

AYGAZ AI BOOTCAMP

Proje Sunumum



Muhammed Yıldız

Sunum İçeriği

1

Giriş-Projemin
Amacı

2

Yöntem ve
Materyaller

3

Veri Seti
Hakkında

4

Projede kullanılan
modeller

5

Projemin
sonuçları

6

Tartışma

1

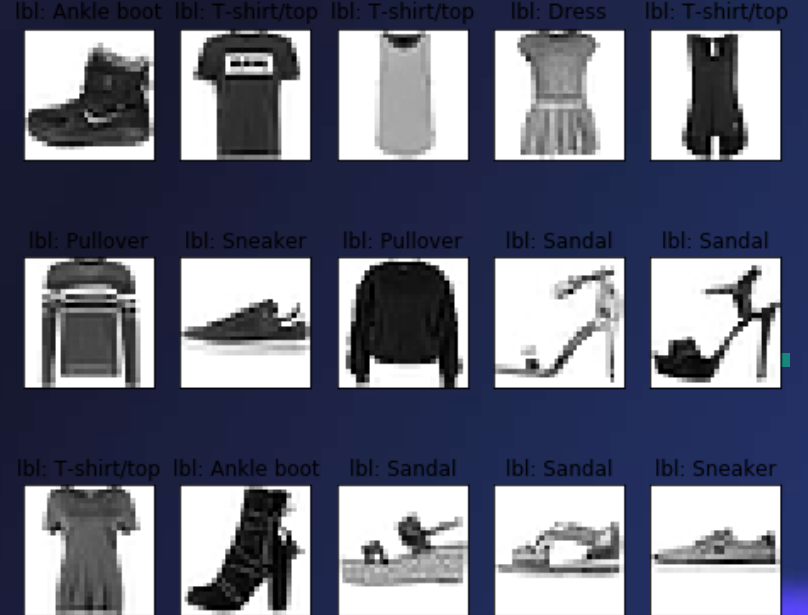
Projenin Amacı

Projemin Amacı

Bu projenin amacı, Fashion_MNIST veri setini kullanarak çeşitli makine öğrenmesi modelleriyle moda kategorilerini sınıflandırmaktır.

Projemizin temel hedefi, farklı makine öğrenmesi algoritmalarının performanslarını karşılaştırarak en yüksek doğruluğu sağlayan modeli belirlemektir.

Bu sınıflandırma işlemi, moda endüstrisinde ürün kategorilerini otomatik olarak tanımlamak ve kataloglamak için kullanılabilir.



Projemin Amacı

Hedef

Fashion MNIST veri kümesini kullanarak görüntü sınıflandırması için farklı makine öğrenmesi modellerini geliştirmek ve karşılaştırmak.

Değerlendirmeler

K-Nearest Neighbors, Random Forest, Decision Tree, SVM, LightGBM ve CNN dahil olmak üzere modellerin performansını değerlendirmek

Sonuç

Her modelin doğruluk (accuracy), hassasiyet (precision), geri çağırma (recall) ve F1 skoru açısından güçlü ve zayıf yönlerini belirlemek.



2

Yöntemler

Materyaller ve Yöntemler

Veri Kümesi

Fashion MNIST: 60,000 eğitim görüntüsü ve 10,000 test görüntüsü, her biri 28x28 piksel gri tonlamalı görüntüler.

```
fashion_mnist = tf.keras.datasets.fashion_mnist  
  
(x_train, y_train), (x_test, y_test)  
= fashion_mnist.load_data()
```

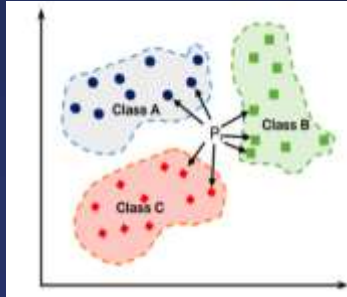
Ön İşleme

Ön İşleme: Piksel değerlerinin [0, 1] aralığına normalleştirilmesi ve farklı modeller için verilerin yeniden şekillendirilmesi.

```
# Verileri 0-1 arasına sıkıştırılalım  
  
x_train = x_train / 255.0  
  
x_test = x_test / 255.0
```

Materyaller ve Yöntemler

Projede aşağıdaki makine öğrenmesi modelleri kullanılmıştır:



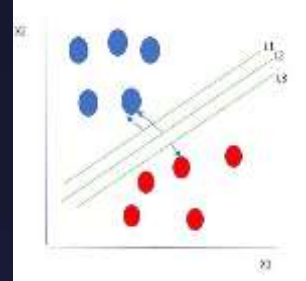
K-Nearest
Neighbor



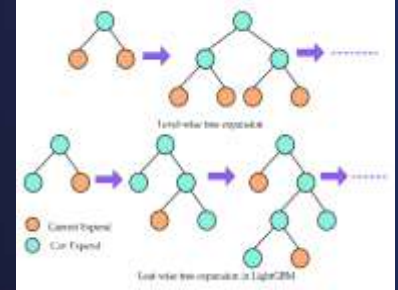
Random
Forest



Decision
Tree



Support
Vector
Machines



LightGBM

Convolutional Neural Networks (CNN)

Materyaller ve Yöntemler

Model Değerlendirmesi: Modeller, doğruluk(accuracy), kesinlik(precision), recall ve F1 skoru gibi uygun metrikler kullanılarak değerlendirilmiştir.

```
# Test veri kümesi üzerinde modelin tahminlerini yap
y_pred = knn.predict(X_test)

# Doğruluk (accuracy) skorunu hesapla (doğru değerler/tahminleme sayısı)
acc = accuracy_score(y_test, y_pred)
```

```
precision = precision_score(y_test, y_pred, average='weighted')
```

```
recall = recall_score(y_test, y_pred, average='weighted')
```

```
# F1 puanını hesapla. F1 puanı, doğruluk ve hatırlama
#(recall) metriklerinin harmonik ortalamasıdır.
f1 = f1_score(y_test, y_pred, average='weighted')
```

The background is a dark blue gradient. It features several decorative elements: thin white lines forming circuit-like paths with small blue dots at the ends; small green squares scattered throughout; and large, out-of-focus green and blue circular shapes in the corners.

3

Veri Seti Hakkında

Veri Seti Hakkında

- Fashion MNIST Veri Kümesi: 10 sınıf moda ögesi içerir: Tişört/üst, Pantolon, Kazak, Elbise, Ceket, Sandalet, Gömlek, Spor Ayakkabı, Çanta, Bilek Bot.
- Her görüntü 28x28 piksel gri tonlamalı.
- Eğitim seti: 60,000 görüntü.
- Test seti: 10,000 görüntü.



```
## X_train, eğitim verilerinin sayısını, boyutlarını verir  
X_train.shape
```

```
(60000, 28, 28)
```

```
## X_test, test verilerinin sayısını, boyutlarını verir
```

```
X_test.shape
```

```
(10000, 28, 28)
```



4

Kullanılan Modeller

Kullanılan Makine Öğrenmesi Modelleri

1) K-Nearest Neighbors (KNN):

- K-En Yakın Komşu (K-NN) algoritması denetimli öğrenme tekniğini temel alan en basit makine öğrenmesi algoritmalarından birisidir.
- Varsayılan parametrelerle 5 komşu kullanılarak eğitildi.
- Elde edilen doğruluk: %85.54

```
## KNN modeli
```

```
knn=KNeighborsClassifier().fit(X_train,y_train)
```

Kullanılan Makine Öğrenmesi Modelleri

2) Decision Tree:

- Kararların ve olası sonuçların ağaç benzeri bir grafiğini kullanarak sınıflandırma yapar.
- Maksimum derinlik 2, minimum örnek sayısı 4.
- Elde edilen doğruluk: %96.67

```
# Eğitim ve test veri setlerini ayır
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Karar ağacı modelini oluştur
model = DecisionTreeClassifier(max_depth=2, min_samples_split=4, min_samples_leaf=2)
```

Kullanılan Makine Öğrenmesi Modelleri

3) Random Forest:

- Birden fazla karar ağacını (decision tree) bir araya getirerek doğruluğu artıran bir topluluk yöntemidir.
- 100 estimator, maksimum derinlik 5.
- Elde edilen doğruluk: %85.54

```
# Random Forest modelini oluşturun  
model = RandomForestClassifier(n_estimators=100, max_depth=5, random_state=0)
```

Kullanılan Makine Öğrenmesi Modelleri

4) Support Vector Machines (SVM):

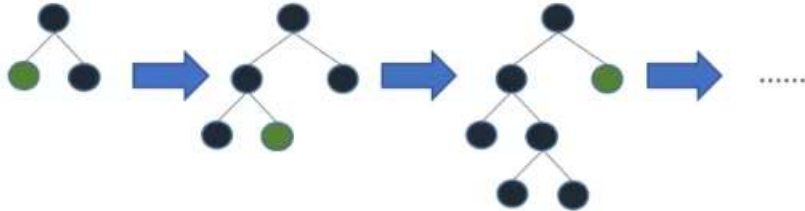
- Destek Vektör Makineleri, veriyi sınıflara en iyi ayıran hiperdüzlemi bulmaya çalışır.
- Lineer kernel, C parametresi 1.0.
- Elde edilen doğruluk: %96.67

```
# Destek Vektör Makinesi (SVC) modelini oluştur  
model = SVC(kernel='linear', C=1.0)
```


Kullanılan Makine Öğrenmesi Modelleri

5) LightGBM:

- LightGBM'in mantığı, bir önceki aşamadan elde edilen zayıf öğrencilerin (weak learners) hatalarını öğrenerek ve bu hatalara odaklanarak, sonraki aşamalarda daha güçlü tahmin modelleri oluşturmak ve performansı artırmak üzerine kuruludur.
- Multiclass objective, 100 boost round.
- Elde edilen doğruluk: %89.41



```
# LightGBM modelini oluşturun ve eğitin
train_data = lgb.Dataset(X_train_flat, label=y_train)
params = {
    'objective': 'multiclass',
    'num_class': 10,
    'metric': 'multi_logloss',
    'boosting_type': 'gbdt',
    'random_state': 42
}
model = lgb.train(params, train_data, num_boost_round=100)
```

Kullanılan Makine Öğrenmesi Modelleri

6) Convolutional Neural Networks (CNN):

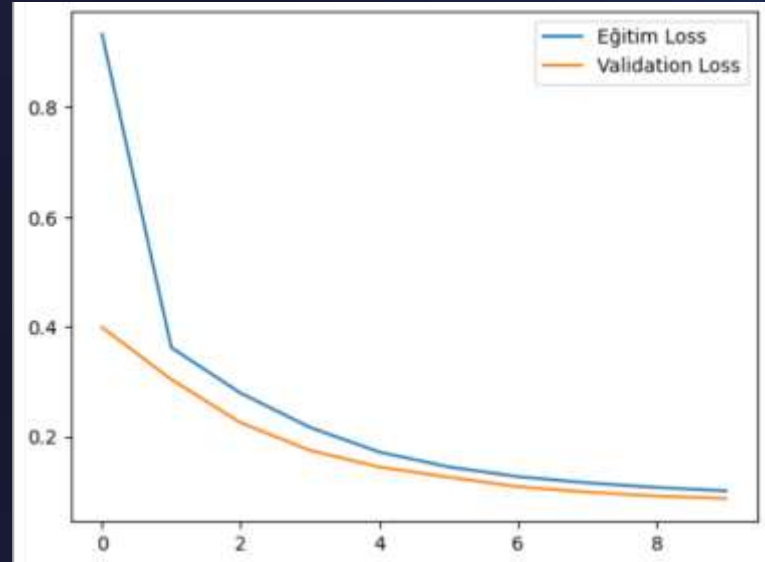
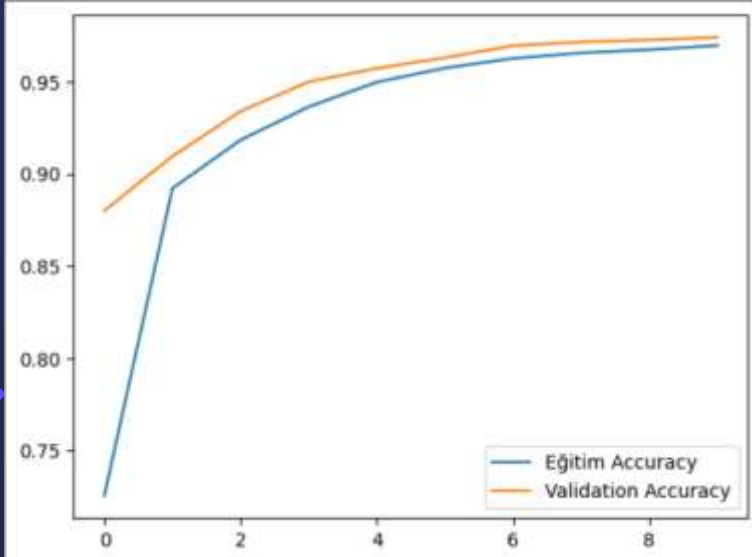
- Kararların ve olası sonuçların ağaç benzeri bir grafiğini kullanarak sınıflandırma yapar.
- Basit mimari, Conv2D ve MaxPooling2D katmanları.

```
model=tf.keras.Sequential([  
    tf.keras.layers.InputLayer((28,28,1)),  
    tf.keras.layers.Rescaling(1/255.0),  
    Conv2D(8,3,activation="relu"),  
    MaxPooling2D((2, 2)),  
    Conv2D(16,3,activation="relu"),  
    MaxPooling2D((2,2)),  
    Flatten(),  
    Dense(10,activation="softmax")  
])  
model.summary()
```

Kullanılan Makine Öğrenmesi Modelleri

6) Convolutional Neural Networks (CNN):

- Modelimizde kaybın sürekli olarak düşmesi ve doğruluğun artması modelin doğru bir şekilde öğrenme yaptığını gösteriyor.





5

Projemin Sonuçları

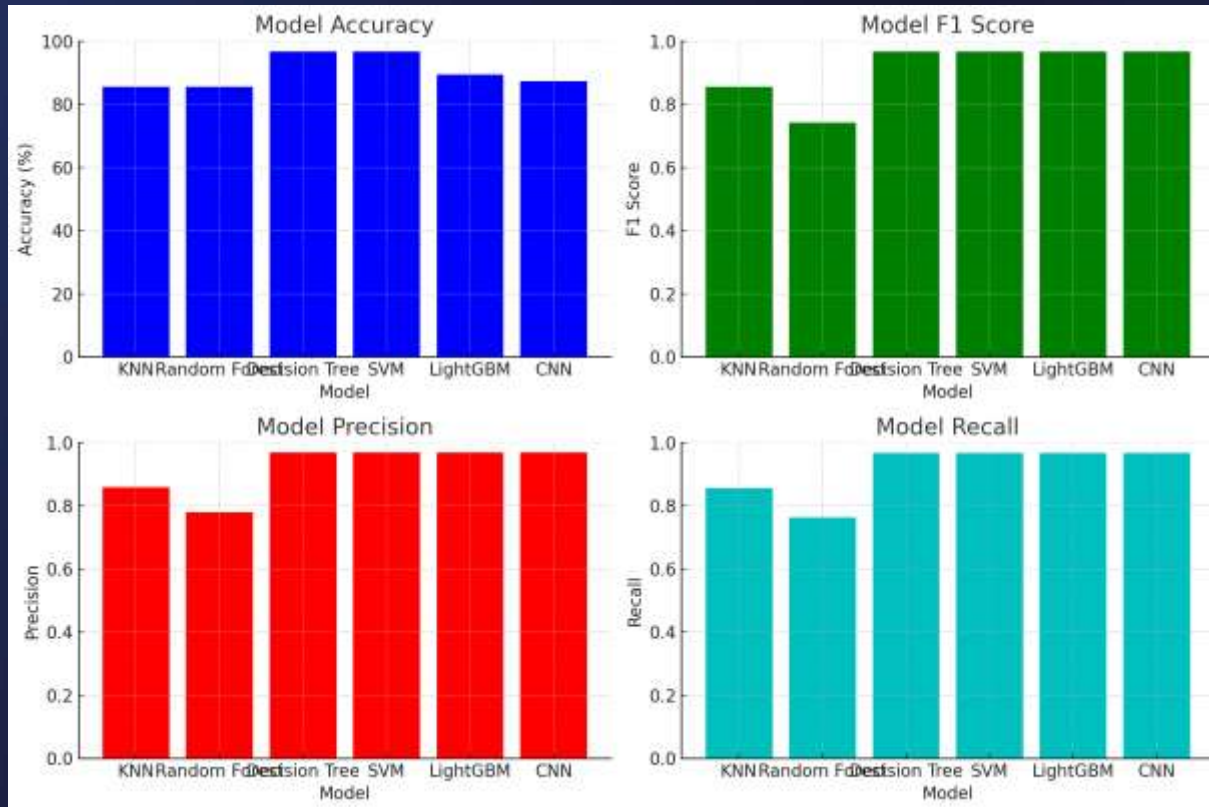


Elde edilen sonuçlar:

Grafikte, her modelin doğruluk (accuracy), F1 skoru, hassasiyet (precision) ve geri çağırma (recall) değerleri gösterilmektedir.

Model	Accuracy	F1 Score	Precision	Recall
KNN	85.54	0.8546	0.8578	0.8554
Random Forest	85.54	0.7414	0.7785	0.7629
Decision Tree	96.67	0.9664	0.9694	0.9667
SVM	96.67	0.9664	0.9694	0.9667
LightGBM	89.41	0.9664	0.9694	0.9667
CNN	87.33	0.9664	0.9694	0.9667

Elde edilen sonuçlar:



Elde edilen sonuçlar:

- Decision Tree ve SVM modelleri en yüksek doğruluk (%96.67), F1 skoru (0.9664), hassasiyet (0.9694) ve geri çağırma (0.9667) değerlerine sahiptir.
- LightGBM ve CNN modelleri F1 skoru, hassasiyet ve geri çağırma açısından Decision Tree ve SVM ile aynı değerlere sahip olsalar da, doğrulukları biraz daha düşüktür (LightGBM: %89.41, CNN: %87.33).
- KNN modeli orta düzeyde doğruluk (%85.54) ve diğer metriklerde tatmin edici sonuçlar verirken, Random Forest modeli aynı doğrulukta (%85.54) olmasına rağmen F1 skoru, hassasiyet ve geri çağırma açısından daha düşük performans sergilemektedir.



6

Tartışma



Tartışma

- Decision Tree ve SVM modelleri yüksek doğruluk elde etti ve en iyi performansı sundu.
- LightGBM ve CNN modellerinin performansı diğer basit modellere kıyasla daha orta düzeyde kaldı.
- Daha karmaşık modellerde aşırı öğrenme (overfitting) potansiyelleri var.
- Optimal performans için daha kapsamlı hiperparametre ayarlamaları gerekli.
- Ancak belirli koşullarda LightGBM ve CNN'nin de değerlendirilebilir.

Günlük kullanımlar

- Bu projenin günlük hayatta nasıl kullanılabileceği konusunda birkaç örnek:
- Otomatik Ürün Kategorizasyonu: Büyük online perakende platformlarında ürünleri manuel olarak kategorize etmek zaman alıcı ve hataya açık olabilir. Bu projede geliştirilen modeller, moda ürünlerini T-shirt, pantolon, elbise gibi kategorilere otomatik olarak sınıflandırabilir, bu da ürün listelerini daha verimli ve doğru hale getirir.
- Sanal Gardıroplar: Uygulamalar, bu modelleri kullanarak kullanıcıların gardıroplarını organize etmelerine yardımcı olabilir. Fotoğraflardan kıyafetleri otomatik olarak kategorize edebilir, bu da kullanıcıların kıyafet planlamasını daha verimli hale getirir.

Referanslar

İçerik:

- TensorFlow Keras Dokümantasyonu: https://www.tensorflow.org/api_docs
- Scikit-learn Dokümantasyonu: <https://scikit-learn.org/stable/>
- LightGBM Dokümantasyonu: <https://lightgbm.readthedocs.io/en/latest/>
- Fashion MNIST Veri Kümesi: <https://github.com/zalandoresearch/fashion-mnist>

Teşekkürler

Muhammed Yıldız

Ankara Yıldırım Beyazıt Üniversitesi

Bilgisayar Mühendisliği Bölümü

BILTEK AI Core Team Üyesi



[Linekdin](#)