BIL 470 Final Raporu

Berkay Yıldız  
Bilgisayar Mühendisliği Bölümü  
TOBB Ekonomi ve Teknoloji Üniversitesiberkay.yildiz@etu.edu.tr

*Özet*—Bu projede amacımız, çeşitli makine öğrenme metodları kullanarak kişilerin H1N1 ve mevsimsel grip aşısı olup olmayacaklarını sahip olduğumuz bilgilerle tahmin etmeye çalışmaktır.

Anahtar Kelimeler—makine öğrenmesi, aşı, toplum sağlığı

# Giriş

Bu projede amacımız bireylerin paylaşmış olduğu bilgilerden faydalanarak H1N1 ve mevsimsel grip aşısını olup olmayacağını tahmin etmektir. 36 öznitelikten ve 26706 veriden oluşan bu veri seti, 2009 yılında yapılan ulusal H1N1 grip araştırmasından alınmıştır.

Geçtiğimiz pandemi günlerinde karşılaştığım aşı karşıtları ve aşı taraftarları kişilerin tartışmalarına şahitlik ettikten sonra böyle bir projeyi görünce yapmak istedim ve bu projenin de devam etmekte olan bir yarışma olması beni daha çok motive etti. Yarışma linki:

<https://www.drivendata.org/competitions/66/flu-shot-learning/page/210/>

Problemin çıktı sonucu aşı yaptırmak ve aşı yaptırmamak olmak üzere 2 farklı sonuç olabileceğinden sınıflandırma problemidir.

# Veri Seti, Veri Özellikleri, Veri Öznitelikleri

Veri kaynağı:

https://www.drivendata.org/competitions/66/flu-shot-learning/data/

## Veri Setindeki Öznitelikler ve Tanımları

* h1n1\_vaccine: H1N1 grip aşısı olup olmadığı bilgisi. (Çıktı değer)
* seasonal\_vaccine - Mevsimsel grip aşısı olup olmadığı bilgisi. (Çıktı değer)
* h1n1\_concern: H1N1 virüsünden korkma seviyesi

0 = Neredeyse hiç; 1 = Çok endişeli değil;

2 = Biraz endişeli; 3 = Çok endişeli.

* h1n1\_knowledge – H1N1 gribi hakkında bilgi seviyesi

0 = Bilgisiz; 1 = Az bilgili; 2 = Çok bilgili.

* behavioral\_antiviral\_meds: Antiviral ilaç kullandı (binary)
* behavioral\_avoidance: Grip tarzı belirtileri olan kişilerle yakın temastan kaçındı (binary)
* behavioral\_face\_mask: Yüz maskesi aldı (binary)
* behavioral\_wash\_hands: Elini sıkça yıkıyor veya dezenfektan kullanıyor (binary)
* behavioral\_large\_gatherings: Çok insanlı buluşmalarda az zaman geçirdi(binary)
* behavioral\_outside\_home: Ev halkı dışı insanlarla az zaman geçirdi (binary)
* behavioral\_touch\_face: Göz, ağız, buruna dokunmaktan kaçınıyor (binary)
* doctor\_recc\_h1n1: H1N1 grip aşısı doktor tarafından tavsiye edildi (binary)
* doctor\_recc\_seasonal: Mevsimsel grip aşısı doktor tarafından tavsiye edildi. (binary)
* chronic\_med\_condition: Yazanlardan herhangi birisine sahip: astım veya herhangi bir akciğer hastalığı, diyabet, kalp hastalığı, böbrek rahatsızlığı, orak hücreli anemi veya diğer anemi, kronik bir hastalık veya kronik bir hastalık için alınan ilaçlar nedeniyle nörolojik veya nöromüsküler bir durum, bir karaciğer rahatsızlığı veya zayıflamış bir bağışıklık sistemi. (binary)
* child\_under\_6\_months: Altı aylıktan küçük bir çocukla düzenli yakın temas halinde. (binary)
* health\_worker: Sağlık çalışanı. (binary)
* health\_insurance: Sağlık sigortası var. (binary)
* opinion\_h1n1\_vacc\_effective: H1N1 aşısının etkinliği hakkındaki görüşü.

1 = Hiç etkili değil; 2 = Çok etkili değil;

3 = Bilmiyor; 4 = Biraz etkili; 5 = Çok etkili.

* opinion\_h1n1\_risk: Aşısız H1N1 gribine yakalanma riski hakkındaki görüşü

1 = Çok düşük; 2 = Biraz düşük; 3 = Bilmiyor;

4 = Biraz yüksek; 5 = Çok yüksek.

* opinion\_h1n1\_sick\_from\_vacc: H1N1 aşısı almaktan hastalanma endişesi.

1 = Hiç endişeli değil; 2 = Çok endişeli değil;

3 Bilmiyor; 4 = Biraz endişeli; 5 = Çok endişeli.

* opinion\_seas\_vacc\_effective: Mevsimsel grip aşısının ne kadar etkili olduğu hakkındaki düşüncesi.

1 = Hiç etkili değil; 2 = Çok etkili değil;

3 = Bilmiyor; 4 = Biraz etkili; 5 = Çok etkili.

* opinion\_seas\_risk: Aşısız mevsimsel grip hastalığına yakalanma riski hakkındaki görüşü.

1 = Hiç etkili değil; 2 = Çok etkili değil;

3 = Bilmiyor; 4 = Biraz etkili; 5 = Çok etkili.

* opinion\_seas\_sick\_from\_vacc: Mevsimsel grip aşısı yapmaktan hastalanma endişesi.

1 = Hiç endişeli değil; 2 = Çok endişeli değil;

3 Bilmiyor; 4 = Biraz endişeli; 5 = Çok endişeli.

* age\_group: Yaş grubu.
* Education: Bildirdiği eğitim seviyesi.
* Race: Irk.
* Sex: Cinsiyet.
* income\_poverty: 2008 Nüfus Sayımı yoksulluk eşiklerine göre katılımcının hane yıllık geliri.
* marital\_status: medeni durumu.
* rent\_or\_own Barınma durumu.
* employment\_status: İşsizlik durumu
* hhs\_geo\_region: Davalının ABD Sağlık ve İnsan Hizmetleri Departmanı tarafından tanımlanan 10 bölgeli bir coğrafi sınıflandırmayı kullanan ikametgahı. Değerler, kısa rastgele karakter dizileri olarak temsil edilir.
* census\_msa: Davalının ABD Nüfus Sayımı tarafından tanımlanan metropol istatistik alanlarında (MSA) ikametgahı.
* household\_adults: Hanedeki diğer yetişkinlerin sayısı.
* household\_children Hanedeki çocuk sayısı.
* employment\_industry: Sektördeki katılımcının türü istihdam edilmektedir. Değerler, kısa rastgele karakter dizileri olarak temsil edilir.
* employment\_occupation: Katılımcının meslek türü. Değerler, kısa rastgele karakter dizileri olarak temsil edilir.

# Onişleme Aşamaları

Kullandığımız veri setini makine öğrenmesi için direk bu şekilde kullanamayız. Çünkü veri setinde makine öğrenmesi için kullanılamayacak değerler var. Bunu makine öğrenmesi için uygun hale getirmeliyiz.

## Null Değerleri Tamamlama

Bunun için 2 method kullandım.

#### Silme: Bazı özniteliklerde boş değerler verilerin yarısından fazlasını oluşturuyordu. Elimde yeterli sayıda öznitelik olmasının sayesinde bahsedilen özniteliklerin silinmesinin daha doğru olabileceğini araştırmalarım sonucu gördüm.

#### Uygun Değer Yerleştirme: Bazı verilerimde ise null değerler için mod, medyan gibi matematik operatörleri kullandım.

## Numerik Dataya Çevirme

Veri setindeki bazı özniteliklerin verileri String(yazı) türü gibi numerik olmayan verilere sahipti. Bu değerler bu şekli ile makine öğrenmesi için kullanılamaz fakat uygun formata çevrilebilir. Bu gibi durumlar için kullanılan 2 farklı yaklaşımı birkaç farklı şekilde denedim.

İlk olarak birlikte kullanmaya çalıştım. Bazılarında One-Hot Encoding, bazılarındaysa Label encoding kullandım. İkinci yol olarak bütün kategorik veriyi One-Hot encoding ile ve son olarak ise kategorik veriye sahip özniteliklerin tamamını Label encoding ile kullandım.

Sonuçlar arasında en iyisi tamamen Label encoding kullandığım method olduğundan, sadece Label encoding kullanmaya karar verdim.

Feature Selection da denedim fakat sonuçlarımı çok olumsuz etkilediği için bunu daha sonradan bunu da kullanmayı bıraktım.

#### One-Hot Encoding: Bu yaklaşım kategorik değişkenlerin binary olarak temsili olarak kullanılmasıdır. Bunun için ilk başta verilerin bir numerik tamsayıya eşitlenmesi gerekir ve daha sonrasında binary vektör şeklinde gösterilir.

#### Label Encoding: Veriyi direk olarak sayısallaştırmada kullanılır. Her kategorik veri için farklı bir sayısal değer kullanılır.

## Normalizing

Veri setindeki değerleri uygun oranlamayla genellikle 0 ve 1 arasında sıkıştırma yöntemidir. Makine büyük sayılarla uğraşmayacağı için öğrenmenin daha hızlı olmasına yardımcı olur. Ayrıca büyük sayılar ve küçük sayılar arasındaki oranı daha mantıklı kurmasını sağlar.

# Kullanılan Modeller

Makine öğrenmesi için birçok sayıda model mevcut. Bunlardan bazıları bu proje için kullanmaya uygun gördüm. Bu modelleri ve neden kullandığımı madde madde açıklayacağım. Ayrıca test\_size değerini 0.1 olarak belirledim.

## SVM(Support-Vector Machine)

Sınıflandırma problemleri için kullanılabilen bu model, projedeki problem için kullanılabilir bir model. Bu projede öznitelikler fazla olduğundan SVM hiper düzlem ile noktaları ayrıştırarak sınıflandırmaktadır.

## Logistic Regression

Sınıflandırma yöntemleri arasında bulunan lojistik regresyon sonucunda iki sonuç çıkabilir yani binary’dir. Bu modelin amacı, bağımlı ve bağımsız değişkenlerin arasındaki ilişkiyi tanımlamak için en uygun modeli bulmaktadır. Bu projede de birçok özniteilk ve sonuç olarak 2 çıktı olduğundan bu modeli kullanmayı uygun gördüm.

## XGBoost

Bu modelde Gradient Boostingde yapıldığı gibi hataları tahminleyen karar ağacı kurulur. Diğer algoritmalardan daha hızlı çalışmasıyla avantaj sağlasa da bu modelin küçük veri setli problemler için uygun değildir. XGBoost modeli genellikle tek bir karar ağacından daha yüksek doğruluk elde ederken, karar ağaçlarının anlaşılabilir olma özelliğinden vazgeçerek bunu yapar Hem performans hem de yorumlanabilirlik elde etmek için, bazı model sıkıştırma teknikleri kullanılmaktadır..

## LightGBM

Microsoft tarafından geliştirilen bu modelde gradient boosting yöntemi kullanılmaktadır. Durumlara göre XGBoosttan daha hızlı çalışabilmektedir ve accuracy’si de daha yüksek çıkabilmektedir. Ayrıca LightGBM, sıralanmış özellik değerleri üzerinde en iyi ayırma noktasını arayan, yaygın olarak kullanılan sıralı tabanlı karar ağacı öğrenme algoritmasını kullanmaz, Bunun yerine LightGBM, hem verimlilik hem de bellek tüketiminde büyük avantajlar sağlayan, yüksek düzeyde optimize edilmiş bir histogram tabanlı karar ağacı öğrenme algoritması uygular. Ağaç yapısını sevdiğim için bu modeli de elimdeki veri setiyle denemek istedim.

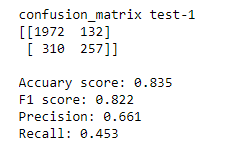
## Multilayer Perceptron

MLP terimi belirsiz bir şekilde, bazen gevşek bir şekilde herhangi bir ileri beslemeli YSA anlamına gelir, bazen de kesin olarak birden çok algılayıcı katmanından oluşan ağları ifade etmek için kullanılır. Bir MLP, en az üç düğüm katmanından oluşur: bir giriş katmanı, bir gizli katman ve bir çıkış katmanı. Giriş düğümleri dışında, her düğüm, doğrusal olmayan bir aktivasyon işlevi kullanan bir nörondur. 1 tane de derin öğrenme modeli kullanmak istediğim için ve multilayer olayı da ilgimi çektiğinden bu modeli denemek istedim.

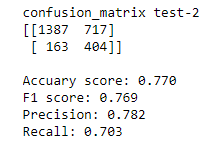
# Test Sonuçları

Sınıflandırma problem içersinide sayılabilecek kimin aşı olup olmacağını tahmin etme probleminde farklı modeller kulandım ve bunun sonuçlarını F1-score ve karmaşıklık matrisi ile inceledim. 2 farklı labelim olduğundan ikisi içinde aynı modelden 2 tane aynı işlevde model yaratarak ayrı ayrı eğittim ve sonuçlarını karşılaştırdım. Bu şekilde tek bir sonuca bağlı kalmaktansa kullandığım modelin ne kadar tutarlı olduğunu da görmüş oldum. Fakat elimde train, test olmak üzere bunlardan ikişer tane olduğu için burada sadece test sonuçlarını paylaşacağım.

## SVM(Supoort-Vector Machine)

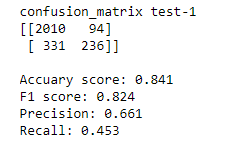


**Şekil 1**. SVM modeli Test-1

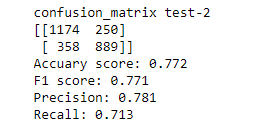


**Şekil 2**. SVM modeli Test-2

## Logistic Regression

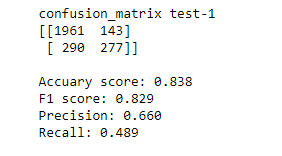


**Şekil 1**. Logistic Regression modeli Test-1

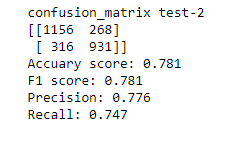


**Şekil 1**. Logistic Regression modeli Test-2

## XGBoost

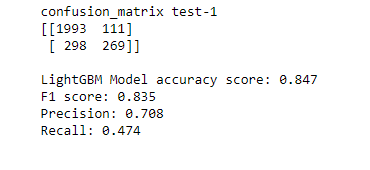


**Şekil 1**. XGBoost modeli Test-1

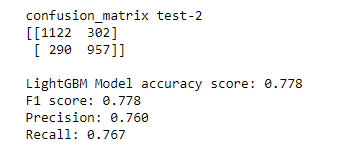


**Şekil 1**. XGBoost modeli Test-2

## LightGBM

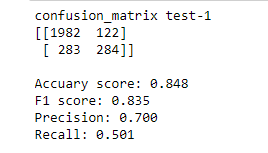


**Şekil 1**. LightGBM modeli Test-1

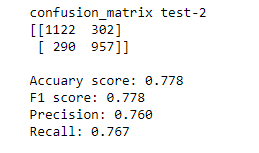


**Şekil 1**. LightGBM modeli Test-2

## Multilayer Perceptron



**Şekil 1**. Multilayer Perceptron modeli Test-1



**Şekil 1**. Multilayer Perceptron modeli Test-2

(Burada f1 scoreların ortalaması ve accuracy scoreların ortalamalarını karşılaştırdım fakat tablonun boyutunun uygun olmamasından dolayı ekleyemiyorum. Jupiter notebook dosyasının en altında görebilirsiniz.)

# Sonuçlar

Bu çalışmada sahip olduğum veri setiyle insanların aşı olup olmayacağını tahminlemek için makine öğrenmesi modelleri uyguladım. Bu modellerin bazılarını derste öğrendim bazılarınıysa internetten araştırıp modelime uygulanabilecek olanları seçtim.

Makine öğrenmesi modellerimi eğitmek için öncelikle veri setimi uygun hale getirmeliydim. Feature selection yöntemiyle ilk başta uygun olmadığını düşündüğüm öznitelikleri belirledim ve veri setimden çıkarttım. Daha sonra veri setimdeki null valueları tespit edip bunları uygun yöntemler ile doldurdum. Mean, mod, medyan bu yöntemler arasındadır.

Makine öğrenmesinin öğrenmeye uygun olabilmesi için veri setindeki özniteliklerin datatype’ı numerik olmalıdır. Benim veri setimde numerik harici verilerde olduğundan bunları uygun yöntemlerle numerik şekle getirdim. İlk başta bütün uygun olmayan verileri One-Hot-Top Encoding ile doldurdum, ikinci olarak bütün uygunsuz verileri Label-Encoding ile doldurdum ve son olarak ise 2sini birlikte kullandım ve eğittim modellerde ayrı ayrı sonuçlarına baktım. En iyi sonucu tamamen Label-Encoding kullandığım metodda aldığım için verisetimde bu yöntemi kullanmayı tercih ettim.

Yapay öğrenme için 4 tane makine öğrenmesi bir tane de derin öğrenme kullandım. Yapay öğrenme metodlarını çalıştırıp f1-score, accuracy ve confusion matrix’lerini elde ettim ve sonuçlarını karşılaştırdım.

Tüm bunları yaparken teorikte öğrenmiş olduğum bilgileri pratiktede görmüş oldum. Bazı şeylerin bana mantıklı gelmesinin makine öğrenmesi için doğru olamayabileceğini farkettim. Örneğin ben One-Hot-Top Encoding ile Label-Encoding’i birilkte kullanmanın daha iyi olacağını düşünürken öyle olmadı. Derin öğrenme modellerini araştırdım. Modelimi normalizasyon çalıştırırken yaklaşık 1 saat zaman alırken normalizasyon ile birlikte yaklaşık 20 saniyede makine öğernince normalizasyon yapmanın önemini bir kere daha farkettim. Gelecek çalışmalarımda bu çalışmada yaparak öğrendiğim yanlışları tekrar yapmamak benim için zaman kazandırıcı olacaktır. Ayrıca bu gibi çalışmaların gerçek hayatta ne kadar uygulanabilir olduğumu tekrar farketmem sonucu bu alana ilgim olduğunu farketmemde yardımcı oldu.

Bu modeli geliştirmek için daha fazla veriye sahip olmak derin öğrenmeye de yardımcı olacağı gibi diğer modellerin gelişimine de katkı sağlar. Modellerin az kullandığı öznitelikler yerine daha uygun veriler elde edilip bu öznitelikler ile değiştirilmesi modelin geliştilmesi için iyi olabiliceğini düşünüyorum.

##### References

1. https://medium.com/deep-learning-turkiye/nedir-bu-destek-vektör-makineleri-makine-öğrenmesi-serisi-2-94e576e4223e *(references)*
2. https://veribilimcisi.com/2017/07/18/lojistik-regresyon/.
3. https://machinelearningmastery.com/how-to-one-hot-encode-sequence-data-in-python/.
4. https://www.geeksforgeeks.org/ml-one-hot-encoding-of-datasets-in-python/.
5. https://analyticsindiamag.com/5-ways-handle-missing-values-machine-learning-datasets/
6. <https://www.veribilimiokulu.com/xgboost-nasil-calisir/>
7. <https://machinelearninggeek.com/multi-layer-perceptron-neural-network-using-python/>
8. https://www.kaggle.com/code/prashant111/lightgbm-classifier-in-python/notebook