



**Galatasaray Üniversitesi**  
**Veri Bilimi Tezsiz Yüksek Lisans Programı**  
**Veri Bilimi Uygulamaları Takım Projesi Ödevi**

## ÖZET

Bu projede, *synthetic\_medical\_data* veri seti kullanılarak hastaların yaşam durumunu (*Dead* = 0/1) tahmin eden bir makine öğrenmesi modeli geliştirilmiştir. Çalışmada veri temizleme, eksik değer işlemleri, kategorik değişken kodlamaları, eğitim-test bölünmesi gibi temel veri hazırlama adımları uygulanmıştır. Ardından beş farklı model eğitilmiştir:

- Logistic Regression
- Random Forest
- XGBoost
- Yapay Sinir Ağı (Neural Network)
- Explainable Boosting Machine (EBM)

Modeller hem tek bir train–test ayrimı hem de Stratified K-Fold yöntemleri ile değerlendirilmiştir. Kullanılan metrikler arasında Accuracy, Precision, Recall, F1, ROC-AUC, PR-AUC ve MCC yer almaktadır. Sonuç olarak, sınıf dengesizliğine karşı dayanıklı algoritmalar olan XGBoost ve Random Forest (SMOTE ile) en yüksek performansı göstermiş, proje sonunda en iyi model olarak XGBoost önerilmiştir.

## 1. GİRİŞ

Tıbbi verilerde mortalite tahmini, sağlık bilişiminde kritik bir makine öğrenmesi problemidir. Özellikle klinik karar destek sistemleri, risk skorlamaları ve erken müdahale sistemlerinde ölüm olasılığının tahmini önemli rol oynar. Bu projede amaç, verilen tıbbi sentetik veri üzerinde hastaların ölüp ölmeyeceğini sınıflandırmak ve farklı ML modellerinin performansını karşılaştırmaktır.

Bu proje kapsamında:

- Veri analiz adımlarının sistematik uygulanması,
- Eksik verilerin uygun istatistiksel yöntemlerle işlenmesi,
- Kategorik değişkenlerin makine öğrenmesine uygun formata dönüştürülmesi,
- Dengesiz sınıf probleminin farklı yöntemlerle ele alınması,
- Farklı algoritmaların karşılaştırılması,
- En iyi modelin belirlenmesi

amaçlanmıştır.

## 2. VERİ SETİ

### 2.1 Veri Kaynağı

Veriler "synthetic\_medical\_data.csv" dosyasından elde edilmiştir.

### 2.2 Hedef Değişken

- **Dead (0/1):** Hastanın hayatı olup olmadığı.

### 2.3 Değişken Türleri

- Sayısal değişkenler (yaş, test skorları vb.)
- Kategorik değişkenler (cinsiyet, etnik köken, eğitim seviyesi vb.)

### 2.4 Sınıf Dengesizliği

Veriler incelendiğinde sınıflar arasında dengesizlik olduğu görülmüştür. Bu durum özellikle Recall ve PR-AUC metriklerini önemli hale getirmiştir.

## 3. VERİ ÖN İŞLEME

Veri hazırlama adımları bütün projede aynı şekilde uygulanmıştır.

### 3.1 Eksik Veri İşleme

Kodlarda uygulanan işlem adımları:

#### Sayısal Değişkenler

- Eksik değerler medyan ile doldurulmuştur.

```
23 num_cols = df.select_dtypes(exclude='object').columns  
24 df[num_cols] = df[num_cols].fillna(df[num_cols].median())
```

#### Kategorik Değişkenler

- Eksik kategoriler "Missing" etiketiyle doldurulmuştur.

```
cat_cols = df.select_dtypes(include='object').columns  
df[cat_cols] = df[cat_cols].fillna("Missing")
```

## 3.2 One-Hot Encoding

Kategorik değişkenler dummy değişkenlere dönüştürülmüştür:

```
df_encoded = pd.get_dummies(df, drop_first=True)
```

## 3.3 Train-Test Ayrımı

Veri %80 eğitim – %20 test şeklinde ayrılmıştır:

```
38 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
```

Stratify parametresi sınıf dengesizliğini korumak için doğru bir seçimdir.

# 4. MODELLEME

Beş farklı model eğitilmiş ve karşılaştırılmıştır.

## 4.1 Logistic Regression

- **StandardScaler + LR Pipeline** kullanılmıştır.
- Model sınıf ağırlıklarını `class_weight='balanced'` ile ayarlamıştır.
- ROC-AUC orta seviyededir.
- Lojistik regresyon doğrusal ilişkileri yakalar ancak karmaşık ilişkilerde yetersiz kalır.

**Avantajları:** Yorumlanabilirlik yüksek.

**Dezavantajları:** Karmaşık ilişkileri yakalama gücü düşüktür.

```
#Logistic Regressionda StandardScaler (değişkenleri benzer ölçüye getirmek için) uygulaması.
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

# Pipeline içinde ölçekleme + model
pipe = Pipeline([
    ('scaler', StandardScaler()),
    ('log_reg', LogisticRegression(max_iter=2000, class_weight='balanced', random_state=42))
])

pipe.fit(X_train, y_train)
y_pred = pipe.predict(X_test)
```

## 4.2 Random Forest

İki farklı RF modeli denenmiştir:

1. **Vanilla Random Forest**
2. **SMOTE ile Random Forest**
  - SMOTE kullanımı sınıf dağılımını eşitlemiştir.
  - Ağaç temelli model olduğu için outlier'lara dayanıklıdır.
  - Performansı Logistic Regression'dan anlamlı biçimde yüksektir.

```
# Modeli tanımla
rf = RandomForestClassifier(
    n_estimators=300,
    max_depth=None,
    class_weight='balanced',
    random_state=42
)

# Eğit
rf.fit(x_train, y_train)
```

```
# 2 Model
rf_smote = RandomForestClassifier(
    n_estimators=300,
    random_state=42
)
rf_smote.fit(x_res, y_res)
```

### 4.3 XGBoost

- scale\_pos\_weight=10 ile dengesiz veri telafi edilmiştir.
- 400 ağaç, öğrenme oranı 0.05 kullanılmıştır.
- ROC-AUC değeri en yüksek çıkan modellerden biridir.
- Threshold 0.5 → 0.3/0.4'a çekilerek Recall artırılmıştır.

XGBoost'un avantajları:

- Feature interaction yakalama
- Dengesiz veride başarılı
- Aşırı öğrenmeye karşı düzenleme

Bu model, proje kapsamında en güçlü adaydır.

```
xgb = XGBClassifier(  
    n_estimators=400,  
    max_depth=5,  
    learning_rate=0.05,  
    scale_pos_weight=10, # sınıf dengesizliğini telafi eder  
    random_state=42  
)  
  
xgb.fit(X_train, y_train)  
y_pred = xgb.predict(X_test)  
y_proba = xgb.predict_proba(X_test)[:, 1]
```

### 4.4 Yapay Sinir Ağrı (Neural Network)

- 64 → 32 → 1 katmanlı bir MLP modeli kurulmuştur.
- 5-Fold Stratified K-Fold ile ortalama sonuçlar elde edilmiştir.
- Özellikle PR-AUC ve ROC-AUC değerleri iyidir.
- Ancak küçük veri setlerinde ANN modelleri regresyon ağaçlarına kıyasla daha az stabil olabilir.

```

# Yapay sinir ağı modeli
model = Sequential([
    layers.Input(shape=(x_tr.shape[1],)),
    layers.Dense(64, activation='relu'),
    layers.Dense(32, activation='relu'),
    layers.Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy',
              metrics=[tf.keras.metrics.AUC(name='auc')])

```

## 4.5 Explainable Boosting Machine (EBM)

- Şeffaf, açıklanabilir yapıda bir boosting algoritmasıdır.
- Özellikle sağlık sektöründe yorumlanabilirlik açısından önemli bir alternatiftir.
- Performansı iyi olmakla birlikte XGBoost seviyesine ulaşmamıştır.

```

metrics = {
    "accuracy": [], "precision": [], "recall": [], "f1": [],
    "roc_auc": [], "pr_auc": [], "mcc": []
}

skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

fold = 1
for train_idx, val_idx in skf.split(x, y):
    print(f"\n===== Fold {fold} =====")
    fold += 1

    x_tr, x_val = x.iloc[train_idx], x.iloc[val_idx]
    y_tr, y_val = y.iloc[train_idx], y.iloc[val_idx]

    model = ExplainableBoostingClassifier(interactions=10, random_state=42)
    model.fit(x_tr, y_tr)

    y_proba = model.predict_proba(x_val)[:, 1]
    y_pred = (y_proba >= 0.5).astype(int)

    # Her fold'un metriklerini topla
    metrics["accuracy"].append(accuracy_score(y_val, y_pred))
    metrics["precision"].append(precision_score(y_val, y_pred, zero_division=0))
    metrics["recall"].append(recall_score(y_val, y_pred, zero_division=0))
    metrics["f1"].append(f1_score(y_val, y_pred, zero_division=0))
    metrics["roc_auc"].append(roc_auc_score(y_val, y_proba))
    metrics["pr_auc"].append(average_precision_score(y_val, y_proba))
    metrics["mcc"].append(matthews_corrcoef(y_val, y_pred))

```

## 5. MODEL KARŞILAŞTIRMASI

Aşağıdaki değerlendirme metrikleri tüm modelleri adil bir şekilde karşılaştırmak için kullanılmıştır:

- Accuracy
- Precision
- Recall
- F1-score
- ROC-AUC
- PR-AUC
- MCC
- Confusion Matrix

### 5.1 Genel Sonuçlar (Özet)

Model	Güçlü Yanları	Zayıf Yanları	Genel Sonuç
Logistic Regression	Basit, yorumlanabilir	Karmaşık ilişkileri kaçırır	Orta
RandomForest	Non-linear iyi	Bazen overfit	Güçlü
RF + SMOTE	Dengesizlik iyi çözülür	Veri sentetikleştir	Çok iyi
XGBoost	En yüksek AUC, güçlü öğrenme	Parametre ayarı kritik	En iyi model
Neural Net	Esnek, pattern öğrenir	Veri azsa overfit	İyi
EBM	Çok iyi açıklanabilirlik	Boosting gücü sınırlı	Orta-İyi

## **6. SONUÇ**

Bu proje tıbbi bir sınıflandırma problemi olduğu için Recall, PR-AUC ve MCC gibi metrikler Accuracy'den daha kritik öneme sahiptir. Özellikle ölümleri kaçırılmamak (False Negative'i minimize etmek) sağlık alanında kritik olduğundan, XGBoost + düşük threshold yaklaşımı uygun bir stratejidir.

Ayrıca:

- SMOTE RF modelinin performansını ciddi şekilde artırmıştır.
- Neural Network modeli, veri seti büyündükçe daha da güçlü hale gelecektir.
- EBM'nin açıklanabilirliği, klinik ortamlarda "neden bu tahmin yapıldı?" sorusu için değerlidir.