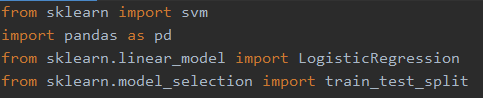
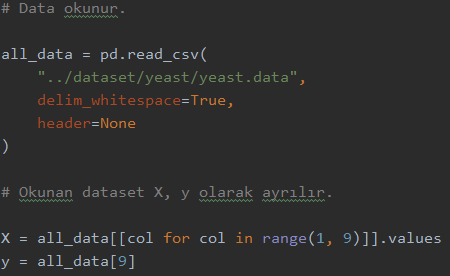
1. **Yeast (yeast\_svm\_logistic.py):**

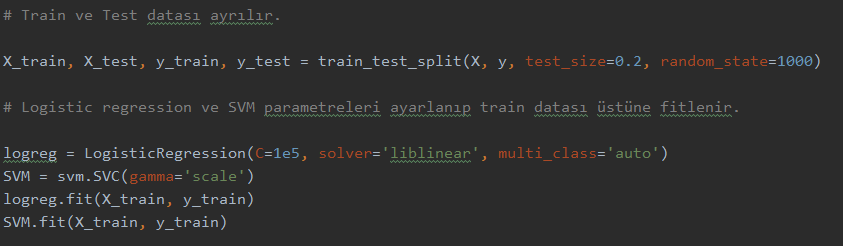
Gerekli kütüphaneler import edilir.



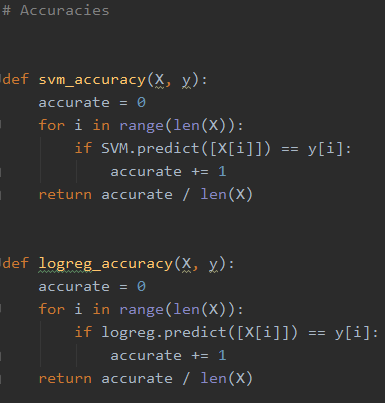
Data okunur. (data arasında virgül yerine boşluk kullanılmış olmasına ve başlık kullanılmamasına dikkat edildi.)



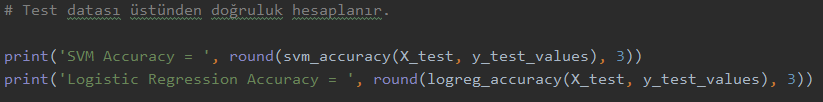
Data, train ve test (%80, %20) olarak ikiye ayrıldıktan sonra logistic regression ve SVM uygulanır.



Kendim yazmaya çalıştığım accuracy fonksiyonları ile test data üstünde doğruluk hesaplanacaktır.



Sonuç olarak :

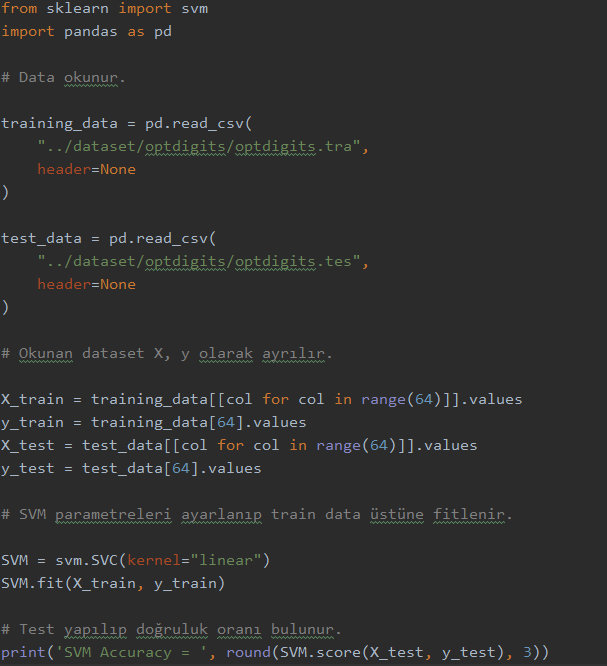




Oran yakın olsa da SVM’nin daha başarılı olduğu görülüyor.

1. **Optdigits (optdigits\_svm.py):**

**a.1.** Linear kernel ile SVM uygulanan kod.

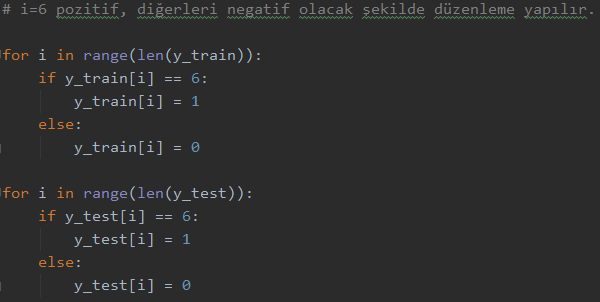


Sonuç:



**a.2. (optdigits\_svm\_pos\_neg.py):**

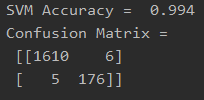
Bir diğer örnekte i = 6 seçilerek i == 6: olan sınıflar 1 diğer sınıflar 0 olarak düzenlendi



Confusion matrix de eklendi.

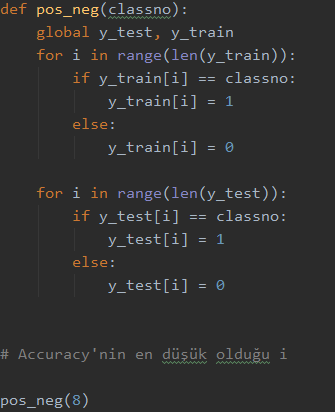


Sonuç:

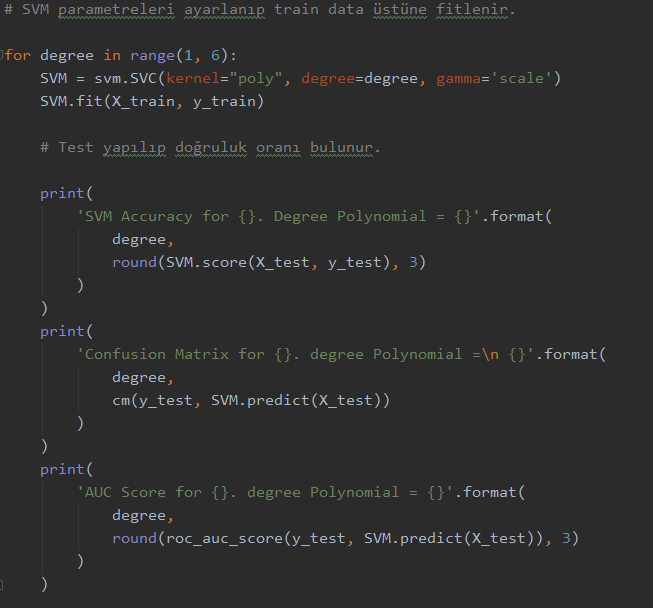


**b. (optdigits\_polynomial.py):**

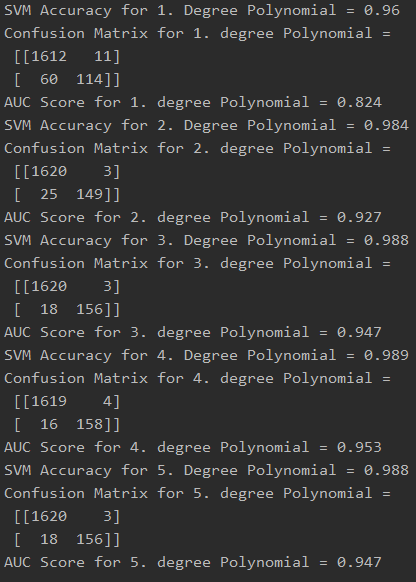
Her bir “i” denenerek en düşük accuracy veren bulundu ve onun üstünden devam edildi.



Sırayla 1,2,3,4,5. Derece polinom çekirdekleri denendi.

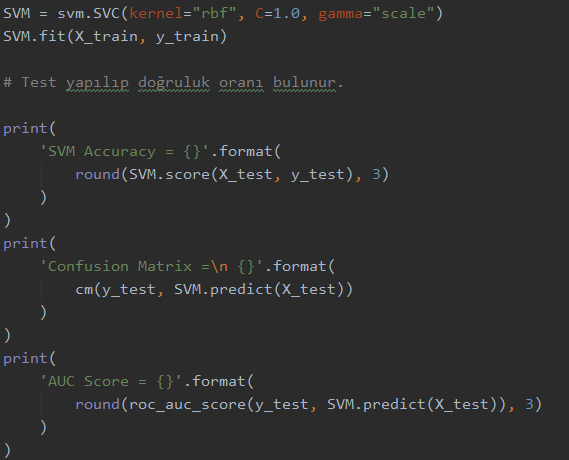


Sonuç:



**c. (optdigits\_svm\_rbf.py):**

RBF çekirdeği kullanıldı, C ve gamma aşağıdaki gibi verildiğinde sonuç yüksek çıktı.



Sonuç:

