# Sonuçlar - En İyi Performanslar

## Geleneksel Yöntemler - Basic False

|  |  |  |
| --- | --- | --- |
| Özellik Temsili / Model | Sınıflandırıcı | F1 Skoru |
| bow + tf-idf | MNB | 0.6050 |
| bow-tf | MNB | 0.5956 |
| bow + binary | MNB | 0.5920 |
| 3-grams + tf | KNN | 0.6374 |
| 3-grams + tf-idf | LR | 0.5869 |

## Geleneksel Yöntemler - Basic True

|  |  |  |
| --- | --- | --- |
| Özellik Temsili / Model | Sınıflandırıcı | F1 Skoru |
| bow + tf-idf | MNB | 0.6225 |
| bow-tf | MNB | 0.6156 |
| bow + binary | MNB | 0.6170 |
| 3-grams + tf | KNN | 0.6184 |
| 3-grams + tf-idf | MNB | 0.6079 |

## Word Embedding - Basic False

|  |  |  |
| --- | --- | --- |
| Özellik Temsili / Model | Sınıflandırıcı | F1 Skoru |
| GloVe | KNN | 0.6640 |
| FastText | KNN | 0.6660 |
| Word2Vec | KNN | 0.6670 |

## Word Embedding - Basic True

|  |  |  |
| --- | --- | --- |
| Özellik Temsili / Model | Sınıflandırıcı | F1 Skoru |
| GloVe | KNN | 0.6650 |
| FastText | KNN | 0.6660 |
| Word2Vec | KNN | 0.6670 |

## Transformer Modeller - Basic False

|  |  |  |
| --- | --- | --- |
| Özellik Temsili / Model | Sınıflandırıcı | F1 Skoru |
| BERT | KNN | 0.6660 |
| MarianMT | KNN | 0.6660 |
| CamemBERT | KNN | 0.6660 |

## Transformer Modeller - Basic True

|  |  |  |
| --- | --- | --- |
| Özellik Temsili / Model | Sınıflandırıcı | F1 Skoru |
| BERT | KNN | 0.6660 |
| MarianMT | KNN | 0.6660 |
| CamemBERT | KNN | 0.6660 |

**SONUÇLARIN YORUMLANMASI**

* BASIC TRUE özelliği kullanıldığında, BASIC FALSE özelliğine göre hafif bir performans artışı gözlemlenmektedir.
* Geleneksel yöntemler, özellikle MNB sınıflandırıcısı ile birlikte kullanıldığında, bu veri kümesi üzerinde makul bir performans sergilemektedir.
* Word embedding ve BERT modelleri, bazı durumlarda daha iyi sonuçlar verebilir, ancak performansları tutarsızdır ve dikkatli bir hiperparametre ayarlaması gerektirmektedir.
* En iyi sonuçlar, kullanılan özelliklere, modele ve hiperparametrelere bağlı olarak değişmektedir. Bu nedenle, belirli bir görev için en iyi modeli ve parametreleri belirlemek için kapsamlı bir deneysel değerlendirme yapılması gerekir.
* **KMEANS** ve **MiniBatchKMeans** gibi kümeleme algoritmaları, genel olarak düşük F1-skora değerleri sunmuş, bu da bu yöntemlerin sınıflandırma için uygun olmadığını gösteriyor.
* "glove", "fasttext" ve "word2vec" gibi kelime gömme yöntemleri kullanıldığında, "KNN" sıklıkla en iyi sonuçları vermiştir. Ancak bazı durumlarda, "MNB" ve "SVM" daha düşük performans göstermiştir.
* **MNB (Naive Bayes)**, her iki grupta da en yüksek F1-skora değerlerine ulaşmıştır. Bu, MNB'nin metin sınıflandırma görevlerinde etkili olduğunu göstermektedir.

 **Geleneksel Yöntemler**:

* **BoW (Bag-of-Words)** ile kullanılan modeller arasında **MNB (Naive Bayes)** genelde daha yüksek bir F1 skoru gösteriyor, bu da basit kelime frekansı temelli yaklaşımların MNB ile etkili çalıştığını gösteriyor.
* Ancak diğer algoritmalarda (**LR, SVM, RF** vb.) performans tutarsız görünüyor ve kullanılan temsil yöntemine bağlı olarak değişiyor. Örneğin, **3-grams + TF-IDF** genelde daha yüksek performans sağlıyor.

 **Word Embedding Yöntemleri**:

* **GloVe, FastText ve Word2Vec** gibi embedding tabanlı temsillerde, **KNN (k-Nearest Neighbors)** sürekli olarak daha yüksek F1 skorları veriyor. Bu, embedding yöntemleriyle basit bir sınıflandırıcının iyi performans gösterebileceğini işaret ediyor.
* Ancak **MNB** gibi modellerin bu temsillerle başarısız olduğunu görüyoruz, bu da yöntemler arasındaki uyumsuzluğu ortaya koyuyor.

 **Transformer Modelleri (BERT, T5 vb.)**:

* Transformer tabanlı modellerde, skorlar genelde daha dengeli ama yüksek performans gösteren açık bir model göze çarpmıyor. Bu, hiperparametre ayarlarının bu modellerde daha kritik olabileceğini gösterebilir.
* **KNN** burada da diğer modellere kıyasla biraz daha öne çıkıyor.

 **Genel Performans ve Öneriler**:

* Geleneksel yöntemler için **TF-IDF + MNB**, Word Embedding için ise **KNN** genelde daha yüksek performans veriyor.
* Daha karmaşık modellerde, özellikle BERT gibi transformer tabanlı modellerde, sonuçların artırılması için hiperparametre optimizasyonu veya daha fazla veriyle eğitim gerekebilir.

### 1. ****Kategorilere Göre Performans Analizi****

* Geleneksel yöntemler (**BoW, TF-IDF**) ile karşılaştırıldığında embedding tabanlı yöntemlerin (**GloVe, FastText**) genel olarak daha iyi performans gösterdiği görülüyor. Özellikle **KNN**, embedding yöntemleriyle yüksek başarıya ulaşıyor.
* Transformer modellerinin (**BERT**) özellikle KNN ile güçlü bir kombinasyon sunduğu anlaşılıyor.

### 2. ****Algoritma Bazlı İnceleme****

* **KNN**: En iyi sonuçları embedding tabanlı yöntemlerle sağlamış. Bunun nedeni, KNN'nin yüksek boyutlu uzaylarda iyi çalışması ve embedding temsillerinin anlamlı bağlamlar sağlaması olabilir.
* **MNB (Naive Bayes)**: Geleneksel yöntemlerle (BoW + TF-IDF) yüksek F1 skorlarına ulaşmış. MNB'nin özellikle kelime frekansı dağılımını iyi analiz etmesi bu performansı açıklayabilir.
* **SVM ve RF (Random Forest)**: Göreceli olarak tutarlı ancak en yüksek skorlara ulaşamamış. Bu algoritmalar için hiperparametre optimizasyonu veya daha fazla veri kullanımı faydalı olabilir.

### 3. ****Özellik Temsili Karşılaştırması****

* **3-grams + TF-IDF** yöntemi, kelime dizilerini dikkate alarak bağlamı daha iyi temsil ediyor gibi görünüyor. Bu temsille KNN yüksek performans göstermiş.
* **Word Embedding'ler** (GloVe, FastText): Genel olarak bağlamsal temsilde güçlü oldukları için KNN ve RF gibi algoritmalara uygun altyapı sağlamışlar.

### 4. ****Hiperparametre ve Kümeleme Etkisi****

* Kümeleme algoritmalarının (ör. KMeans, Agglomerative Clustering) sınıflandırıcıların performansını etkileyip etkilemediği analiz edilebilir. Özellikle embedding yöntemleriyle kümeleme uyumu incelenebilir.

### Öneriler:

1. **Hiperparametre Optimizasyonu**: RF, SVM gibi algoritmaların performansı iyileştirilebilir.
2. **Derin Öğrenme Modelleri**: Transformer tabanlı modellerle daha derin analiz yapılabilir. Örneğin, BERT'in ince ayarı (fine-tuning) performansı artırabilir.
3. **Veri Seti Analizi**: Verinin sınıf dağılımı, dengesizlik gibi faktörler değerlendirilerek SMOTE gibi teknikler kullanılabilir.
4. **Hata Analizi**: Düşük performanslı modellerin hataları incelenerek daha hedefe yönelik iyileştirme yapılabilir.

### Sonuç

Genel olarak, KMEANS ve MiniBatchKMeans algoritmaları, diğerlerine göre daha yüksek doğruluk değerleri sunmaktadır. KNN sınıflandırıcısı, en yüksek doğruluk değerlerini elde eden sınıflandırıcı olarak öne çıkmaktadır. MNB ise, çoğu durumda düşük performans göstermektedir.

**MNB'nin F1 skorunun 0 olmasının olası nedenleri şunlar olabilir:**

* **Veri uyumsuzluğu**: MNB, verilerin belirli bir dağılımı izlediğini varsayar. Eğer veriler bu varsayımı karşılamıyorsa, model beklendiği gibi performans göstermeyebilir.
* **Özellik mühendisliği**: MNB'nin performansı, kullanılan özelliklere duyarlıdır. Bazı durumlarda, BERT modelleriyle çıkarılan özellikler MNB için uygun olmayabilir.
* **Hiperparametre ayarı**: Herhangi bir makine öğrenimi modelinde olduğu gibi, MNB'nin performansı hiperparametrelere bağlıdır. Yanlış hiperparametre seçimi, düşük performansa neden olabilir.
* **Dengesiz Veri:** Eğer veri setinde sınıflar arasında büyük bir dengesizlik varsa, MNB modeli çoğunluk sınıfını tahmin eder ve bu da doğruluğun 0 olmasına yol açabilir.
* **Yetersiz Veri:** Modelin eğitilmesi için yeterli örneklem yoksa, model öğrenme sürecinde başarısız olabilir.