya da variance iyileştirmelerine öncelik vermek için bir kıyas noktası sağlar.
-

33. How to Define Human-Level Performance?

- **English:** Use domain experts to label data and estimate how well humans perform on your specific task, setting a realistic baseline.
 - **Türkçe:** Veri etiketlemek için alan uzmanlarını kullanın ve insanların belirli bir görevde ne kadar başarılı olduğunu tahmin ederek gerçekçi bir referans belirleyin.
-

34. Surpassing Human-Level Performance

- **English:** ML systems often surpass humans in tasks with massive datasets or tasks humans find difficult, such as complex predictions or real-time processing.
 - **Türkçe:** ML sistemleri genellikle büyük veri setlerinde veya insanların zorlandığı karmaşık tahminler veya gerçek zamanlı işlemler gibi görevlerde insanları geride bırakır.
-

35. When to Train and Test on Different Distributions

- **English:** Train and test on different distributions only when it reflects real-world deployment scenarios, but this introduces data mismatch risks.
 - **Türkçe:** Eğitim ve testi, yalnızca gerçek dünya uygulamalarını yansıtıyorsa farklı dağıtımlarda gerçekleştirin, ancak bu durum veri uyumsuzluğu risklerini beraberinde getirir.
-

36. Addressing Data Mismatch Errors

- **English:** Use domain adaptation techniques, reweighting data, or creating synthetic datasets to bridge gaps between training and test distributions.
 - **Türkçe:** Eğitim ve test dağıtımları arasındaki farkları gidermek için alan uyarlama tekniklerini, veri ağırlıklandırmayı veya sentetik veri setlerini kullanın.
-

37. Artificial Data Synthesis

- **English:** Artificial data can augment training datasets, especially when real-world data is scarce or expensive to collect, but requires careful validation.
 - **Türkçe:** Gerçek dünya verilerinin az olduğu veya toplanmasının pahalı olduğu durumlarda, sentetik veriler eğitim veri setlerini artırabilir, ancak dikkatli doğrulama gerektirir.
-

38. The Rise of End-to-End Learning

- **English:** End-to-end learning simplifies pipelines by training all components simultaneously but can demand large datasets and computational power.
- **Türkçe:** Uçtan uca öğrenme, tüm bileşenleri aynı anda eğiterek işlem hatlarını basitleştirir, ancak büyük veri setleri ve hesaplama gücü gerektirebilir.

39. Error Analysis by Parts

- **English:** Break down errors into pipeline stages to locate the most impactful areas for improvement. Use detailed attribution to refine specific components.
- **Türkçe:** Hataları işlem hattı aşamalarına bölerek iyileştirme için en etkili alanları bulun. Belirli bileşenleri iyileştirmek için detaylı atıflar kullanın.

40. Spotting Flawed ML Pipelines

- **English:** A flawed ML pipeline often lacks data alignment, appropriate metrics, or robust preprocessing. Regular pipeline reviews can mitigate these risks.
- **Türkçe:** Kusurlu bir ML işlem hattı genellikle veri hizalamasından, uygun metriklerden veya sağlam ön işleme yöntemlerinden yoksundur. Düzenli incelemeler bu riskleri azaltabilir.

41. Identifying Bias, Variance, and Data Mismatch Errors

- **English:** Distinguish between bias, variance, and data mismatch errors by analyzing performance metrics across training, dev, and test sets.
- **Türkçe:** Eğitim, geliştirme ve test setleri arasındaki performans metriklerini analiz ederek bias, variance ve veri uyumsuzluk hatalarını ayırt edin.

42. Addressing Data Mismatch

- **English:** Data mismatch can be mitigated through techniques like reweighting training data, collecting more representative samples, or domain adaptation.
- **Türkçe:** Veri uyumsuzluğu, eğitim verilerini yeniden ağırlıklandırma, daha temsilci örnekler toplama veya alan uyarlama teknikleriyle giderilebilir.

43. Artificial Data Synthesis

- **English:** Synthetic data can fill gaps in datasets but must closely mimic real-world conditions to avoid introducing biases.
- **Türkçe:** Sentetik veriler, veri setlerindeki boşlukları doldurabilir ancak yanlılıkları önlemek için gerçek dünya koşullarını yakından taklit etmelidir.

44. The Optimization Verification Test

- **English:** This test checks whether optimization algorithms are implemented correctly, helping identify if issues lie in the training process.
- **Türkçe:** Bu test, optimizasyon algoritmalarının doğru uygulanıp uygulanmadığını kontrol ederek sorunların eğitim sürecinde mi olduğunu belirler.

45. General Form of Optimization Verification Test

- **English:** The test involves verifying gradients and loss functions step by step to ensure calculations align with expected outcomes.
- **Türkçe:** Test, hesaplamaların beklenen sonuçlarla uyumlu olduğunu doğrulamak için gradyanlar ve kayıp fonksiyonlarının adım adım kontrol edilmesini içerir.

46. Reinforcement Learning Example

- **English:** Reinforcement learning highlights challenges like sparse rewards and large state spaces, which require careful algorithm tuning.
- **Türkçe:** Pekiştirmeli öğrenme, seyrek ödüller ve geniş durum alanları gibi zorlukları vurgular ve dikkatli algoritma ayarları gerektirir.

47. The Rise of End-to-End Learning

- **English:** End-to-end learning is particularly powerful for tasks like speech recognition or machine translation, though it often requires substantial data and computation.
- **Türkçe:** Uçtan uca öğrenme, konuşma tanıma veya makine çevirisi gibi görevler için özellikle güçlüdür, ancak genellikle büyük miktarda veri ve hesaplama gerektirir.

48. Pros and Cons of End-to-End Learning

- **English:** Pros: Simplifies pipelines and improves accuracy. Cons: Requires more data, limits interpretability, and can be less robust to small changes.
- **Türkçe:** Avantajları: İşlem hatlarını basitleştirir ve doğruluğu artırır. Dezavantajları: Daha fazla veri gerektirir, yorumlanabilirliği sınırlar ve küçük değişikliklere karşı daha az dayanıklı olabilir.

49. Choosing Pipeline Components: Data Availability

- **English:** Select pipeline components based on the availability and quality of data; weaker components can hinder overall performance.
- **Türkçe:** İşlem hattı bileşenlerini veri mevcudiyetine ve kalitesine göre seçin; zayıf bileşenler genel performansı engelleyebilir.

50. Choosing Pipeline Components: Task Simplicity

- **English:** Favor simpler pipelines when possible; they are easier to debug and often require less data for comparable performance.

- **Türkçe:** Mümkün olduğunda daha basit işlem hatlarını tercih edin; bunlar hata ayıklamayı kolaylaştırır ve genellikle benzer performans için daha az veri gerektirir.
-

51. Directly Learning Rich Outputs

- **English:** ML systems increasingly learn complex outputs directly (e.g., entire sentences in translation), avoiding intermediate representations.
 - **Türkçe:** ML sistemleri giderek karmaşık çıktıları (ör. çeviride tam cümleler) doğrudan öğrenerek ara temsillere ihtiyaç duymuyor.
-

52. Error Analysis by Parts

- **English:** Attribute errors to specific pipeline components to understand their impact and prioritize improvements in critical areas.
 - **Türkçe:** Hataları belirli işlem hattı bileşenlerine atfederek bunların etkisini anlayın ve kritik alanlardaki iyileştirmelere öncelik verin.
-

53. Building a Superhero Team

- **English:** Encourage team members to learn and apply these principles to create a collaborative and effective ML development culture.
- **Türkçe:** Ekip üyelerini bu ilkeleri öğrenmeye ve uygulamaya teşvik ederek işbirlikçi ve etkili bir ML geliştirme kültürü oluşturun.