

Yapay Zekada Başarının Gizli Formülü: Algoritmaların Ötesinde Bilgi Keşfi ve Mülakatlarda Öne Çıkarıcı 100+ Soru

Giriş

Yapay zeka projelerindeki potansiyel çok yüksek olsa da pratikteki başarı oranları şaşırtıcı şekilde düşüktür. Gartner'a göre büyük veri projelerinin %85'i başarısızlıkla sonuçlanıyor ¹ ; RAND'ın yakın tarihli bir raporu ise yapay zeka projelerinin %80'inden fazlasının hedefe ulaşamadığını gösteriyor ² . Bu istatistikler, başarılı bir yapay zeka uygulamasının yalnızca iyi bir algorithmadan değil, esasen sağlam bir veri altyapısı ve yönetiminden geçtiğini ortaya koyuyor. Bu yazıda, başarının formülünde "algoritmaların ötesi"nde ne olduğunu; yani veri odaklı bilgi keşfi ve güçlü veri yönetişiminin projeleri nasıl başarıya taşıyabileceğini stratejik bir yaklaşımla ele alacağız. Ayrıca makine öğrenmesi ve veri bilimi mülakatlarına hazırlananlar için **100+ mülakat sorusuyla** teorik bilgilerinizi test edecek örnek sorular sunacağız.

Ana Kavram: Veri Odaklı Bilgi Keşfi ve Yönetimi

Veri Odaklı Bilgi Keşfi (Data-Driven Knowledge Discovery), ham verilerden anlamlı içgörüler çıkarma sürecidir. Veri madenciliği, istatistiksel analiz ve makine öğrenmesi yöntemleriyle veri içindeki gizli düzenler, ilişkiler ve öngörüler ortaya çıkarılır. Bu süreç, *ihtiyaç duyulan iş problemini anlamak* ile başlar, doğru verileri toplamak ve işlemek, sonra bu verilerle modeller kurmak, sonuçları yorumlamak ve sonuca göre strateji belirlemekle devam eder. Başka bir ifadeyle, veri odaklı bilgi keşfi projeyi bir **soru-cevap süreci** olarak ele alır: "Hangi veriler hangi bilgiyi verir ve bu bilgi nasıl iş değeri yaratır?"

Veri Yönetişimi (Data Governance) ise bu sürecin ayrılmaz bir parçasıdır. IBM'e göre veri yönetişimi, kurumsal verilerin **kalitesini, güvenliğini ve erişilebilirliğini** odak alan bir yönetim disiplini ³ . Veriyi toplarken, işlerken, depolarken ve paylaşırken uyulacak politika, standart ve süreçleri belirler. Örneğin bir kurumun veri yönetim stratejisi; veri sahipliğini netleştirir, veri kalitesi kriterleri koyar, gizlilik ve uyumluluk kurallarını uygular. Başka bir deyişle, veri yönetişimi hava trafik kontrolü gibidir: doğru veri akışı güvenli kanallardan ilgili yerlere ulaşır ³ . Yönetişim yoksa, "veri kaosu" eninde sonunda modelleri çarpıklaştırır.

Yapay zeka projeleri genellikle *yaşam döngüsü* (life cycle) denen birden fazla aşamadan geçer. Örneğin CRISP-DM çerçevesinde şu adımlar vardır: **İş Anlayışı, Veri Anlayışı, Veri Hazırlama (Temizleme), Modelleme, Değerlendirme ve Dağıtım** ⁴ . Benzer şekilde, bir diğer yaklaşımda sırasıyla *veri edinme (acquisition), veri temizleme, model geliştirme, model değerlendirme, dağıtım ve bakım* adımları vurgulanır ⁴ . Her aşamada veri odaklı hareket etmek kritik önem taşır. Örneğin, veri toplama aşamasında doğru veriyi seçmek, verinin yeterliliği ve temsil kabiliyeti için temelini oluşturur. Temizleme aşamasında eksik ve hatalı verilerin tespiti ve düzeltilmesi gerekir; bir makine öğrenmesi modeline çarpık veya eksik veri vererseniz sonuçlar da çarpık olur. Modelleme aşamasında hangi algoritmanın verinize uygun olduğunu seçmek önemlidir ancak en iyi algoritma bile kötü veriyi düzeltemez. Değerlendirme aşamasında kullanılan metriklerin (örneğin doğruluk, F1 skoru, ROC-AUC) proje gereksinimlerine uygun olması gerekir. Son olarak,

model dağıtımı ve sonrası aşamaları da veri yönetimiyle devam eder (örneğin modelin gerçek zamanlı beslenmesi, veri akışının takibi, model güncellemesi gibi).

Bu yaşam döngüsünün her aşamasında “**kaliteli veri = kaliteli sonuç**” kuralı geçerlidir. Düşünün, «kalitesiz malzemeyle gökdelen inşa etmek» gibidir: Dayanıksız temele sahip bir yapı uzun süre ayakta kalamaz ⁵ . Yine [15]’teki bir benzetme, verinin temelde inşa malzemesi olduğu fikrini vurgular: “Zayıf veri kalitesi, bataklık üzerinde uçak uçurmaya benzer. En yetenekli pilot bile yanlış verilerle felakete sebep olabilir” ⁶ . Aynı biçimde, kötü kalite veya eksik veriyle eğitilmiş bir yapay zeka modeli, yanlış önerilerde bulunur ya da önyargılı sonuçlar üretir. Bu nedenle analiz sürecine başlamadan önce veri toplama ve temizleme aşamalarına gereken titizliği göstermek, bu alanda başarıyı getirir.

Teoriden Pratiğe: Makine Öğrenmesi Mülakatlarında Uzmanlığınızı Konuşturun

Bu bölümde makine öğrenmesi alanındaki mülakatlarda sıkça sorulan sorulara yönelik örnekler sunacağız. Aşağıdaki başlıklar altında toplanmış temel konulardaki soru örnekleri, hem temel bilgilerinizi pekiştirmenize hem de mülakatta öne çıkmanıza yardımcı olacak:

Temel Kavramlar ve Algoritmalar

- **Supervised ve Unsupervised öğrenme arasındaki fark nedir?** Hangi durumlarda sınıflandırma (classification), hangi durumlarda kümeleme (clustering) tercih edilir?
- **Regresyon (Regression) ve Sınıflandırma (Classification) arasındaki temel fark nedir?** Örnek bir problem vererek açıklayınız.
- **Bias-variance tradeoff nedir?** Overfitting (aşırı öğrenme) ve underfitting (eksik öğrenme) kavramlarını tanımlayın. Bu durumlar nasıl tespit edilir?
- **En yaygın sınıflandırma algoritmalarını açıklayınız.** Lojistik regresyon, Karar Ağaçları, k-NN, Naive Bayes gibi algoritmalar nasıl çalışır, ne tür problemler için uygundur?
- **Doğrusal regresyon nasıl çalışır?** Modelin varsayımları nelerdir? R^2 metriği nedir, nasıl yorumlanır?
- **Kümeleme algoritmalarından K-Means ile DBSCAN’i karşılaştırın.** Hangi durumlarda her ikisi tercih edilir, hangi veri dağılımı için uygundurlar?
- **Karar Ağaçlarında (Decision Tree) ağaç derinliğinin (depth) performans etkisi nedir?** Ağaç derinliği neden modelin karmaşıklığını artırır?
- **L1 (Lasso) ve L2 (Ridge) düzenleme nedir?** Hangi durumlarda hangisini tercih edersiniz, aralarındaki fark nedir?
- **Gradient Descent nedir?** Minimizasyon problemlerinde nasıl kullanılır? Batch gradient descent ile stochastic gradient descent farkları nelerdir?
- **Ensemble yöntemleri nelerdir?** Random Forest, Boosting (örneğin XGBoost) gibi yöntemler nasıl çalışır, avantajları nelerdir?

Model Değerlendirme ve Metrikler

- **Confusion matrix (karmaşıklık matrisi) nedir?** Precision, Recall ve F1 skoru nasıl hesaplanır, neyi ifade eder?
- **Accuracy (doğruluk) metriğinin dezavantajı nedir?** Dengeli ve dengesiz (imbalanced) veri setlerinde hangi metrikler tercih edilmelidir (örn. ROC-AUC, MCC, log-loss)?
- **ROC eğrisi ve AUC nedir?** Modelin performansını nasıl değerlendirmek için kullanılır?

- **Model doğrulaması (validation) nasıl yapılır?** Train-test ayrımı nasıl olmalı, k-katlı çapraz doğrulama (k-fold cross-validation) ne işe yarar?
- **Overfitting'i önlemek için hangi teknikleri kullanırsınız?** Erken durdurma (early stopping), regularization, veri artırma (data augmentation) gibi yaklaşımlar nelerdir?
- **Precision-Recall eğrisi nedir?** AUC-ROC'a kıyasla ne zaman kullanışlıdır?
- **Regresyon modellerinin hata ölçütleri nelerdir?** MAE (Mean Absolute Error), MSE (Mean Squared Error), RMSE ve R^2 arasındaki farkları açıklayınız.
- **Model seçimi (model selection) ve hiperparametre ayarlama (hyperparameter tuning) nasıl yapılır?** Grid search veya rastgele arama (random search) yöntemleri nasıl çalışır?
- **Bir sınıflandırma modelinin yüksek precision fakat düşük recall'a sahip olması ne anlama gelir?** Bu durumda nasıl bir aksiyon alınabilir?
- **Karmaşık modelin kolay yorumlanamaması durumunda nasıl hareket edersiniz?** Interpretability (yorumlanabilirlik) için hangi yöntemler veya modeller tercih edilebilir?

Derin Öğrenme (Deep Learning)

- **Yapay sinir ağında (YSA) ileri beslemeli ağ (feedforward neural network) ile geriye yayılım (backpropagation) nasıl çalışır?** Adım adım anlatınız.
- **Aktivasyon fonksiyonları nelerdir?** Sigmoid, ReLU, tanh gibi fonksiyonlar ne zaman, neden kullanılır?
- **Konvolüsyonel Sinir Ağları (CNN) nedir?** Temel yapı taşları (convolution, pooling) nasıl çalışır, bilgisayarlı görü (computer vision) görevlerinde neden etkilidir?
- **Tekrarlayan Sinir Ağları (RNN) nedir?** LSTM ve GRU hücreleri neden kullanılır, zaman serisi verilerinde nasıl avantaj sağlar?
- **Transfer öğrenme (transfer learning) nedir?** Önceden eğitilmiş modeller nasıl kullanılır, hangi senaryolarda tercih edilir?
- **Overfitting'i önlemek için derin öğrenmede hangi teknikleri kullanırsınız?** Dropout, batch normalization, data augmentation gibi yöntemleri açıklayınız.
- **Derin öğrenmede optimizasyon algoritmalarından birkaçını karşılaştırın.** SGD, Adam, RMSprop gibi algoritmaların özellikleri nelerdir?
- **Derin öğrenmede temel hiperparametreler nelerdir?** Öğrenme hızı (learning rate), epoch sayısı, batch boyutu gibi terimlerin etkisini anlatınız.
- **Konvolüsyon katmanlarıyla tek farkı ne olan ağ: Fully Connected (Dense) katman nedir ve nerede kullanılır?**
- **Rekabetçi avantaj yaratmak için hangi derin öğrenme mimarileri öne çıkıyor?** Örneğin ResNet, Transformer mimarileri hakkında temel bilgi veriniz.

Boyut Azaltma ve Özellik Mühendisliği

- **Boyut indirgeme (dimensionality reduction) nedir?** Neden kullanılır, hangi algoritmalar yaygındır? (PCA, t-SNE, UMAP gibi)
- **Başlıca öznitelik seçimi (feature selection) yöntemleri nelerdir?** *Filter, wrapper, embedded* yöntemlerden örnekler veriniz.
- **Özellik mühendisliği (feature engineering) yaparken nelere dikkat edersiniz?** Kategorik değişkenleri (one-hot encoding, label encoding), sayısal değişkenleri (standardizasyon, normalizasyon) nasıl işlersiniz?

- **Veri setinde çok sayıda öznelik olduğunda bellek/süre açısından nasıl optimizasyon yapabilirsiniz?** Özellik çıkarımı (feature extraction) için PCA veya otomatik özellik öğrenme yöntemleri nasıl kullanılır?
- **PCA'nın temel amacı nedir?** Nasıl çalışır, veriyi 2D/3D'ye indirirken nelere dikkat edilmelidir?
- **Özellik oluşturma için domain bilgilendirmesini nasıl kullanabilirsiniz?** (Örneğin zaman damgasından gün/ay/saat çıkarmak, tarihleri sınıflandırmak, metin verisinde tf-idf vb.)
- **Yüksek boyutlu veride (high-dimensional data) "curse of dimensionality" (boyutsallık laneti) nedir?** Model performansını nasıl etkiler?
- **Bir özelliğin model tarafından öğrenilmesini istemiyorsanız ne yaparsınız?** Örneğin bir özelliğin varyansı çok düşük veya çok yüksekse nasıl yaklaşım belirlersiniz?
- **Öznelik üretimi (feature engineering) ve seçiminde Python kütüphanelerinden nasıl yararlanılır?** (Örneğin scikit-learn'ün `SelectKBest`, `Pipeline`, `ColumnTransformer` gibi araçları)
- **Veri dengesiz (imbalanced) sınıflandırma sorunlarında nasıl veri hazırlık teknikleri kullanırsınız?** Örnek: SMOTE, sınıf ağırlıklandırma vb.

Pratik Kodlama ve SQL

- **Python'da bir pandas DataFrame'de eksik değer (NaN) olduğunda ne yaparsınız?** Örneğin `dropna`, `fillna` kullanımı ve mantığı hakkında örnek veriniz.
- **Pandas ile bir DataFrame'de filtreleme (masking) nasıl yapılır?** Örneğin belirli bir sütundaki koşulları sağlayan satırları seçme şeklinde bir kod parçacığı gösteriniz.
- **Birden fazla DataFrame'i belirli bir anahtara göre birleştirmek (join) istersek Python'da nasıl yaparız?** (Örneğin `merge()` fonksiyonu kullanımı)
- **Scikit-Learn ile basit bir model eğitirken hangi adımları izlersiniz?** Eksik verileri doldurma (`SimpleImputer`), eğitim-veri ayırma (`train_test_split`), eğitme (`fit`) ve tahmin (`predict`) gibi temel adımları sıralayınız.
- **SQL'de iki tabloyu birleştirirken (JOIN) INNER JOIN ile LEFT JOIN arasındaki fark nedir?** Örnek bir sorgu üzerinden açıklayınız.
- **Bir ilişkisel veritabanında "normalizasyon" nedir ve neden önemlidir?** (1NF, 2NF, 3NF gibi temel normalizasyon kuralları)
- **SQL'de grup fonksiyonları (GROUP BY) ve toplama fonksiyonları (SUM, COUNT vs.) örnek bir sorguda nasıl kullanılır?**
- **Pandas'da `groupby` işlemini SQL'e benzeterek açıklayınız.** Kullanım örneği ile anlatınız.
- **Feature mühendisliği kod örneği olarak bir sütunun tipini dönüştürme veya tarih sütunu oluşturma örneği gösteriniz.** (ör. `pd.to_datetime`)
- **Veri görselleştirme için Python'da hangi kütüphaneleri kullanırsınız?** Basit bir matplotlib veya seaborn örneği veriniz.

Kod Örneği: Eksik Veri, Model Eğitimi ve Çapraz Doğrulama

Aşağıda, pandas ve scikit-learn kullanarak örnek bir veri setinde eksik değerleri nasıl işleyebileceğinizi, basit bir model eğitimi ve çapraz doğrulama ile model değerlendirmeyi gösteren bir Python kodu bulunuyor. Yorum satırlarıyla adım adım açıklamalar eklenmiştir.

```
import pandas as pd
import numpy as np
```

```

from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.impute import SimpleImputer
from sklearn.linear_model import LogisticRegression

# Örnek bir DataFrame oluşturalım
data = {
    'age': [25, np.nan, 30, 22, np.nan, 28, 35, np.nan, 40],
    'salary': [50000, 54000, np.nan, 58000, 60000, np.nan, 75000, 80000, 82000],
    'purchased': [0, 1, 0, 0, 1, 1, 0, 1, 0]
}
df = pd.DataFrame(data)

# Eksik verileri sütun ortalaması ile dolduralım
imputer = SimpleImputer(strategy='mean')
df[['age', 'salary']] = imputer.fit_transform(df[['age', 'salary']])

# Özellik ve hedef değişkenlerini ayır
X = df[['age', 'salary']]
y = df['purchased']

# Veriyi eğitim ve test olarak bölelim
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# Lojistik Regresyon modeli tanımla
model = LogisticRegression()

# 5-katlı çapraz doğrulama ile model başarısını değerlendirelim
scores = cross_val_score(model, X_train, y_train, cv=5, scoring='accuracy')
print("Cross-val doğruluk skorları:", scores)
print("Ortalama CV doğruluk:", np.mean(scores))

# Modeli eğitim verisiyle eğit ve testte tahmin yap
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

# Modelin test doğruluğunu hesapla
test_accuracy = (y_pred == y_test).mean()
print("Test doğruluğu:", test_accuracy)

```

Yukarıdaki kodda, önce bir örnek veri çerçevesindeki `age` ve `salary` sütunlarındaki eksik değerler ortalama (mean) ile dolduruluyor. Daha sonra bu veriler eğitim ve test olarak bölünüyor, lojistik regresyon modeli oluşturuluyor ve 5 katlı çapraz doğrulama ile modelin doğruluğu (accuracy) değerlendiriliyor. Son olarak model test verisiyle tahmin yapılıyor ve test doğruluğu hesaplanıyor. Bu örnek, gerçek bir projede de eksik veri ile nasıl başa çıkılabileceği ve model değerlendirmesi yapılabileceği konusunda temel bir şablon sunar.

Büyük Resmi Görmek: Veri Bilimi Mülakatlarında Stratejik Düşünün

Veri bilimi mülakatlarında teknik sorular kadar stratejik düşünme becerisi de değerlidir. Bu bölümde istatistik, iş odaklı vaka çalışmaları, SQL, Python ile veri manipülasyonu ve iletişim becerileri gibi konuları ele alan soru örnekleri yer alıyor. Bu sorular hem teknik bilginizi hem de analitik ve iş bakışınızı sınavacak niteliktedir.

İstatistik ve Olasılık

- **Olasılık temelleri:** Örneğin Bayes teoremi nedir, basit bir problem üzerinden Bayes hesaplaması yapınız.
- **Dağılımlar:** Normal dağılım, Bernoulli, Binom, Poisson dağılımı nedir, nerelerde uygulanır?
- **Hipotez testi ve p-değeri:** Null hipotezi (H0) ve alternatif (H1) nedir? p-değeri nasıl yorumlanır? Örneğin iki grup arasındaki farkı karşılaştırmak için hangi testleri kullanırsınız?
- **Güven aralığı (confidence interval) nedir?** Bir tahminin kesinliği nasıl belirlenir?
- **Örnekleme dağılımı ve merkezi limit teoremi:** Büyük veri analizi bağlamında merkezi limit teoremi neden önemlidir?
- **İstatistiksel güç (power) ve örnek büyüklüğü:** Bir çalışmada anlamlı sonuç elde etmek için yeterli örnek büyüklüğünü nasıl hesaplayabilirsiniz?
- **Veri analizi sorununda varyans vs varyans analizi (ANOVA):** Tek yönlü ve iki yönlü ANOVA farkı nedir, ne zaman kullanılır?
- **Covariance ve korelasyon nedir?** Bu kavramları gerçek dünya örnekleriyle açıklayınız.
- **Korelasyon ve nedensellik arasındaki fark nedir?** Yanlış yorumlamaları önlemek için nelere dikkat edilmelidir?
- **Central limit theorem'un veri analizindeki rolü nedir?** Örnek dağılımının normalizasyonu nasıl sağlanır?

Ürün Analizi ve İş Odaklı Vaka Çalışmaları

- **Bir ürün analizi senaryosu:** Diyelim ki e-ticaret sitesinde yeni bir özellik yayınlandı ve satışlar düştü. İlk 5 yapacağınız analiz ve sorular neler olur?
- **KPI (Key Performance Indicator) seçimi:** Bir projenin başarısını hangi metriklerle ölçersiniz? Örneğin bir mobil uygulama için kullanıcı bağlılığı nasıl ölçülür?
- **A/B testi tasarımı:** A/B testi nedir, nasıl planlanır? Örnek bir hipotez kurarak, deney ve kontrol grubu nasıl belirlenir?
- **ROI (Return on Investment) hesaplama:** Bir projenin yatırım geri dönüşünü hesaplamak için hangi bilgiler gereklidir, nasıl bir formül kullanırsınız?
- **Trafik kanalı analizi:** Bir sitenin trafik kaynaklarını (örneğin organik arama, sosyal, e-posta) incelerken hangi metriklere odaklanırsınız?
- **Zaman serisi analizi senaryosu:** Bir perakende şirketinin aylık satış verisinde mevsimsel etkiler olabilir mi, bunu nasıl incelersiniz?
- **Dönüşüm hunisi (conversion funnel) analizi:** Satış veya kullanıcı kaybı (drop-off) nerede oluyor, nasıl tespit edersiniz?
- **İş odaklı soru:** "Müşteri kaybını (churn) azaltmak için bir model geliştirmeniz istense, nasıl bir yol izlersiniz?" Bu durumda hangi analizleri yaparsınız?
- **Gerçek dünya verisiyle çalışma:** Veri setinde anormallikler (outlier) fark edilirse ne yaparsınız? (Sorunun iş etkisini de belirtin.)

- **E-Ticaret senaryosu:** İki farklı ürün grubunda satışları karşılaştırırken hangi testleri kullanırsınız? Örneğin fiyata duyarlı mı, kullanıcı davranışı nasıl analiz edilir?

SQL ve Veritabanı Yönetimi

- **SQL sorgusu örneği:** Çalıştığınız bir veri tabanında, belirli bir tarih aralığında en çok satış yapan 5 müşteriyi listeleyen bir SQL sorgusu yazınız. (JOIN ve GROUP BY kullanımını içersin.)
- **JOIN tipleri:** INNER JOIN, LEFT JOIN ve RIGHT JOIN arasındaki farkları örnek tablo yapıları üzerinden açıklayınız.
- **İndeks (index) nedir?** Bir veritabanında indeksleme nasıl performans kazandırır, örnek veriniz.
- **Transaction ve ACID ilkeleri:** Veritabanı işlemlerinde tutarlılığı (consistency) sağlamak için transaction özellikleri nelerdir? (Atomicity, Consistency, Isolation, Durability)
- **Veri normalizasyonu:** Bir tablonun 1NF, 2NF veya 3NF'de olup olmadığını belirlemek için tablo yapısını inceleyen bir örnek veriniz.
- **Veri modelleme:** Bir senaryoda (örneğin müşteri-bölge ilişkisi) uygun tablo yapısını nasıl tasarlıyorsunuz? (İlişkisel anahtarlar, çoktan çoğa ilişkiler vs.)
- **SQL performansı:** Karmaşık bir sorgu yavaş çalışıyorsa nasıl iyileştirirsiniz? (Subquery'leri kaldırmak, indeks eklemek vb.)
- **NoSQL farkları:** Geleneksel ilişkisel veritabanı yerine NoSQL (örneğin doküman tabanlı MongoDB) tercih edilirse avantajları/dezavantajları nelerdir?
- **Veri aktarımı:** Büyük bir tabloyu (örneğin birkaç milyon satır) birleştirirken veya kopyalarken karşılaşılabilecek zorluklar neler olabilir?
- **SQL fonksiyonları:** `OVER()` ile pencere fonksiyonları (window functions) kullanarak kumulatif toplam veya sıralama yapmak için örnek bir sorgu veriniz.

Python ile Veri Manipülasyonu

- **Pandas veri temizleme:** Belirli bir sütunda çok fazla eksik değer varsa ne yaparsınız? (`dropna`, `fillna` kararları gibi)
- **Python'da veri setleri arasında birleştirme:** Örneğin iki farklı DataFrame'de ortak bir anahtarla veri birleştirmek için nasıl kod yazarsınız? (`pd.merge` örneği)
- **Grup bazlı işlemler:** Bir DataFrame'i birden fazla sütun bazında gruplamak ve her grup için özel bir hesaplama yapmak için nasıl kod yazarsınız? (`groupby` kullanım örneği)
- **Python'da pivot tablolar:** `pivot_table` veya `crosstab` fonksiyonlarını örnekleyerek nerede kullanılır açıklayınız.
- **Veri görselleştirme kod örneği:** Matplotlib veya Seaborn ile basit bir çubuk grafiği oluşturma adımlarını gösteriniz.
- **Zaman serisi verileri:** Pandas'ta tarih sütunu oluşturma ve bunu indeks olarak ayarlama nasıl yapılır? (örnek kod ile)
- **Python ile büyük veri seti yükleme:** Bellek kısıtlaması varsa `chunksize` ile nasıl veri okunur? Örnek veriniz.
- **Veri sorgulama:** Pandas'ta bir koşulu sağlayan satırları tek satırda nasıl filtrelersiniz? (örnek: `df[df['year'] == 2020]`)
- **API veya veri çekme:** Python'da bir web API'dan veri çekmek için hangi kütüphaneler kullanılır? Örneğin `requests` ile JSON veriyi DataFrame'e çevirme örneği.
- **Kod açıklamaları:** Kodunuzu başkalarının da anlaması için nelere dikkat edersiniz? (Fonksiyon, değişken isimlendirme, yorum satırları gibi.)

İletişim ve Sunum Becerileri

- **Sonuçları nasıl sunarsınız?** Teknik olmayan bir yöneticinize, karmaşık bir modelin sonuçlarını nasıl özetler ve anlatırsınız?
- **Veri hikayesi (data storytelling) nedir?** Örneğin bir grafik veya model çıktısı üzerinden iş impact'ı nasıl vurgulanır?
- **Takım çalışması:** Çapraz fonksiyonel bir ekipte (örneğin ürün, mühendislik ve iş birimleri) veri projenizi nasıl yürütür ve iletişim kurarsınız?
- **Müşteri odaklı yaklaşım:** Bir proje sunumu sırasında teknik seçimlerinizin müşteriye faydasını nasıl açıklarsınız?
- **Etik ve gizlilik:** Veri bilimi yaparken veri gizliliği veya etik sorunlarla karşılaşırsanız nasıl önlem alır veya hangi kurallara uyar sağlarsınız?
- **Geribildirim yönetimi:** Yöneticiler veya müşteriler projenin seyrini değiştirecek bir talep getirdiğinde nasıl adaptasyon sağlarsınız?
- **Hipotez bazlı çalışma:** Bir analize başlamadan önce varsayımlarınızı nasıl test edersiniz? Örneğin, hipotez belirleme adımlarını anlatınız.
- **Zaman yönetimi:** Birden fazla proje veya acil iş geldiğinde önceliklerinizi nasıl belirlersiniz?
- **Teknik olmayan sorulara cevap:** "Bu model nasıl çalışıyor?" gibi teknik olmayan bir soruya basit bir benzetmeyle nasıl yanıt verirsiniz?
- **Takım içi sunum:** Daha önce bir ekip toplantısında gerçekleştirdiğiniz önemli bir sunum deneyimini ve çıktısını kısaca anlatınız. (Hikaye anlatımı ve sonuç vurgusu.)

Sonuç

Yukarıda detaylıca ele aldığımız üzere, yapay zekada başarı **yalnızca en iyi algoritmayı seçmekle değil**, veriye dayalı stratejiyle mümkün olur. Proje yaşam döngüsünün her aşamasında – veriyi toplarken, temizlerken, modellerken ve sonuçları sunarken – *veri kalitesine ve uygun yönetişime* odaklanmak, projeyi başarıya taşıyan "gizli formül"dendir. Girişte gördüğümüz yüksek başarısızlık oranları, çoğu zaman algoritmadan ziyade veri eksikliği, kötü veri yönetimi veya yanlış problem tanımlamasından kaynaklanır ¹

² .

Teknik sorulara hazırlıklı olmak önemlidir; ancak bir adım daha ileri giderek **neden ve nasıl** sorularını da sormak gerekir. Başardığınız her model veya analiz, iş hedefine hizmet etmeli ve sonuçları somut bir değere dönüştürmelidir. Unutmayın ki işe alım uzmanları ve yöneticiler artık sadece kod bilen değil, "veri bilimini iş stratejisine çeviren" profesyoneller arıyor ⁷ . Onlar için önemli olan "veri karmaşasını yöneticiler için anlaşılır bilgiye dönüştürebilmek"tir ⁷ . Sizi fark ettirecek olan, algoritmayı doğru uygulamanın ötesinde büyük resmi görmeniz ve yaptığınız işin arkasındaki mantığı ifade edebilmenizdir.

Kısacası, **kaliteli veri ve sağlam yönetişim** ile birlikte geniş bir teknik yelpazede bilgi birikiminizi bir araya getirirseniz, yapay zeka projelerinde gerçekten fark yaratabilirsiniz. Sorularımız ve kod örneğimiz size bu anlamda bir çerçeve sunuyor; önemli olan üzerinde düşünmek, pratik yapmak ve her zaman "neden böyle yapıyoruz?" sorusunu sorabilmektir. Başarılı projeler kod satırlarından ziyade bu tür stratejik yaklaşımlardan doğar ⁷ ⁵ .

1 Why Big Data Science & Data Analytics Projects Fail

<https://www.datascience-pm.com/project-failures/>

2 The Root Causes of Failure for Artificial Intelligence Projects and How They Can Succeed: Avoiding the Anti-Patterns of AI | RAND

https://www.rand.org/pubs/research_reports/RRA2680-1.html

3 What is Data Governance? | IBM

<https://www.ibm.com/think/topics/data-governance>

4 Life Cycle of a Data Science Project

<https://www.projectpro.io/article/life-cycle-of-a-data-science-project/270>

5 6 What is Data Quality & why it is important?

<https://scikiq.com/blog/what-is-data-quality-why-it-is-important/>

7 The 6 Real Reasons Companies Hire Data Scientists (And How to Use Each to Your Advantage)

<https://www.linkedin.com/pulse/6-real-reasons-companies-hire-data-scientists-how-use-dan-sanz--oj3ae>