BM5702 MAKİNE ÖĞRENMESİNE GİRİŞ

Hafta 7

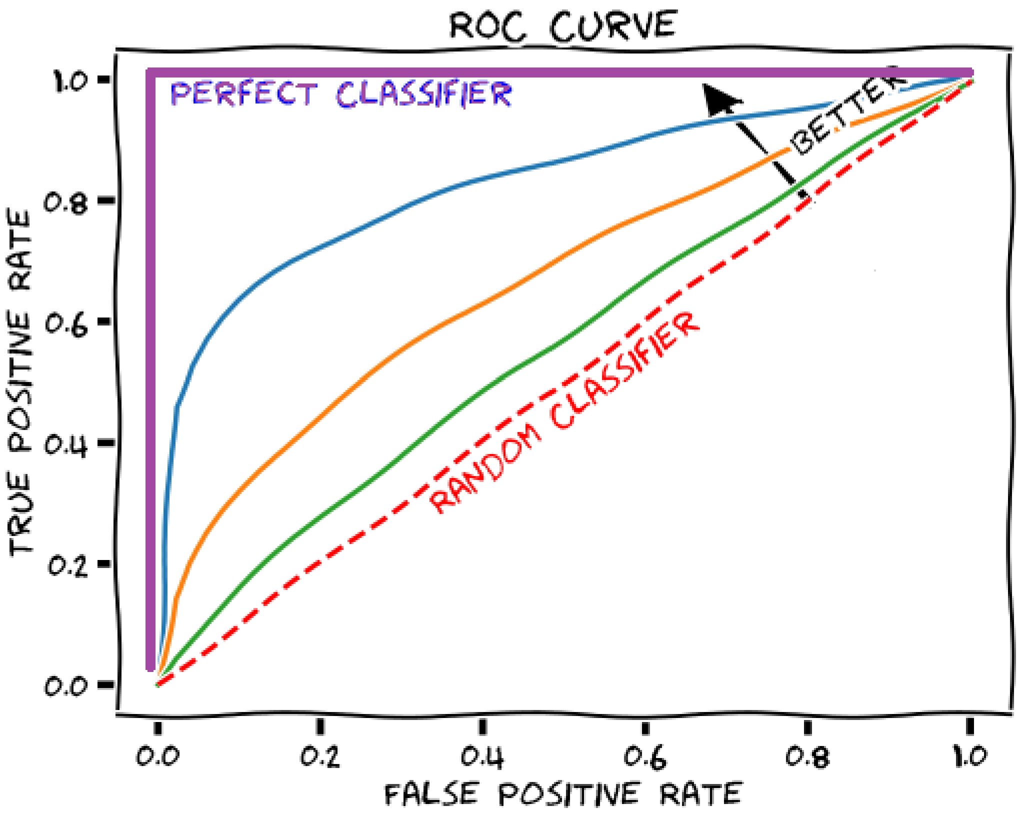
Doç. Dr. Murtaza CİCİOĞLU

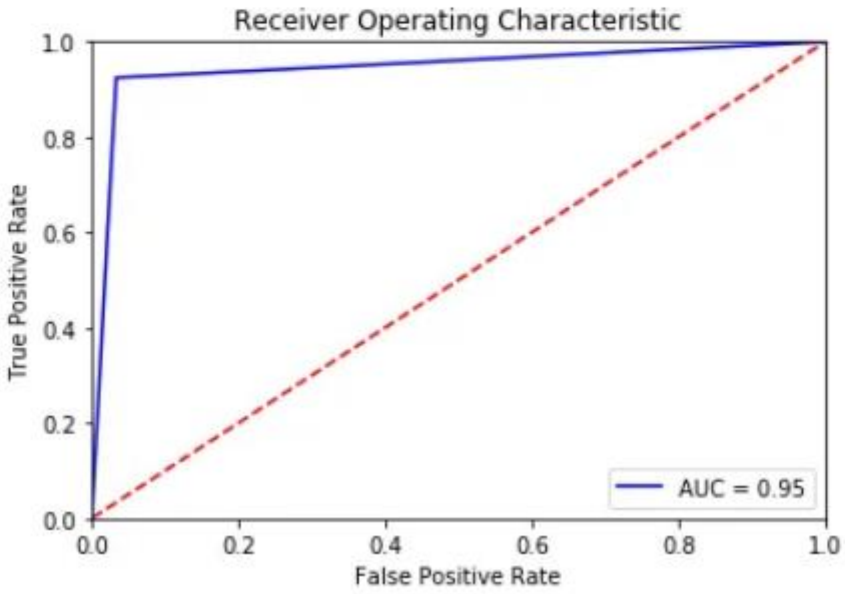
Receiver operating characteristics (ROC) and AUC

- There is another tool that is commonly used to analyze the behavior of classifiers at different thresholds: the receiver operating characteristics curve, or ROC curve for short.
- Similar to the precision-recall curve, the ROC curve considers all possible thresholds for a given classifier, but instead of reporting precision and recall, it shows the false positive rate (FPR) against the true positive rate (TPR).
- Recall that the true positive rate is simply another name for recall, while the false positive rate is the fraction of false positives out of all negative samples:

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}$$

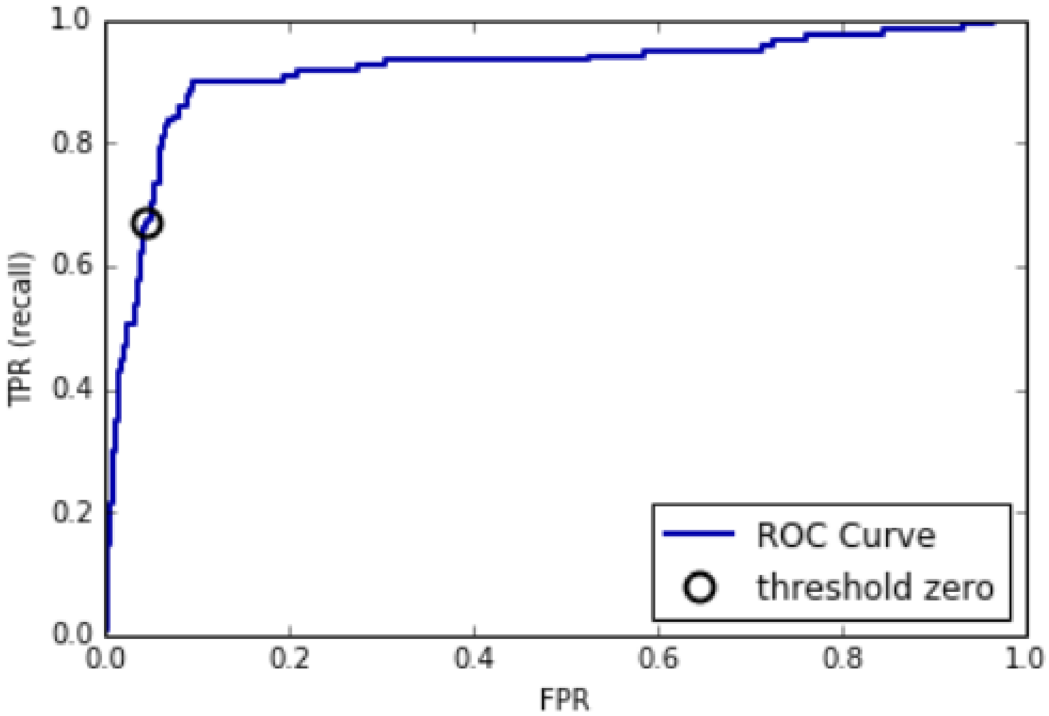
Receiver operating characteristics (ROC) and AUC





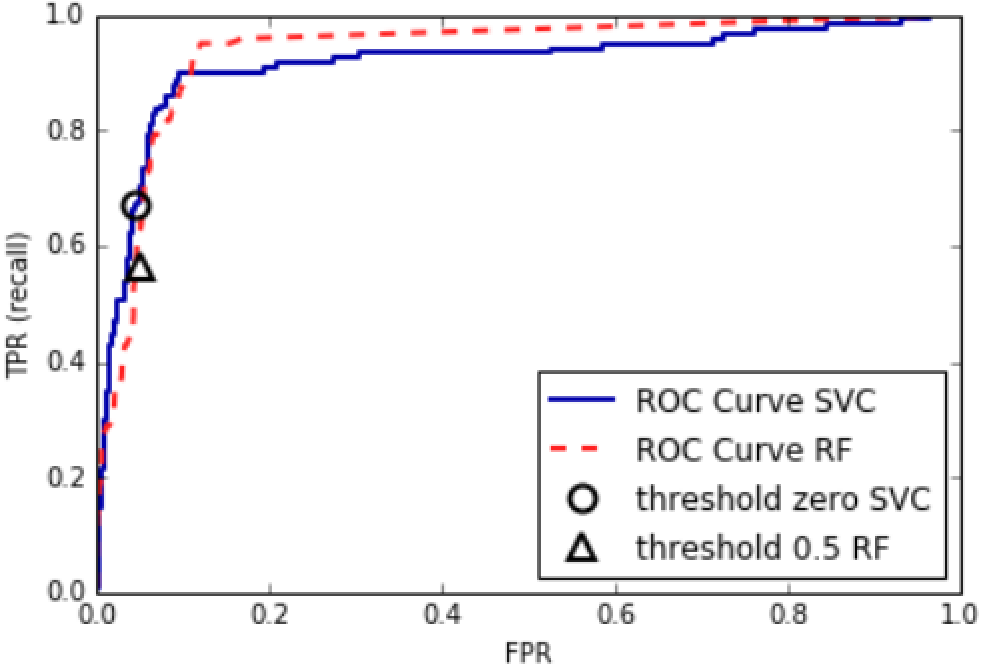
Receiver operating characteristics (ROC) and AUC

- For the ROC curve, the ideal curve is close to the top left: you want a classifier that produces a high recall while keeping a low false positive rate.
- Compared to the default threshold of 0, the curve shows that we can achieve a significantly higher recall (around 0.9) while only increasing the FPR slightly. The point closest to the top left might be a better operating point than the one chosen by default. Again, be aware that choosing a threshold should not be done on the test set, but on a separate validation set.



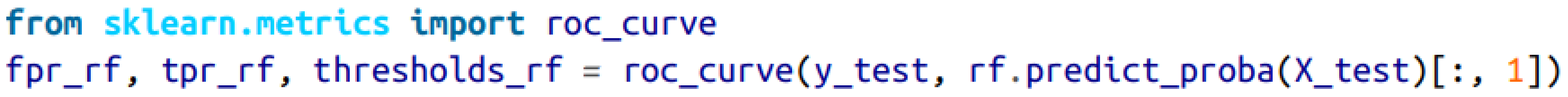
Receiver operating characteristics (ROC) and AUC

- As for the precision-recall curve, we often want to summarize the ROC curve using a single number, the area under the curve (this is commonly just referred to as the AUC, and it is understood that the curve in question is the ROC curve). We can compute the area under the ROC curve using the roc_auc_score function:

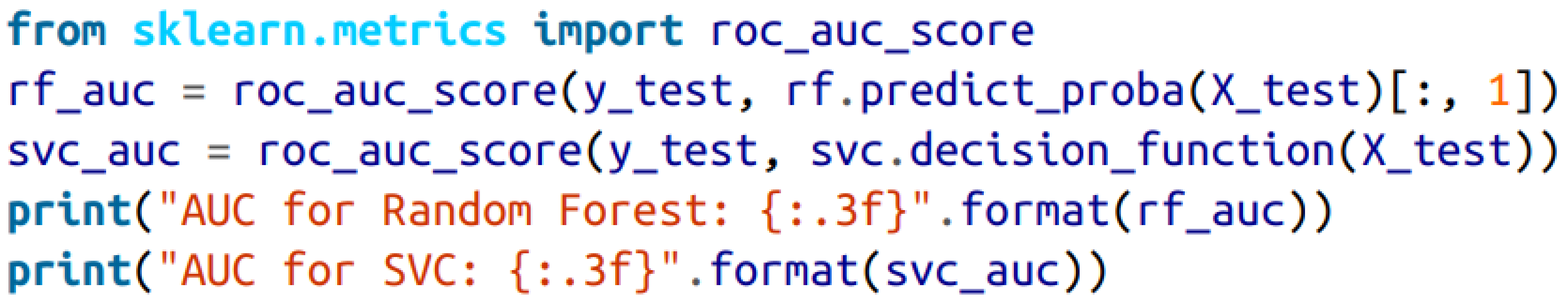


Receiver operating characteristics (ROC) and AUC

- For classification problems with imbalanced classes, using AUC for model selection is often much more meaningful than using accuracy



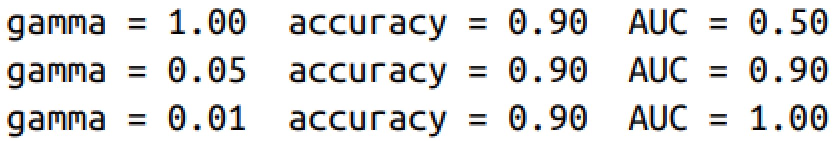
```python
from sklearn.metrics import roc_curve
fpr_rf, tpr_rf, thresholds_rf = roc_curve(y_test, rf.predict_proba(X_test)[:, 1])
```



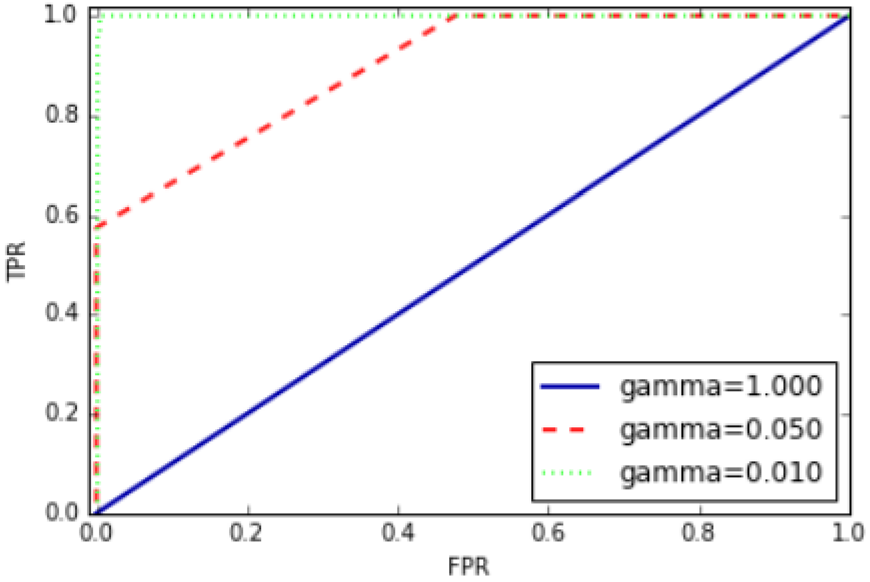
```python
from sklearn.metrics import roc_auc_score
rf_auc = roc_auc_score(y_test, rf.predict_proba(X_test)[:, 1])
svc_auc = roc_auc_score(y_test, svc.decision_function(X_test))
print("AUC for Random Forest: {:.3f}".format(rf_auc))
print("AUC for SVC: {:.3f}".format(svc_auc))
```

Receiver operating characteristics (ROC) and AUC

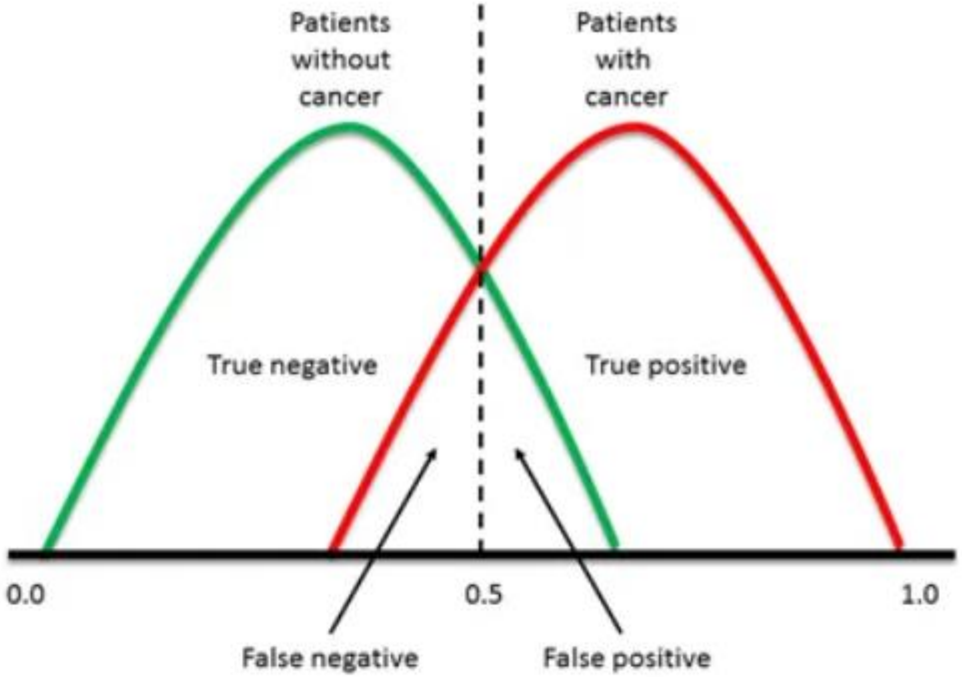
- For classification problems with imbalanced classes, using AUC for model selection is often much more meaningful than using accuracy

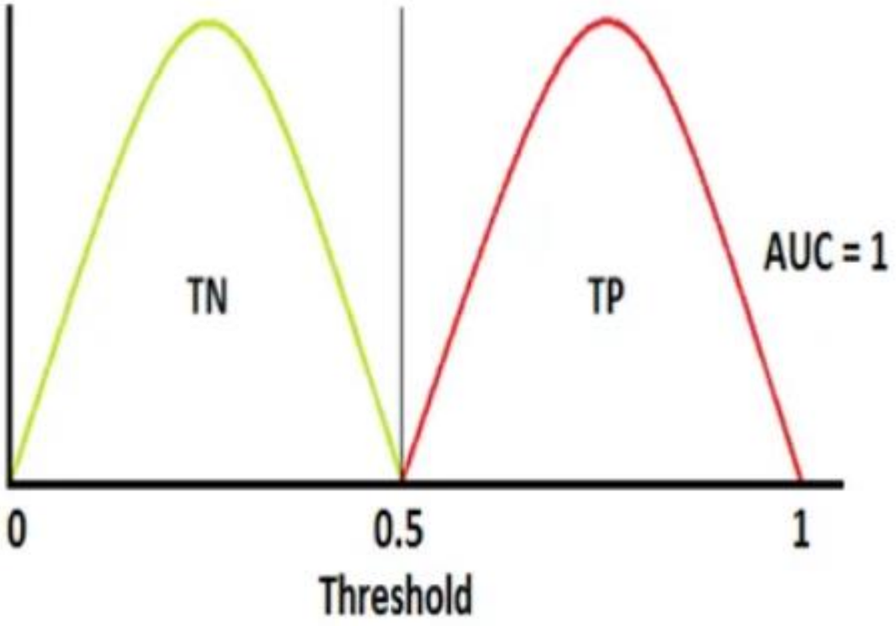


```
gamma = 1.00  accuracy = 0.90  AUC = 0.50
gamma = 0.05  accuracy = 0.90  AUC = 0.90
gamma = 0.01  accuracy = 0.90  AUC = 1.00
```



Receiver operating characteristics (ROC) and AUC





Naive Bayes Classifiers

Likelihood

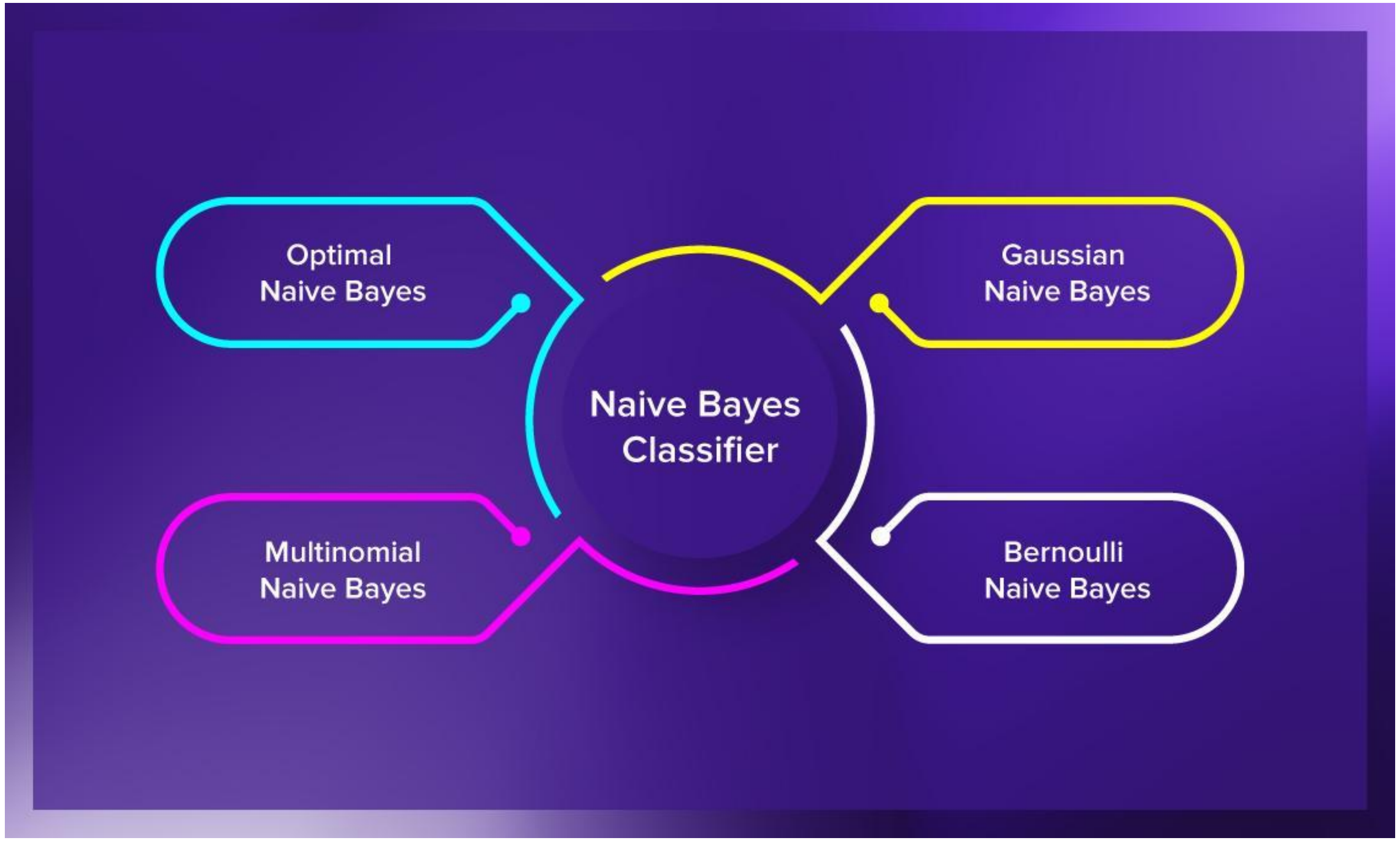
Class Prior Probability

$$P(c \mid x) = \frac{P(x \mid c)\,P(c)}{P(x)}$$

Posterior Probability

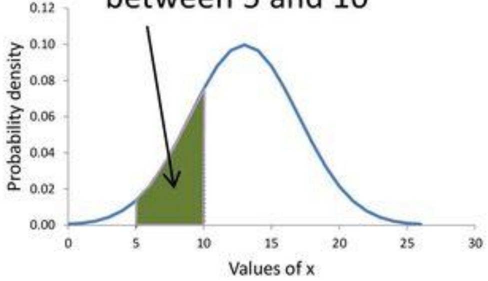
Predictor Prior Probability

$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$



Naive Bayes Classifiers

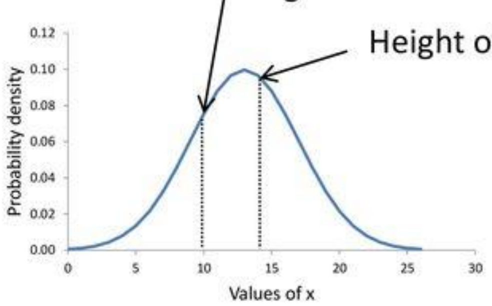
Probability


Area under curve between 5 and 10

What is the *probability* that $5 \le x \le 10$ given a normal distribution with $\mu = 13$ and $\sigma = 4$? Answer: 0.204

What is the *probability* that $-1000 \le x \le 1000$ given a normal distribution with $\mu = 13$ and $\sigma = 4$? Answer: 1.000

Likelihood


Height of curve at $x = 10$

Height of curve at $x = 14$

What is the *likelihood* that $\mu = 13$ and $\sigma = 4$ if you observed a value of

(a) $x = 10$ (answer: the *likelihood* is 0.075)

(b) $x = 14$ (answer: the *likelihood* is 0.097)

Conclusion: if the observed value was 14, it is *more likely* that the parameters are $\mu = 13$ and $\sigma = 4$, because 0.097 is higher than 0.075.

Naive Bayes Classifiers

- Naive Bayes classifiers are a family of classifiers that are quite similar to the linear models. However, they tend to be even faster in training.
- The price paid for this efficiency is that naive Bayes models often provide generalization performance that is slightly worse than that of linear classifiers like LogisticRegression and LinearSVC.
- The reason that naive Bayes models are so efficient is that they learn parameters by looking at each feature individually and collect simple per-class statistics from each feature.

Naive Bayes Classifiers

- There are three kinds of naive Bayes classifiers implemented in scikit-learn: GaussianNB, BernoulliNB, and MultinomialNB.
- GaussianNB can be applied to any continuous data, while BernoulliNB assumes binary data and MultinomialNB assumes count data (that is, that each feature represents an integer count of something, like how often a word appears in a sentence).
- BernoulliNB and MultinomialNB are mostly used in text data classification.
- The other two naive Bayes models, MultinomialNB and GaussianNB, are slightly different in what kinds of statistics they compute.
- MultinomialNB takes into account the average value of each feature for each class, while GaussianNB stores the average value as well as the standard deviation of each feature for each class.

Naive Bayes Classifiers

- To make a prediction, a data point is compared to the statistics for each of the classes, and the best matching class is predicted.
- Interestingly, for both MultinomialNB and BernoulliNB, this leads to a prediction formula that is of the same form as in the linear models
- Unfortunately, coef_ for the naive Bayes models has a somewhat different meaning than in the linear models, in that coef_ is not the same as w.

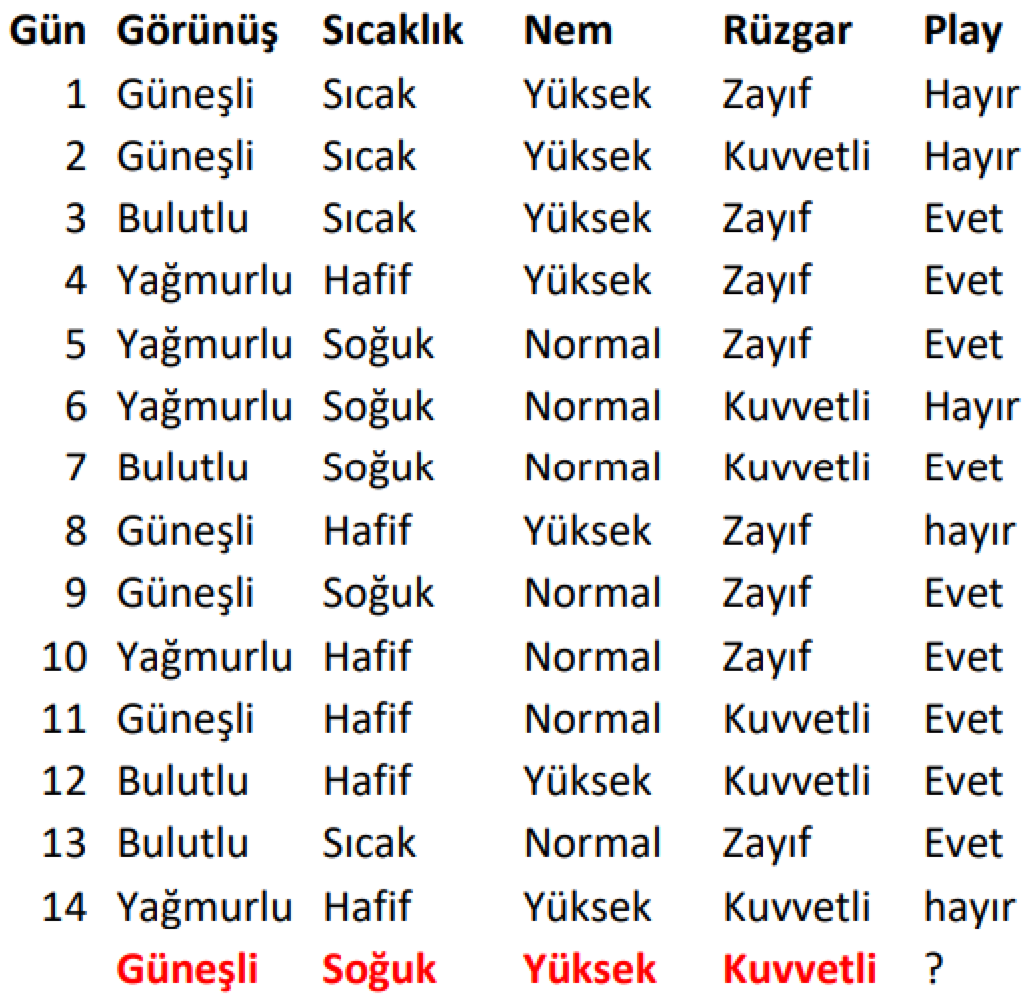
Strengths, weaknesses, and parameters

- MultinomialNB and BernoulliNB have a single parameter, alpha, which controls model complexity.
- The way alpha works is that the algorithm adds to the data alpha many virtual data points that have positive values for all the features. This results in a "smoothing" of the statistics. A large alpha means more smoothing, resulting in less complex models. The algorithm's performance is relatively robust to the setting of alpha, meaning that setting alpha is not critical for good performance. However, tuning it usually improves accuracy somewhat.
- GaussianNB is mostly used on very high-dimensional data, while the other two variants of naive Bayes are widely used for sparse count data such as text. MultinomialNB usually performs better than BernoulliNB, particularly on datasets with a relatively large number of nonzero features (i.e., large documents).

Strengths, weaknesses, and parameters

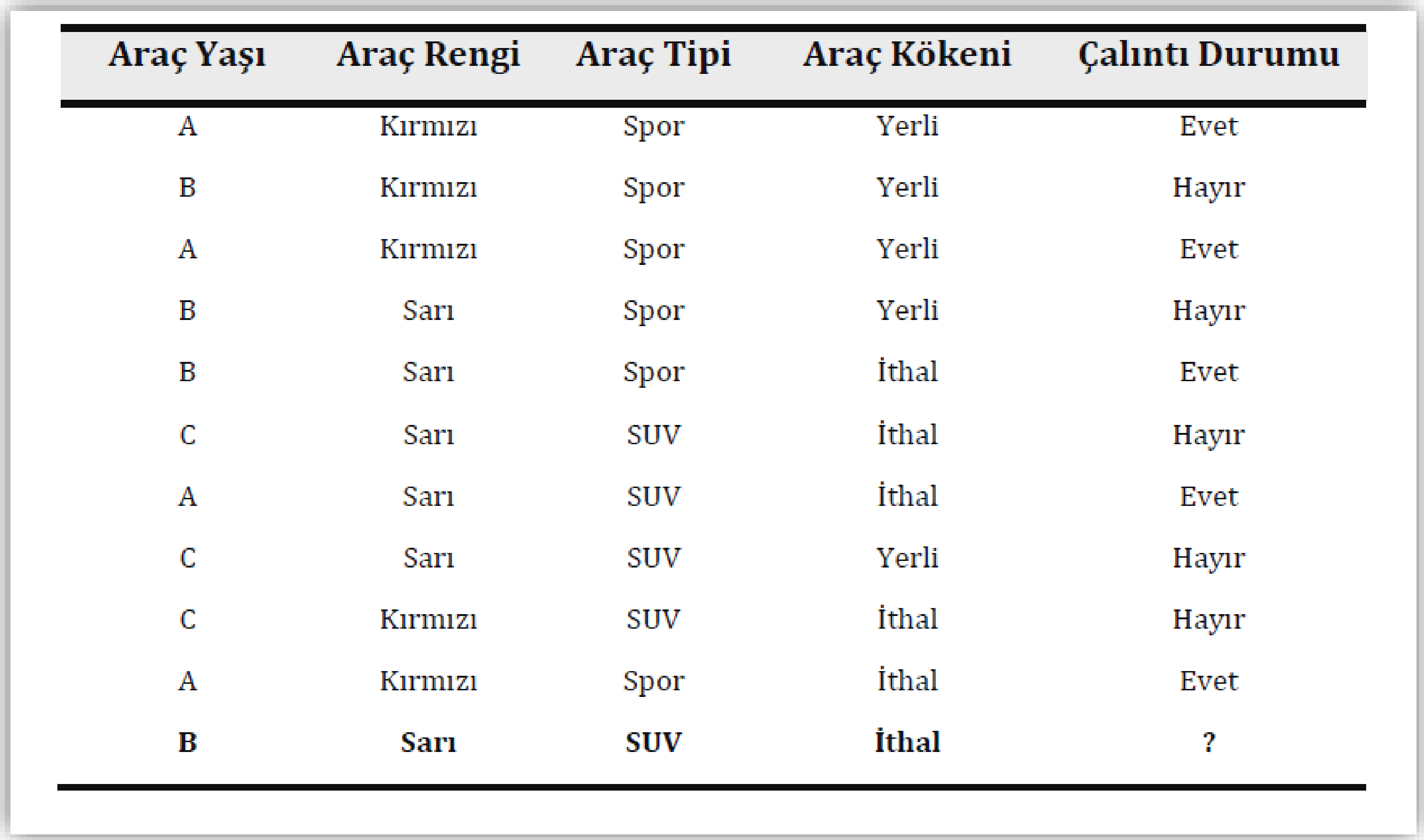
- The naive Bayes models share many of the strengths and weaknesses of the linear models.
- They are very fast to train and to predict, and the training procedure is easy to understand.
- The models work very well with high-dimensional sparse data and are relatively robust to the parameters.
- Naive Bayes models are great baseline models and are often used on very large datasets, where training even a linear model might take too long.

???



Gün	Görünüş	Sıcaklık	Nem	Rüzgar	Play
1	Güneşli	Sıcak	Yüksek	Zayıf	Hayır
2	Güneşli	Sıcak	Yüksek	Kuvvetli	Hayır
3	Bulutlu	Sıcak	Yüksek	Zayıf	Evet
4	Yağmurlu	Hafif	Yüksek	Zayıf	Evet
5	Yağmurlu	Soğuk	Normal	Zayıf	Evet
6	Yağmurlu	Soğuk	Normal	Kuvvetli	Hayır
7	Bulutlu	Soğuk	Normal	Kuvvetli	Evet
8	Güneşli	Hafif	Yüksek	Zayıf	hayır
9	Güneşli	Soğuk	Normal	Zayıf	Evet
10	Yağmurlu	Hafif	Normal	Zayıf	Evet
11	Güneşli	Hafif	Normal	Kuvvetli	Evet
12	Bulutlu	Hafif	Yüksek	Kuvvetli	Evet
13	Bulutlu	Sıcak	Normal	Zayıf	Evet
14	Yağmurlu	Hafif	Yüksek	Kuvvetli	hayır
	Güneşli	**Soğuk**	**Yüksek**	**Kuvvetli**	?

???



Araç Yaşı	Araç Rengi	Araç Tipi	Araç Kökeni	Çalıntı Durumu
A	Kırmızı	Spor	Yerli	Evet
B	Kırmızı	Spor	Yerli	Hayır
A	Kırmızı	Spor	Yerli	Evet
B	Sarı	Spor	Yerli	Hayır
B	Sarı	Spor	İthal	Evet
C	Sarı	SUV	İthal	Hayır
A	Sarı	SUV	İthal	Evet
C	Sarı	SUV	Yerli	Hayır
C	Kırmızı	SUV	İthal	Hayır
A	Kırmızı	Spor	İthal	Evet
B	**Sarı**	**SUV**	**İthal**	**?**