Spam E-Posta Tespit Sistemi Projesi

Hazırlayan:Mustafa Gökhan GÜLDAŞ

Tarih: 24.12.2025

1.Giriş

Bu proje, "AI TASK 2" kapsamında verilen e-posta veri seti kullanılarak Spam/Normal sınıflandırması yapan bir yapay zeka sistemi geliştirmek amacıyla hazırlanmıştır. Proje, NLP ve makine öğrenmesi pipeline'ı oluşturularak gerçekleştirilmiştir.

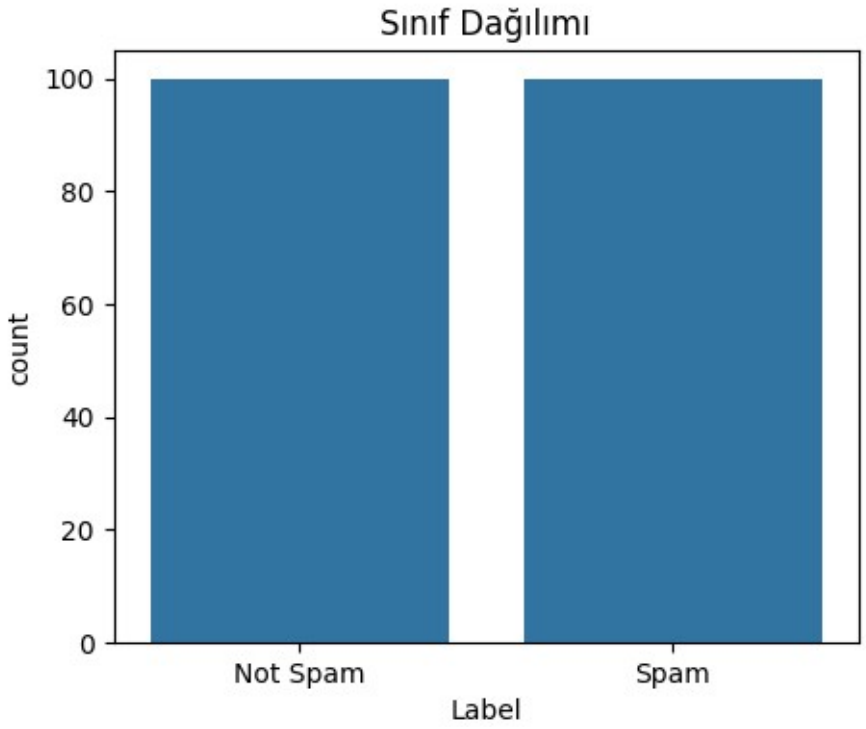
2. Task Kapsamı ve Analizler

1) Veri İnceleme (EDA)

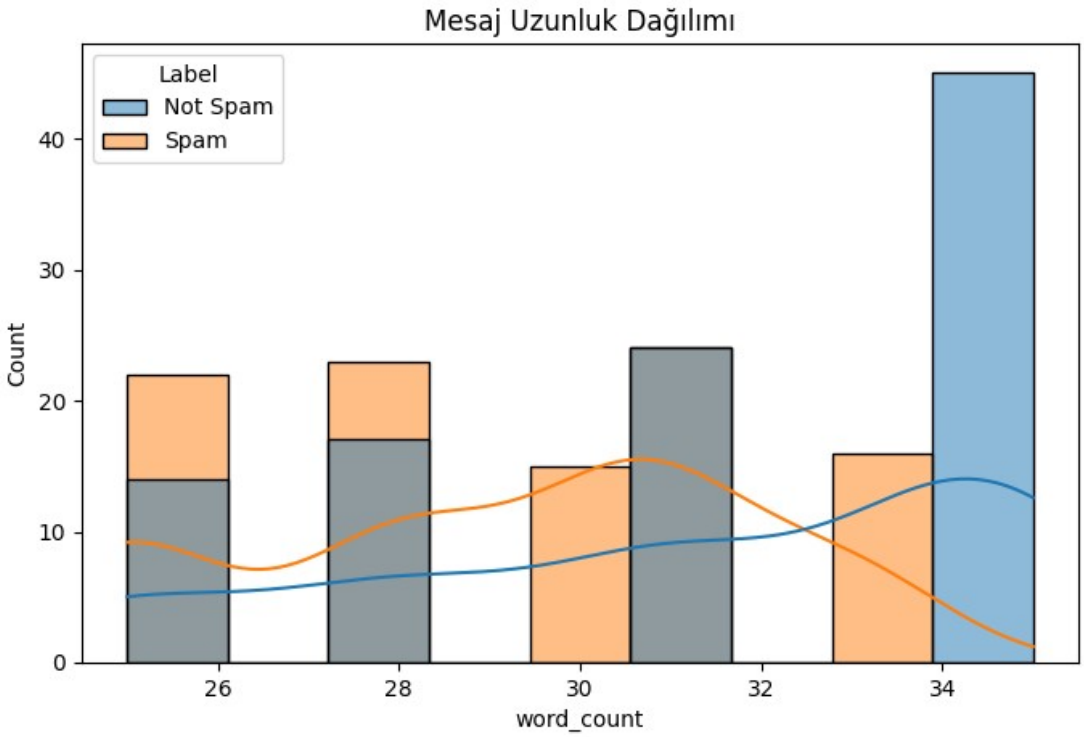
Veri seti pandas ile incelenmiş ve tamamen dengeli olduğu görülmüştür.

* Boyut: 200 satır (100 Spam, 100 Normal).
* Kelime Analizi: Ortalama kelime sayısı 30.2'dir. Spam ve Normal maillerin uzunlukları benzerdir.
* En Sık Geçen Kelimeler:`please` (75 kez), `get` (62 kez), `review` (51 kez). (Stopwords temizlendikten sonra).

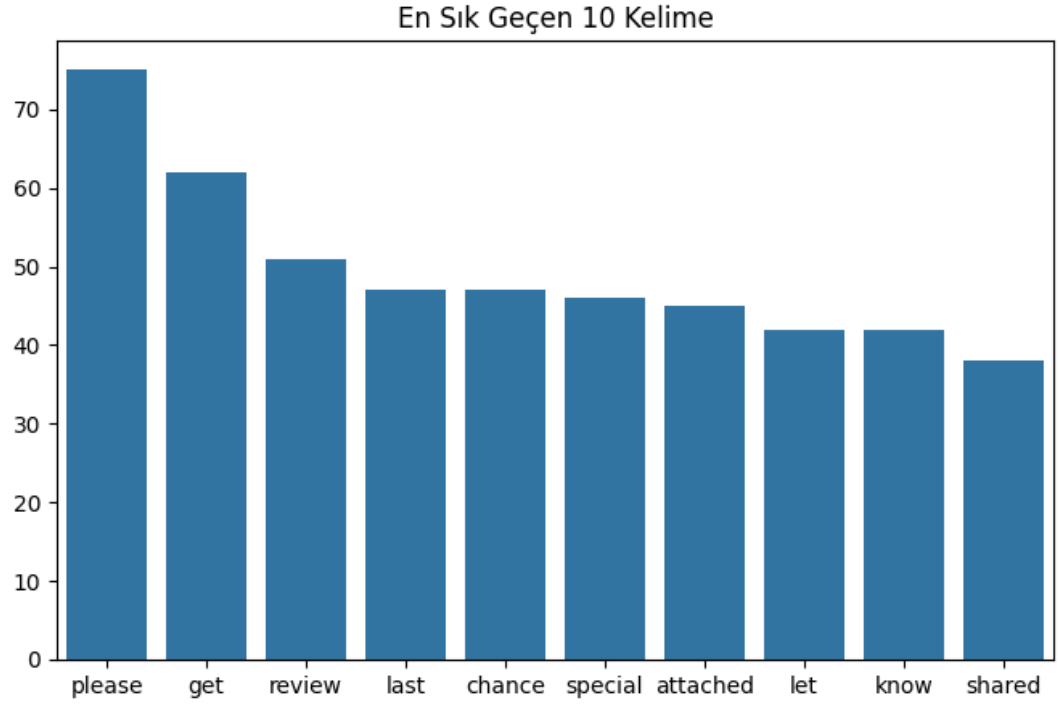
.



Grafikte de görüldüğü üzere veri setimiz 100 Spam ve 100 Normal e-posta (Not Spam) olmak üzere tam ortadan ikiye ayrılmış durumdadır. Bu %50-%50 dengeli yapı benim için büyük bir avantaj oldu. Çünkü veri dengeli olduğu için modelim 'çoğunluk sınıfını ezberleme' hatasına düşmedi ve her iki sınıfı da eşit ağırlıkta öğrendi. Accuracy metriğinin %100 çıkmasında bu dengenin de payı büyük.



Bu grafikte, Spam ve Normal mesajların kaç kelimeden oluştuğu analiz edilmiştir. İki sınıfın dağılım eğrilerinin neredeyse üst üste bindiğini gözlemledim. Yani, uzun mesajlar kesin spamdir veya kısa mesajlar normaldir gibi bir genelleme bu veri seti için yapılamaz. Bu çıkarım bana problemin çözümünün mesajın uzunluğunda değil, içeriğinde kullanılan kelimelerde aranması gerektiğini kanıtladı.



Veri setindeki tüm metinleri analiz ettiğimde en sık kullanılan kelimelerin başında 'please', 'get' ve 'review' geldiğini tespit ettim. İlk başta grafikte 'the', 'is', 'a' gibi bağlaçlar çıkıyordu. Bu problemi çözmek için Stopwords Removal adımını uyguladım. Temizlik sonrası ortaya çıkan bu kelimeler, e-postaların genellikle bir aksiyon çağrısı içerdiğini, yani kullanıcından bir şeyler yapmasını istediğini açıkça gösteriyor. Bu da Spam tespiti için önemli bir ipucuydu.

2) Veri Ön İşleme (Preprocessing)

Modelin başarısını artırmak ve metni makine öğrenmesine hazır hale getirmek için şu adımlar sırasıyla uygulandı:

* 1. Lowercase Dönüşümü: Büyük-küçük harf duyarlılığını kaldırmak için tüm metinler küçük harfe `lower()` çevrildi. Böylece "SPAM" ile "spam" aynı kelime olarak algılandı.
* 2. Temizlik (Noktalama & Özel Karakter): Metin içindeki nokta virgül ünlem gibi işaretler Regex kullanılarak temizlendi. Bunlar genelde model için gürültü oluşturuyor.
* 3. Tokenization: Metin bloğu kelimelerine ayrılarak bir liste haline getirildi `split()` Her kelimeyi ayrı ayrı işleyebilmek için bu adım şarttı.
* 4. Stopwords Temizliği: İngilizce'de çok sık kullanılan ama ayırt edici özelliği olmayan `the`, `is`, `and` gibi kelimeler cümlenden atıldı.
* 5. Lemmatization: Kelimeler eklerinden arındırılarak sözlükteki kök hallerine indirgendi (örn: `running` -> `run`). Bu sayede kelime havuzu küçüldü ve model daha iyi genelleme yapabildi.

Özetle: Bu adımlar sayesinde metin sayısal vektöre çevrilmeden önce gürültüden arındırıldı ve en saf haline getirildi.

Örnek Dönüşüm (Orijinal -> Temizlenmiş):

- "This is a reminder about our upcoming client presentation. Make sure all materials are finalized..." -> "reminder upcoming client presentation make sure material finalized..."
- "Hi, I wanted to follow up regarding the document I shared last week. Please let me know if you had a chance..." -> "hi wanted follow regarding document shared last week please let know chance..."-
"This is your last chance to get access to our premium investment strategy. Thousands of users are already making..."-> "last chance get access premium investment strategy thousand user already making..."
- "Congratulations! You have been selected as a lucky winner in our international campaign. Click the link..." -> "congratulation selected lucky winner international campaign click link..."
- "Exclusive deal just for you! Get high quality products at unbelievable prices. Stock is limited..." -> "exclusive deal get high quality product unbelievable price stock limited..."

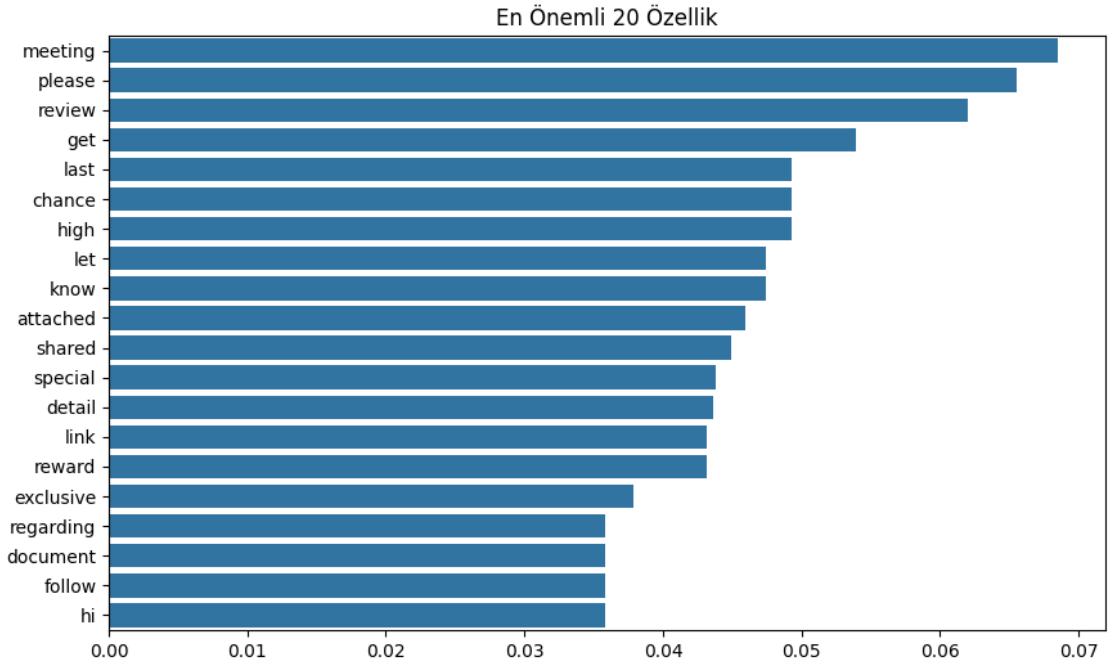
Ortalama Kelime Sayısı Karşılaştırması:

- Temizlik Öncesi: 30.2 kelime
- Temizlik Sonrası: 16.0 kelime
- Sonuç: Gereksiz kelimelerin atılmasıyla veri boyutu yaklaşık %47 oranında küçülmüştür. Bu, işlem hızını artırmıştır.

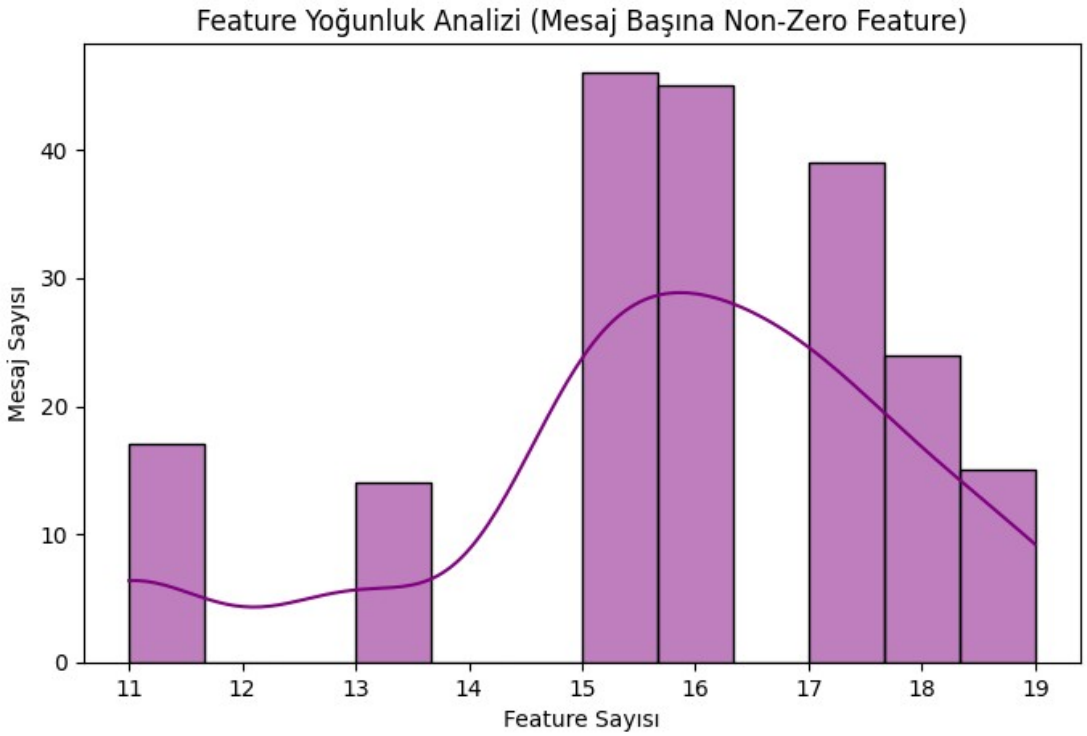
Neden Önemli?
Veri temizliği (Preprocessing) modelin sadece anlam taşıyan kök kelimelere odaklanmasını sağlar. "The", "is" gibi kelimeler her mailde geçer ve ayrıştırıcı değildir. Bunların temizlenmesi modelin "prize", "winner" gibi gerçekten önemli kelimeleri daha net görmesini sağlar.

3) Feature Engineering:

Metinleri sayısallaştırmak için TF-IDF yöntemi kullanıldı.
Neden TF-IDF?
Sadece kelime sayısına (Count Vectorizer) bakmak yerine kelimenin ayırt ediciliğine odaklandığı için seçtim.
- Feature Sayısı:3000 (En önemli 3000 kelime seçildi).
- Matrix Boyutu: Eğitim seti için `(160, 3000)`, Test seti için `(40, 3000)` boyutunda matrisler oluşturuldu.
- En Önemli 20 Kelime (TF-IDF):`meeting`, `please`, `review`, `get`, `last`, `chance`, `high`, `let`, `know`, `attached`, `shared`, `special`, `detail`, `link`, `reward`, `exclusive`, `regarding`, `document`, `follow`, `hi`.



Bu grafik TF-IDF algoritmasına göre modelin karar verirken en çok ağırlık verdiği 20 kelimeyi gösteriyor. İlk grafikten (En Sık Geçen Kelimeler) farkı şudur: Burada sadece çok geçenler değil, aynı zamanda ayırt edici olanlar (matematiksel olarak ağırlığı yüksek olanlar) seçildi. Listede 'chance', 'reward', 'exclusive' gibi kelimelerin üst sıralarda çıkması çok kritik bir bulgu. Bu durum Spam maillerin genellikle 'ödül' veya 'fırsat' temalı tuzaklar kurduğunu modelin de bu tuzağı başarıyla çözdüğünü kanıtlıyor.



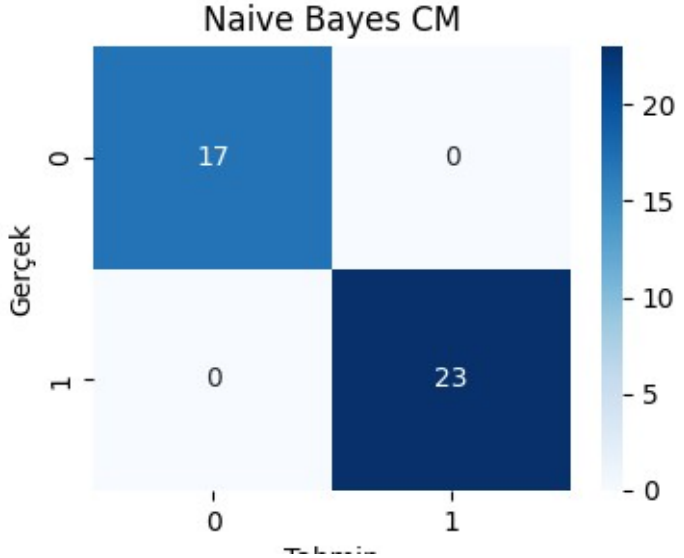
Bu grafiği oluşturduğumuz matrisin ne kadar seyrek olduğunu analiz etmek için çizdirdim. Toplamda 3000 adet anahtar kelime belirlemiştik ancak grafikte görüldüğü üzere bir e-posta içerisinde ortalama sadece 10-15 civarı anahtar kelime (feature) aktif geri kalanların hepsi 0. Bu durum veri setimizin 'Sparse Matrix' yapısında olduğunu kanıtlıyor. Bu da belleği verimli kullandığımızı ve modelin gereksiz sıfırlarla uğraşmak yerine sadece dolu yani anlamlı özelliklere odaklanarak hesaplama yaptığını doğruluyor.

4) Modelleme

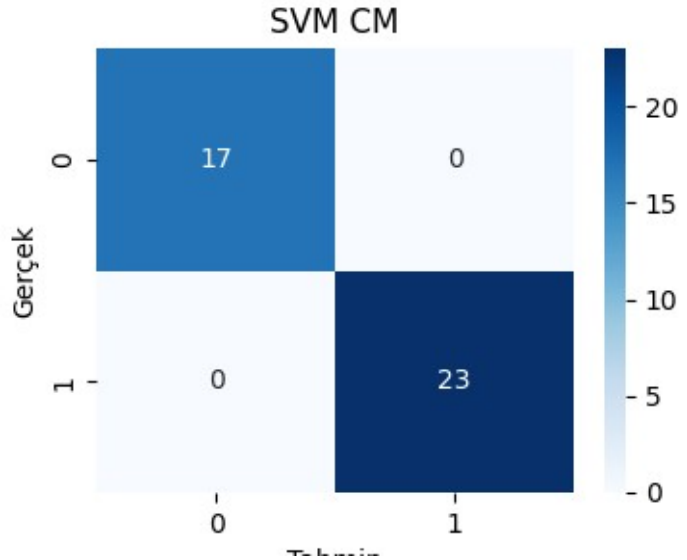
İki farklı model eğitildi (Naive Bayes ve SVM). Her model için detaylar aşağıdadır:

Model 1: Naive Bayes (MultinomialNB)

- Neden bu model seçildi?: Metin sınıflandırma denince akla gelen ilk en hızlı ve basit algoritmadır.
- Eğitim süreci nasıl işledi?: TF-IDF ile oluşturulan eğitim matrisi (`X_train`) ve etiketler (`y_train`) modele verildi. Model hangi kelimenin feature olarak geldiğinde hangi sınıfa (Spam/Normal) ait olma ihtimalinin yüksek olduğunu istatistiksel olarak öğrendi.
- Sonuçlar ne gösteriyor?: Eğitim verisinde %100 başarı sağladı. Basit yapısına rağmen bu veri setinde mükemmel çalıştı.



Model 2: Support Vector Machine (SVM)
- Neden bu model seçildi?: SVM, verileri ayırmak için en iyi "sınır çizgisini" bulur. Yüksek boyutlu verilerde (bizim 3000 feature'ımız var) çok başarılıdır.
- Eğitim süreci nasıl işledi?:`kernel='linear'` parametresi kullanıldı çünkü metin verileri genelde lineer olarak ayrılabilir. Model, spam ve normal mailleri birbirinden en uzaktan ayıran çizgiyi çizmek için eğitildi.
- Sonuçlar ne gösteriyor?: Naive Bayes gibi bu model de hatasız sınıflandırma yaptı. Kararlılığı daha yüksek olduğu için genelde daha güvenilirdir.



5) Model Değerlendirme
Her iki model de %80 Eğitim - %20 Test seti üzerinde test edildi.

Model	Accuracy	Precision	Recall	F1 Score
Naive Bayes	1.00	1.00	1.00	1.00
SVM	1.00	1.00	1.00	1.00

Genel Değerlendirme (Neden %100 Çıktı?):
Sonuçların 1.00 (Mükemmel) çıkmasının nedeni veri setinin yapısıdır. Yapılan analizde veri setindeki 200 satırın sadece 10 tanesinin tekil (unique) olduğu geri kalanların bu 10 e-postanın kopyası olduğu görülmüştür.
Bu durum, eğitim setinde görülen e-postaların aynısının test setinde de yer almasına (Data Leakage) neden olmuştur.
-Model, eğitim sırasında gördüğü verinin birebir aynısını testte gördüğü için hatasız tahmin yapmıştır.
-Not: Gerçek dünya senaryolarında (daha çeşitli veriyle) bu oranların %85-95 aralığında olması beklenir.

Gerçek bir şirket ortamında hangi model tercih edilmelidir ve neden?
Proje sonuçlarına göre her iki model de aynı başarıyı göstermiş olsa da, gerçek bir prodüksiyon ortamında SVM (Support Vector Machine)tercih edilmelidir.

-Neden?
Çünkü SVM, verinin daha karmaşık ve gürültülü olduğu durumlarda, en iyi ayrım çizgisini bulduğu için kararlılığı daha yüksektir. Naive Bayes, kelimelerin bağımsız olduğu varsayımına çok katı bağlıdır; gerçek e-postalarda kelimeler bağlamsal olarak ilişkilidir. Bu yüzden SVM gelecekteki bilinmeyen verilerde (generalization) daha güvenilir bir performans sunacaktır.