داده کاوی - علی پاکدامن

آنالیز و پیشبینی سرطان ریه در دیتاست پزشکی بر اساس شاخص های باینری در پایتون با تمرین و تست الگوریتم های داده کاوی

منبع دیتاست و کد های jupyter notebook:

https://github.com/alipakdamangh/lung_cancer_analysis

فرمول معیار های استفاده شده

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Accuracy} = \frac{\text{True Positives (TP)} + \text{True Negatives (TN)}}{\text{Total Number of Instances}}$$

معیار Roc-Auc میزان Recall را با معکوس آن در Threshold های متفاوت مقایسه می‌کند تا با اندازه گیری trade off بین کلاس ها یک مقدار ارائه کند.

مدل های استفاده شده و معیار ها

	F1 Score	Recall	Accuracy	Precision	ROC-AUC
Naive Bayes	94.79	95.88	90.79	93.72	91.99
SVM	93.26	100.00	87.37	87.37	93.24
Logistic Regression	92.39	86.83	87.51	98.71	94.78
Gradient Boosting	98.59	98.87	97.53	98.31	99.43
Random Forest	99.77	99.54	99.60	100.00	100.00
Decision Tree	99.77	99.54	99.60	100.00	100.00
KNN	99.85	100.00	99.73	99.69	100.00

معیار های مدل Logistic Regression

```
Accuracy:     92.85393258426966 %
Recall Score: 97.69467213114754 %
Precision:    94.35922810489856 %
F1-score:     95.99798640825573 %
ROC-AUC:      90.2842205608599 %
Cross-validation scores: [0.91685912 0.91791908 0.92716763 0.9132948  0.92138728 0.92716763]
Mean Cross-Validation Score: 0.9206325897644697
```

بهبود معیار ها

بالانس دیتا را که بیشتر متمرکز روی مقادیر False است با روش SMOTE به هر طرف 50 درصد می‌رسانیم

```
Before SMOTE: Counter({1: 4528, 0: 663})
After SMOTE: Counter({1: 4528, 0: 4528})
```

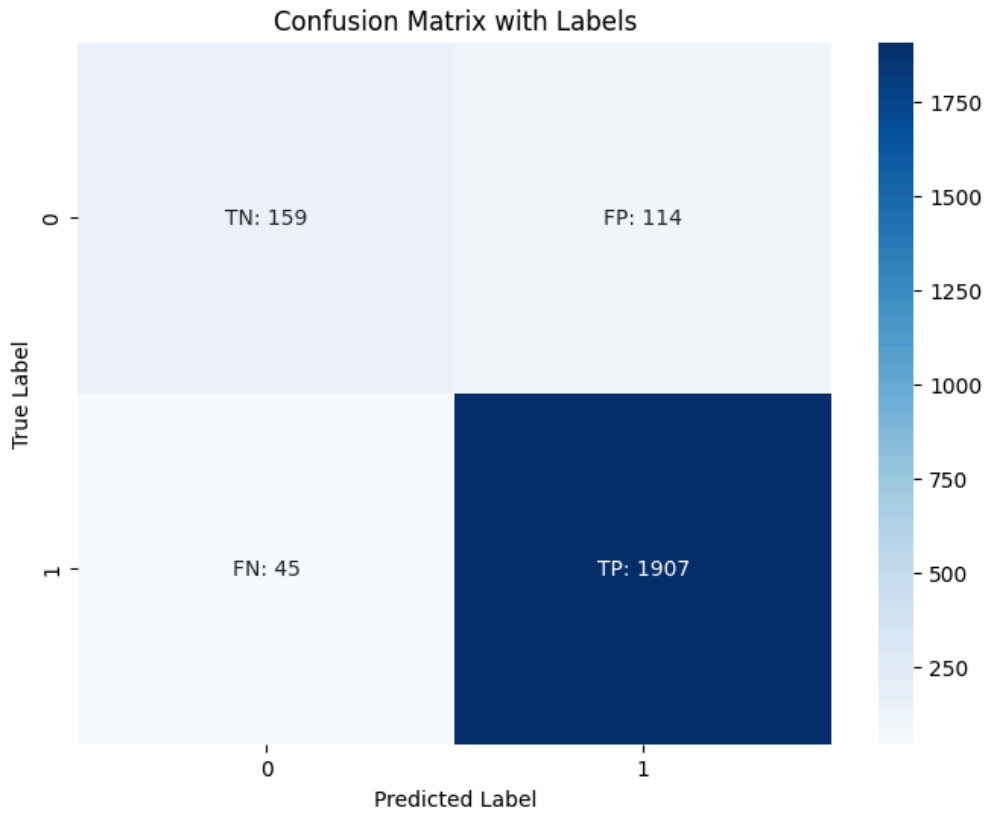
```
LUNG_CANCER
1    87.378641
0    12.621359
Name: proportion, dtype: float64
```

ستون هایی که تاثیر کمتری بر نتایج دارند را حذف کرده تا مدل با دیتاهای مهمتری آموزش و تست را انجام دهد

```
df = df.drop(['YELLOW_FINGERS', 'SWALLOWING_DIFFICULTY', 'SMOKING'], axis=1)
```

نتایج بهبود مدل

```
Accuracy:     95.4177897574124 %
Recall Score: 99.07692307692308 %
Precision:    95.83333333333334 %
F1-score:     97.42813918305598 %
ROC-AUC:      96.65551839464884 %
Cross-validation scores: [0.93021277 0.92163543 0.9241908  0.92248722 0.91567291 0.92248722]
Mean Cross-Validation Score: 0.9227810600843332
```



معیار های مدل Support Vector Machine - SVM

```
Accuracy: 87.40%
Precision: 87.40%
Recall: 100.00%
F1 Score: 93.28%
ROC-AUC Score: 87.76%
Cross-validation scores: [0.87360971 0.87360971 0.87360971 0.87360971 0.87348178 0.87449393]
Mean Cross-Validation Score: 0.8737357559333506
```

بهبود معیار ها

این الگوریتم نیازمند مقادیر باینری true وfalse است که ستون سن جزو این مقادیر نیست و باید با استاندارد سازی آن را به مدل بفهمانیم تا به صورت بهینه از آن در آموزش استفاده شود (فرمول استاندارد سازی در پایین شکل مثال زده شده)

```python
# Standardize the features ( scale the data so that it has a mean of 0 and a standard deviation to 1)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

$$18, 22, 30, 35, 40, 50, 60, 70$$

Step 1: Calculate Mean and Standard Deviation

- Mean (μ): 40.625
- Standard Deviation (σ): 17.0656

Step 2: Standardize Each Age

Each age is transformed using the formula:

$$z = \frac{x - \mu}{\sigma}$$

Transformed Ages (Mean = 0, Standard Deviation ≈ 1):

$$[-1.33, -1.09, -0.62, -0.33, -0.04, 0.55, 1.14, 1.72]$$

نتایج بهبود مدل

```
Accuracy: 97.37%
Precision: 99.14%
Recall: 97.85%
F1 Score: 98.49%
ROC-AUC Score: 98.74%
```

معیار های مدل Naive Bayes

```
F1 Score: 93.67%
Accuracy: 88.14%
Recall: 100.00%
Precision: 88.10%
ROC-AUC Score: 53.06%
```

بهبود معیار ها

برای کاهش ابعاد و نگه داشتن پارامتر های مهم از PCA استفاده می‌کنیم (درصد 95 را برای نگه داشتن حداکثر مقدار اصلی انتخاب می‌کنیم تا تفاوت زیادی بین دیتاست پیش فرض نباشد)

```python
# Apply PCA for dimensionality reduction (principle component analysis)
pca = PCA(n_components=0.95)  # Choose the number of components to retain 95% of variance
X_train = pca.fit_transform(X_train)
X_test = pca.transform(X_test)
```

همانند SVM سطون سن را استاندارد سازی می‌کنیم تا آموزش روی دیتای باینری انجام شود

```python
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

نتایج بهبود مدل

```
F1 Score: 95.72%
Accuracy: 92.31%
Recall: 97.70%
Precision: 93.82%
ROC-AUC Score: 91.39%
```

معیار های مدل KNN - K-Nearest Neighbors

Accuracy: 98.99%
Precision: 99.69%
Recall: 99.16%
F1 Score: 99.42%
ROC-AUC Score: 99.94%

معیار های مدل Decision Tree Classifier

Accuracy: 99.06%
Precision: 99.61%
Recall: 99.31%
F1 Score: 99.46%
ROC-AUC Score: 99.96%
Cross-validation scores: [0.99393327 0.98685541 0.98786653 0.98988878 0.9888664 0.98684211]
Mean Cross-Validation Score: 0.989042080974389

معیار های مدل Random Forest

Accuracy: 99.06%
Precision: 99.61%
Recall: 99.31%
F1 Score: 99.46%
ROC-AUC Score: 99.96%
Cross-validation scores: [0.99393327 0.98685541 0.98786653 0.98988878 0.9888664 0.98684211]
Mean Cross-Validation Score: 0.989042080974389

معیار های مدل Gradient Boosting

Accuracy: 97.17%
Precision: 97.07%
Recall: 99.77%
F1 Score: 98.40%
ROC-AUC Score: 99.75%
Cross-validation scores: [0.97876643 0.9817998 0.97371082 0.97168857 0.97165992 0.97975709]
Mean Cross-Validation Score: 0.9762304376481376