Gen İfadesi Veri Seti Üzerinde Makine Öğrenmesi Yöntemleriyle Sınıflandırma

Proje Raporu

Öğrenci No / Adı: Musap Yıldız (Numara: 20011083)

Öğrenci No / Adı: Ubeyde Alnaccar (Numara: 24011937)

Danışman: Prof. Dr. Nizamettin Aydın

Ders: Biyoinformatiğe Giriş

Tarih: June 4, 2025

YILDIZ TEKNİK ÜNİVERSİTESİ

Abstract

Bu proje raporunda, *clear cell renal cell carcinoma* (CCRCC) hastalığına ait gen ifadesi verilerinin makine öğrenmesi yöntemleri ile sınıflandırılması incelenmiştir. **GEO** veri tabanından `GSE53757` serisi kullanılarak normal böbrek dokusu ile tümör dokusu örnekleri ayrıştırılmıştır. Veri ön işleme adımları ve varyans bazlı öznitelik seçimi ile en yüksek varyansa sahip n gen belirlenmiş, ardından üç farklı sınıflandırıcı (Lojistik Regresyon, Rastgele Orman ve Destek Vektör Makinesi) eğitilip değerlendirilmiştir. Hiperparametre optimizasyonu için `GridSearchCV` kullanılmıştır. Model performansları, stratified 5-fold çapraz doğrulama ile doğruluk, precision, recall ve F1-score metrikleri üzerinden karşılaştırılmıştır. En yüksek genel başarı $\mathbf{n = 10}$ gen ile Lojistik Regresyon modelinde elde edilmiş, accuracy ortalaması 0.9724 ± 0.0378 olarak bulunmuştur. Ortalama confusion matrix değerleri de raporda yer almaktadır.

Contents

1 Giriş

Böbrek kanserleri içinde *clear cell renal cell carcinoma* (CCRCC), en sık görülen alt tiplerden birisidir. Hastalığın erken evrelerinde klinik belirti vermemesi nedeniyle tanı genellikle ileri aşamalardadır. Günümüzde moleküler biyoloji yaklaşımları sayesinde *in silico* analizler ve makine öğrenmesi yöntemleri, kanser dokusu ile sağlıklı dokunun ayrımında önemli rol oynamaktadır. Bu projede, yüksek boyutlu gen ifadesi verilerini kullanarak normal böbrek dokusu ile CCRCC dokusunu sınıflandırmayı amaçladım.

Projenin ana hedefleri:

- **Veri edinme ve ön işleme:** `GEO` adresinden `GSE53757` serisi elde edildi. Satır/kolon formatına getirilen veri seti üzerinde eksik değer kontrolü, normalizasyon ve varyans analizi uygulandı.
- **Öznitelik seçimi:** Tüm gende varyans hesabı yapıldı ve her katmanda varyans bazlı en yüksek n gen seçildi. İlerleyen aşamalarda farklı n değerleri (5, 10, 20, 50, 100) test edildi.
- **Modelleme ve karşılaştırma:** Üç farklı sınıflandırıcı (Lojistik Regresyon, Rastgele Orman, Destek Vektör Makinesi) kullanıldı. Hiperparametre optimizasyonu `GridSearchCV` ile yapıldı. Model performansları 5-fold stratified çapraz doğrulama ile değerlendirildi.
- **Analiz ve yorum:** Accuracy, precision, recall ve F1-score metriklerine göre modellerin başarısı karşılaştırıldı. Ayrıca her modelin ortalama confusion matrix değerleri raporda sunuldu.

Bir sonraki bölümde kullanılan veri seti ve ön işleme adımları ayrıntılı şekilde tanımlanacaktır.

2 Veri Seti ve Ön İşleme

2.1 Veri Setinin Kaynağı

CCRCC hastalığına ait gen ifadesi verileri, *GEO* (Gene Expression Omnibus) platformundan **GSE53757** dizin numarasıyla elde edildi. Bu seri, *Affymetrix Human Genome U133 Plus 2.0* mikroarray platformu kullanılarak ölçülen örnekleri içerir. Toplamda 144 örnek vardır:

- 72 normal böbrek dokusu (`normal kidney`)
- 72 CCRCC dokusu (`clear cell renal cell carcinoma`)

Her örneğe karşılık 54 675 adet probe (gen) ölçümü bulunmaktadır. Veri ham hali bir `.txt` dosyasında satır meta-verileriyle birlikte yer alır. Kod ile meta-veri satırlarını atlayıp ifade matrisine ulaşıldı.

2.2 Verinin Yüklenmesi ve Düzenlenmesi

1. `load_series_matrix(path)` fonksiyonu:
   - `pd.read_csv(..., comment = "!")` parametresiyle "!" ile başlayan meta-veri satırlarını atladım.
   - `index_col=0` ile ilk sütunu (probe ID) satır indeksi olarak kullandım.
   - Okuma sonrasında elde edilen DataFrame, (54675×144) boyutlarındadır.
   - Kodda bu satırları tekrar örnek $\times$ gen matrisine döndürmek için `expr_df.T` kullandım: $\mathbf{X} \in R^{144\times 54675}$.

2. `extract_labels_by_field(path, "tissue")` fonksiyonu:

   - "!Sample_characteristics_ch1" ile başlayan her satır incelenip "tissue:" ifadesi arandı.
   - Satırdan çıkarılan tırnaklı metinler `"clear cell renal cell carcinoma"` veya `"normal kidney"` şeklinde geldi.
   - Bu etiketleri büyük-küçük harf duyarsız şekilde kontrol ettim, içinde "carcinoma" geçenleri pozitif (1), diğerlerini negatif (0) olarak kodladım.
   - Sonuç olarak $y \in R^{144}$ boyutlu bir ikili vektör elde edildi.

2.3 Önişleme (Preprocessing)

- **Eksik Veriler:** Sağlanan `series_matrix.txt` dosyasında eksik gözlem bulunmuyordu, bu adımı atladım.
- **Normalizasyon:** Mikroarray verileri zaten RMA (Robust Multi-array Average) veya benzeri metotlarla normalize edilmişti. Buna ek olarak, modellerde `StandardScaler` kullanarak *ortalama* 0*, standart sapma* 1 biçimine getirdim.
  - Bu adım, özellikle Lojistik Regresyon ve SVM için önemlidir; çünkü farklı genler arasındaki değer aralıkları önemli farklılıklar barındırabilir.

2.4 Öznitelik Seçimi (Feature Selection)

Yüksek boyutlu biyolojik verilerde $p \gg n$ (probe sayısı $\gg$ örnek sayısı) durumu aşırı uyum (overfitting) riskini artırır. Bu nedenle, her CV katmanında sadece **en yüksek varyanslı** N **gene** odaklanmak mantıklı bir yoldur:

- Eğitim seti (X_{train}) üzerinde her genin varyansı hesaplandı: $\text{Var}(X_{\text{train}}[:, j])$.
- Varyans değerlerine göre azalan sırada N gen seçildi.
- Bu seçilen N gen, hem eğitim hem test setlerine uygulanarak veri boyutu N'ye indirildi.
- Bu işlem modeli daha az boyutlu bir uzayda eğitmek, gürültüyü azaltmak ve genellenebilirliği artırmak amacıyla yapıldı.

Test edilen N değerleri: {5, 10, 20, 50, 100}. Kod akışı:

1. `variances = X_train.var(axis=0)`
2. `topN = variances.sort_values(ascending=False).head(N).index`
3. `X_train_N = X_train[topN], X_test_N = X_test[topN]`

3 Yöntemler

Bu bölümde, kullandığım sınıflandırma yöntemlerinin algoritma mantığını, hiperparametre ayarlarını ve çapraz doğrulama stratejisini açıklıyorum.

3.1 Modeller ve Hiperparametre Ayarları

Üç farklı makine öğrenmesi yöntemi tercih edildi. Kodda pipeline hâlinde tanımladığım modeller ve grid search ızgaraları aşağıda özetlenmiştir.

3.1.1 Lojistik Regresyon (Logistic Regression)

- Lineer bir ayırıcı (classifier) modeli:

$$P(y = 1 \mid x) = \sigma(\mathbf{w}^T\mathbf{x} + b), \quad \sigma(z) = \frac{1}{1 + e^{-z}}.$$

- L2 düzenlileme (ridge) varsayıldı.
- Hiperparametre ızgarası:

$$C \in \{0.01,\ 1,\ 100\}, \quad \text{penalty} = \text{“l2”}.$$

- Kod parçası:

```
make_pipeline(
    StandardScaler(),
    LogisticRegression(max_iter=1000, random_state=42)
)
```

- GridSearchCV parametreleri: `param_grid = {"logisticregression__C": [0.01,1,100], "logisticregression__penalty": ["l2"]}`
- Çıkış: En iyi C değeri bulunan model.

3.1.2 Rastgele Orman (Random Forest Classifier)

- Ensemble temelli, çok sayıda karar ağacı (decision tree) oluşturup, oylama (bagging) ile sınıflandırma yapan bir yöntemdir.
- Ağaç sayısı ($n\_estimators$) ve derinlik ($\max\_depth$) optimizasyonu yapıldı.
- Hiperparametre ızgarası:

$$n\_estimators \in \{50,\ 100\}, \quad \max\_depth \in \{\text{None},\ 10\}.$$

- Kod parçası:

```
make_pipeline(
    StandardScaler(),
    RandomForestClassifier(random_state=42, n_jobs=-1)
)
```

- GridSearchCV parametreleri: `param_grid = {"randomforestclassifier__n_estimators":[50,100], "randomforestclassifier__max_depth":[None,10]}`

3.1.3 Destek Vektör Makinesi (SVM, RBF Kernel)

- Temelde lineer ayırma yapabilen SVM, RBF çekirdeği ile doğrusal olarak ayrılmayan verileri de yüksek boyutlu uzayda ayırabilir:

$$K(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2).$$

- Hiperparametreler: C (penaltı katsayısı) ve γ (çekirdek parametresi).

- Hiperparametre ızgarası:

$$C \in \{0.1,\ 1\}, \quad \gamma \in \{\text{"scale"},\ \text{"auto"}\}.$$

- Kod parçası:

```
make_pipeline(
    StandardScaler(),
    SVC(kernel="rbf", probability=True, random_state=42)
)
```

- GridSearchCV parametreleri: `param_grid = {"svc__C":[0.1,1], "svc__gamma":["scale","auto"]}`

3.2 Stratified 5-Fold Çapraz Doğrulama

- Veri setimizde iki sınıf (normal vs tümör) dengeli olduğu için, her fold'da sınıf dağılımını korumak amacıyla `StratifiedKFold(n_splits=5)` kullandık.
- Her fold'da:
  1. Eğitim seti içerisinden varyans bazlı N gen seçildi.
  2. N gene indirgenmiş eğitim kümesiyle `GridSearchCV` içinde modeller eğitilip en iyi hiperparametre belirlendi (3-fold iç CV).
  3. Aynı N gen ile test kümesinde tahmin yapıldı, metrikler hesaplandı.
- Bu işlemler beş kez tekrarlandı ve each fold'dan elde edilen metriklerin ortalama ile standart sapmaları raporlandı.

4 Sonuçlar

Bu bölümde, farklı N değerleri (5, 10, 20, 50, 100) ve üç model (LR, RF, SVM) için elde edilen ortalama performans metrikleri ve confusion matrix özetleri sunulmaktadır.

4.1 Performans Metrikleri

Aşağıdaki Tablo 1 her (N, Model) ikilisi için stratified 5-fold CV sonuçlarının ortalama ve standart sapmalarını içermektedir.

Tablo 1'den öne çıkan noktalar:

- $N = 5$**:** Lojistik Regresyon (LR) ve SVM ortalama ≈ 0.9446; Random Forest (RF) ≈ 0.9305 doğruluk gösterdi.
- $N = 10$**:** LR en yüksek ortalama ≈ 0.9724 (sapma ≈ 0.0378) ile zirvedeydi. RF ≈ 0.9515, SVM ≈ 0.9446 düzeyindeydi.
- $N = 20$**:** LR $\approx 0.9655 \pm 0.0488$; RF $\approx 0.9443 \pm 0.0579$; SVM $\approx 0.9515 \pm 0.0392$.
- $N = 50$**:** LR $\approx 0.9517 \pm 0.0672$; RF $\approx 0.9515 \pm 0.0522$; SVM $\approx 0.9584 \pm 0.0449$.
- $N = 100$**:** LR $\approx 0.9515 \pm 0.0522$; RF $\approx 0.9446 \pm 0.0625$; SVM $\approx 0.9515 \pm 0.0522$.

En yüksek ortalama doğruluğu, $N = 10$ gen ile **Lojistik Regresyon** modeli gösterdi. Bu nedenle final kombinasyon önerisi:

$$N = 10 \text{ gen}, \quad \text{Model} = \text{LR (Logistic Regression)}.$$

Table 1: Farklı N ve Modeller İçin 5-Fold CV Metrikleri

N	Model	Accuracy Mean	Accuracy Std	Precision Mean	Precision Std	Recall Mean	Recall Std	F1 Mean	F1 Std
5	LR	0.9446	0.0461	0.9338	0.0446	0.9581	0.0634	0.9451	0.0471
5	RF	0.9305	0.0545	0.9442	0.0546	0.9181	0.0885	0.9290	0.0574
5	SVM	0.9446	0.0461	0.9445	0.0315	0.9448	0.0757	0.9437	0.0486
10	LR	0.9724	0.0378	0.9724	0.0379	0.9724	0.0379	0.9724	0.0379
10	RF	0.9515	0.0576	0.9567	0.0401	0.9438	0.0930	0.9490	0.0638
10	SVM	0.9446	0.0461	0.9445	0.0315	0.9448	0.0757	0.9437	0.0486
20	LR	0.9655	0.0488	0.9713	0.0395	0.9581	0.0634	0.9644	0.0512
20	RF	0.9443	0.0579	0.9450	0.0545	0.9438	0.0930	0.9426	0.0634
20	SVM	0.9515	0.0392	0.9454	0.0310	0.9581	0.0634	0.9511	0.0420
50	LR	0.9517	0.0672	0.9493	0.0778	0.9581	0.0634	0.9528	0.0648
50	RF	0.9515	0.0522	0.9463	0.0538	0.9581	0.0634	0.9515	0.0528
50	SVM	0.9584	0.0449	0.9579	0.0386	0.9581	0.0634	0.9575	0.0474
100	LR	0.9515	0.0522	0.9513	0.0739	0.9571	0.0639	0.9522	0.0514
100	RF	0.9446	0.0625	0.9360	0.0725	0.9581	0.0634	0.9459	0.0602
100	SVM	0.9515	0.0522	0.9379	0.0687	0.9714	0.0639	0.9527	0.0515

4.2 Confusion Matrix Özetleri

Her model için 5-fold CV'nin ortalama confusion matrix değerlerini Tablo 2'da görebilirsiniz. "Avg" ifadesi, beş fold'daki her bir bileşenin ortalamasını gösterir.

Table 2: Ortalama Confusion Matrix Değerleri (5-Fold CV)

N	Model	Avg TN	Avg FP	Avg FN	Avg TP
5	LR	13.4	1.0	0.6	13.8
5	RF	13.6	0.8	1.2	13.2
5	SVM	13.6	0.8	0.8	13.6
10	LR	14.0	0.4	0.4	14.0
10	RF	13.8	0.6	0.8	13.6
10	SVM	13.6	0.8	0.8	13.6
20	LR	14.0	0.4	0.6	13.8
20	RF	13.6	0.8	0.8	13.6
20	SVM	13.6	0.8	0.6	13.8
50	LR	13.6	0.8	0.6	13.8
50	RF	13.6	0.8	0.6	13.8
50	SVM	13.8	0.6	0.6	13.8
100	LR	13.6	0.8	0.6	13.8
100	RF	13.4	1.0	0.6	13.8
100	SVM	13.4	1.0	0.4	14.0

Açıklama: Beş fold'daki TN, FP, FN, TP değerleri ortalanmıştır (örnek: $N = 10$, LR için Avg TN $= 70/5 = 14$).

4.3 En İyi Kombinasyon

Özetle,

$$N = 10, \quad \text{Lojistik Regresyon}$$

kombinasyonu 5-fold CV'de **Accuracy Mean = 0.9724** $\pm$ 0.0378, **Precision Mean = 0.9724**, **Recall Mean = 0.9724** olarak öne çıkmıştır. Ortalama confusion matrix:

$$\begin{pmatrix} \text{TN} & \text{FP} \\ \text{FN} & \text{TP} \end{pmatrix} \approx \begin{pmatrix} 14.0 & 0.4 \\ 0.4 & 14.0 \end{pmatrix}$$

5 Tartışma

Elde edilen sonuçlar üzerinden önemli noktalar:

- **Öznitelik Seçimi Etkisi:**
  - $N = 5$ genle bile LR ve SVM $\approx 94.5\%$ doğruluk sağladı.
  - $N = 10$ zirveyi gördü: LR $\approx 97.2\%$; bu, seçilen 10 genin güçlü biyobelirteçler olduğunu gösterir.
  - $N = 20$, 50 ve 100'de model performansları bir miktar dalgalansa da $\approx 95\%$ bandını korudu.
- **Model Karşılaştırması:**
  - **Lojistik Regresyon** (N=10): En yüksek ortalama accuracy ve en düşük sapmayı verdi.
  - **Rastgele Orman** genel olarak $\approx 95\%$–$\approx 96\%$ doğrulukta, sapması LR'dan bir miktar daha yüksekti.
  - **SVM** RBF çekirdeğiyle 10 gen sonrası dengeli performans sergiledi (≈ 94–$\approx 95\%$).
- **Cross-Validation Sayısı:** Stratified 5-fold, test setini yeterince büyük (29 örnek) tutup sapmayı makul ($\pm$0.04–0.06) seviyede tuttu.
- **Biyoenformatik Katkı:** "Top 10 gen" listesi biyolojik olarak incelenerek CCRCC'ye özgü patolojik yollar aydınlatılabilir. Bu genler, sütür, anjiyogenez veya sinyal transdüksiyon mekanizmalarında kilit roller oynuyor olabilir.
- **Klinik Uygulama:** $N = 10$ ve LR kombinasyonu, $\approx 97\%$ doğrulukla CCRCC tanısına yardımcı olabilir. Yine de model, bağımsız bir veri seti (ör. TCGA) üzerinden validasyona ihtiyaç duyar.

6 Sonuç

Bu çalışmada, CCRCC ve normal böbrek dokularının gen ifadesi verileri üzerinde üç farklı sınıflandırıcı kullanıldı ve varyans bazlı öznitelik seçimi ile boyut indirgeme yapıldı.

Ana sonuçlar:

- **Veri Seti:** `GSE53757`, 144 örnek ve 54675 gen.

- **Özellik Seçimi:** Her fold'da varyansa göre en yüksek N gen belirlendi. Denenen değerler: {5, 10, 20, 50, 100}.
- **Modeller:** LR, RF, SVM. GridSearchCV ile hiperparametre optimizasyonu yapıldı.
- **Performans:**
  - **En iyi:** $N = 10$ LR $\rightarrow$ AccuracyMean = 0.9724 ± 0.0378, PrecisionMean = RecallMean = 0.9724.
  - Diğer kombinasyonlar ($N = 5, 20, 50, 100$) $\approx 94\%$–$\approx 96\%$ arasında doğruluk sağladı.
- **Genellenebilirlik:** Modelin hem ortalama hem de sapma değerleri, 5-fold CV'de dengeli ve makul bulundu.

Bu rapor, CCRCC tanısına destek oluşturacak biyoinformatik bir yaklaşım sunmaktadır. Gelecekte bu model, farklı veri setlerinde test edilerek ve top 10 gen biyolojik validasyona sokularak daha da güçlendirilebilir.

References

[1] NCBI GEO. *GSE53757: Comparative gene expression analysis in renal cell carcinoma.* `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE53757`

[2] Breiman, L. (2001). *Random Forests.* Machine Learning, 45(1), 5–32.

[3] Cortes, C., & Vapnik, V. (1995). *Support-Vector Networks.* Machine Learning, 20(3), 273–297.

[4] Hosmer, D. W., & Lemeshow, S. (2000). *Applied Logistic Regression*, 2nd ed. John Wiley & Sons.

[5] Pedregosa, F. *et al.* (2011). *Scikit-learn: Machine Learning in Python.* JMLR 12, 2825–2830.