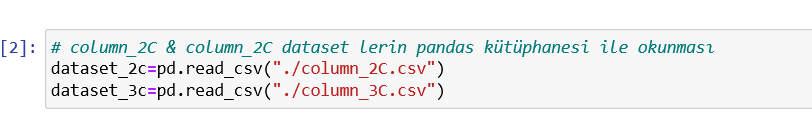
**YAPAY ZEKA PROJE ÖDEVİ HAKKINDA**

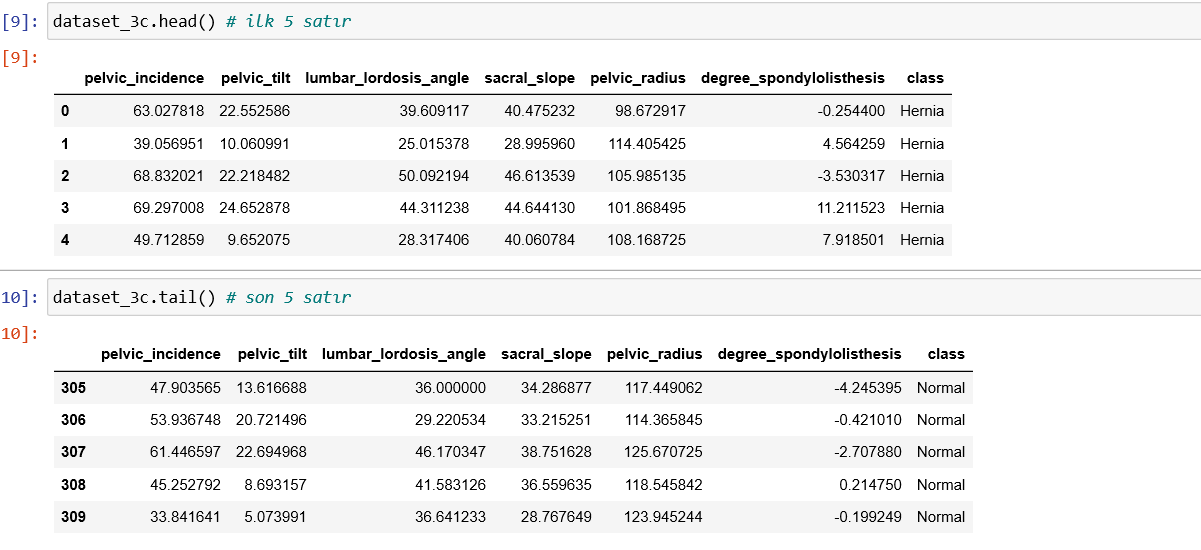
* Veri analizi projeye başlanıldı. Bunun için kütüphaneler dahil edildi. ( Kütüphaneler numpy , pandas , matplotlib , seaborn )



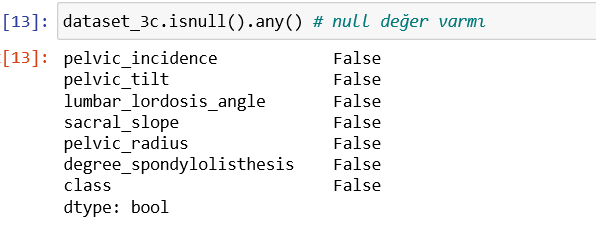
* İlk olarak datasetler olan ” column\_2c ve column\_3c “ yi okunup projeye dahil edildi.



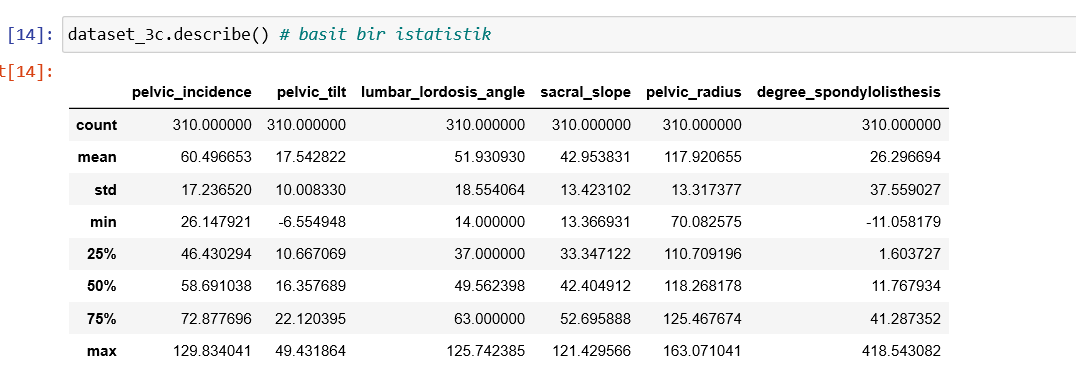
* Data içerikleri kontrol edildi. Bunun için datanın ilk beş ve son beş öğesine bakıldı.



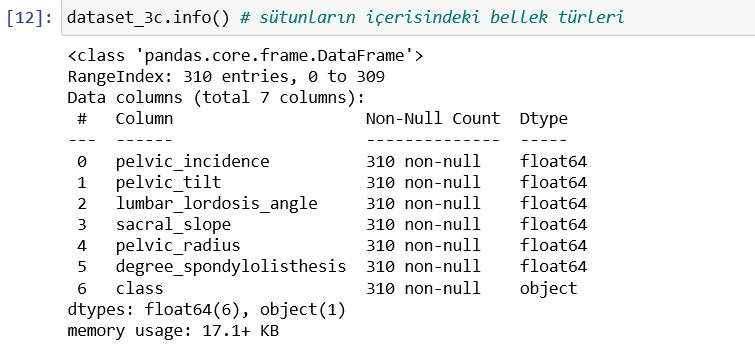
* Datasetler içerisinde null değer olup olmadığına bakıldı.



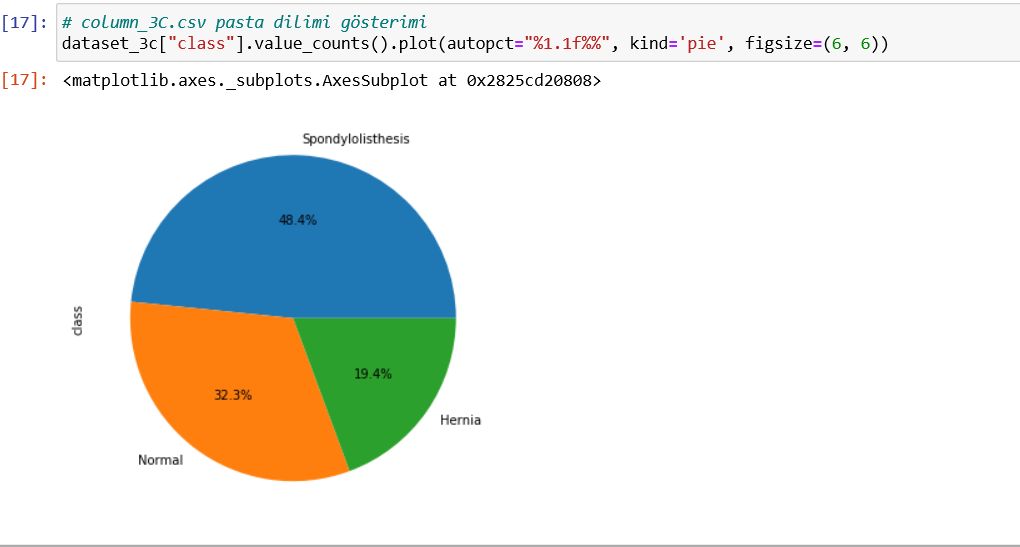
* Describe metodu ile küçük bir istatistik değerlendirmesi yapıldı.

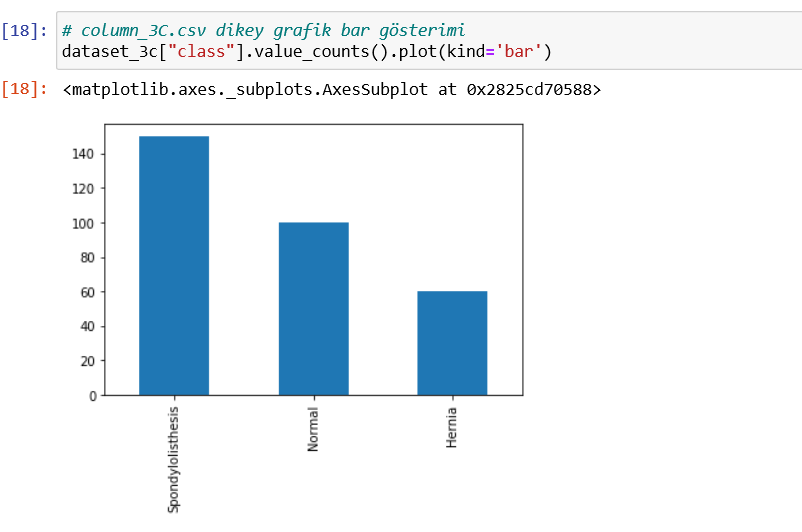


* İnfo metodu ile veri ve bellek türlerine bakıldı.
* Ayrı ayrı iki veri setine bu işlemler uygulandı.

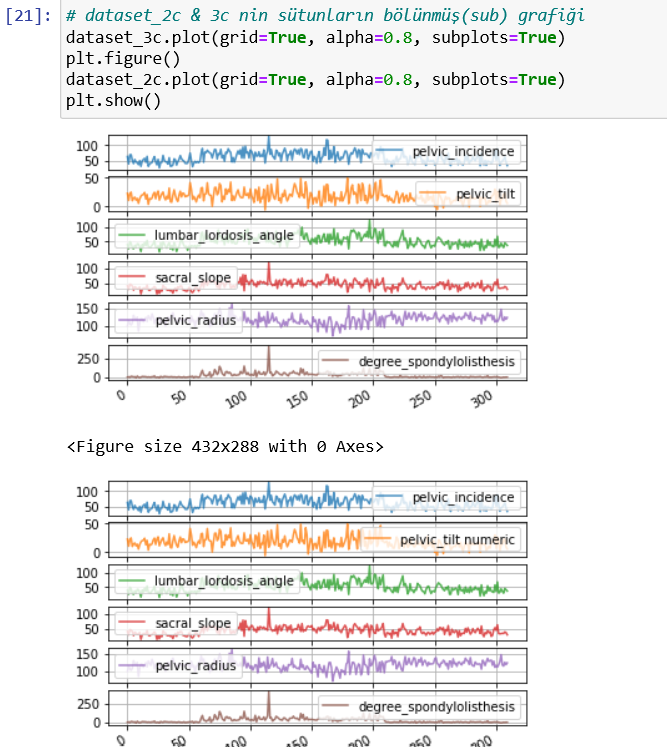


* “ pie , bar , grid , histogram “ metotlarıyla veri setlerine görselleştirme yapıldı.

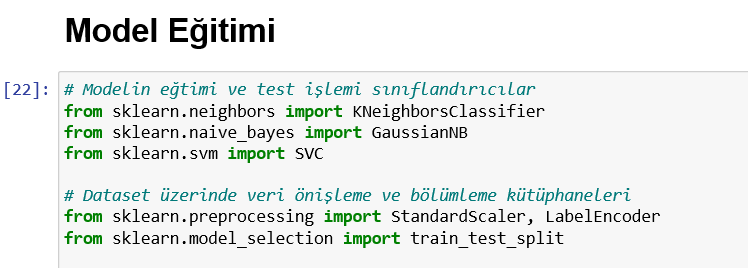




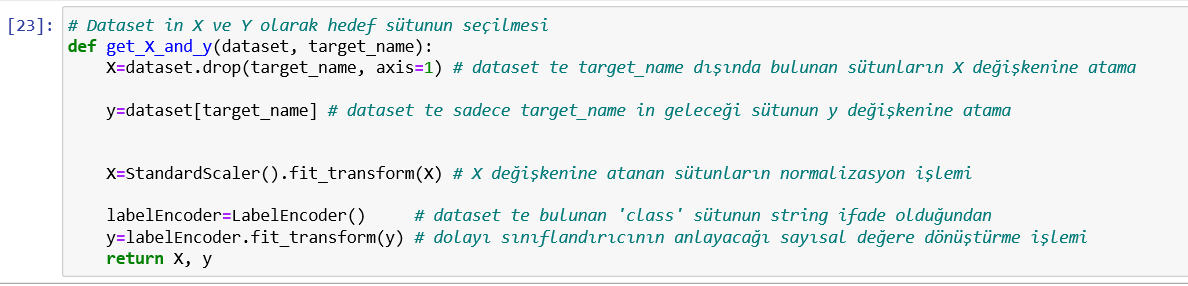
* Görselleştirme yapıldıktan sonra grafikteki değerlerde karışıklık olmaması için her özelliğe karşılık gelen grafik çizimleri isimlendirildi.



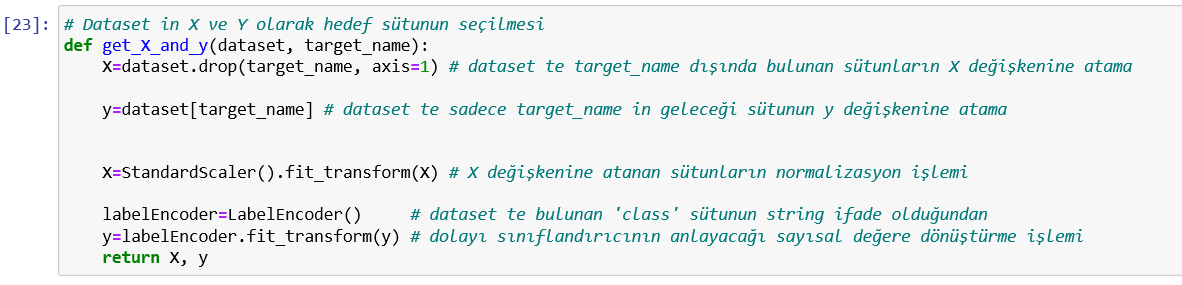
* Daha sonra veri setlerinden oluşacak modelin eğitim işlemi yapıldı.
* Eğitim işlemi yapılırken “ KneighborsClassifier ( kNN ) , GaussianNB , SVC ( svm ) “ algoritmaları kullanıldı.
* Bu 3 farklı algoritma , hangisinin daha iyi sonuç verdiği görülmesi açısından kullanıldı ve gözlemlendi.
* Bu sınıflandırmaların alt metotları kullanarak veriler üzerinde işlem yapıldı ve model eğitimi gerçekleştirildi.



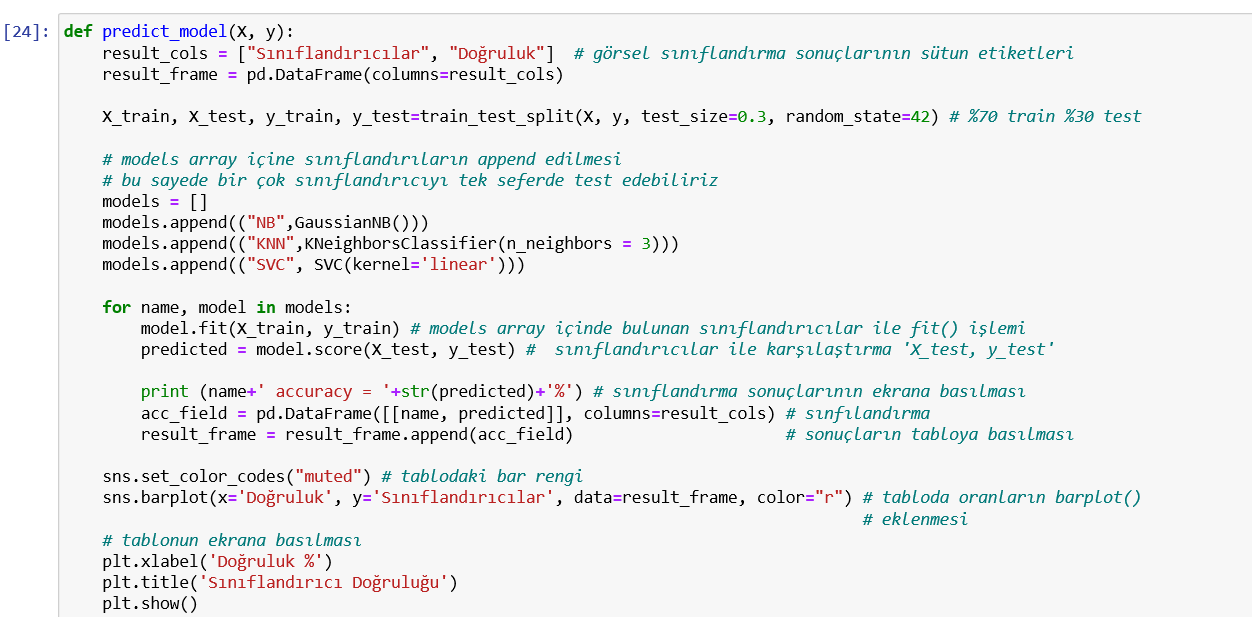
* Gerekli açıklamalar kod satırları üzerinde de yapıldı.
* Üzerinde işlem yapılması için hedef sütunlar seçildi.
* Data setler üzerindeki string değerlerin sayısal karşılığı için fit\_transform işlemi gerçekleştirildi.



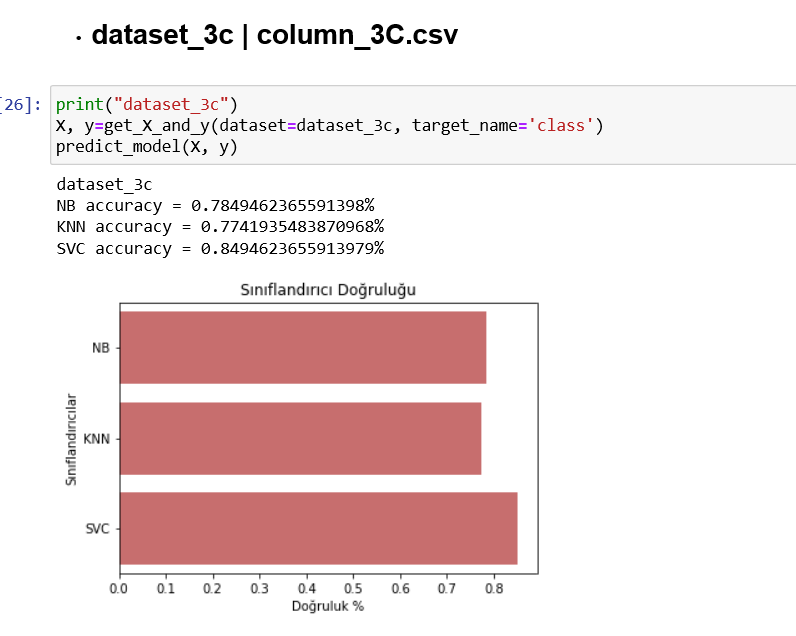
* Hedef sütunlar üzerinde ( X , Y ) normalizasyon işlemi yapıldı.

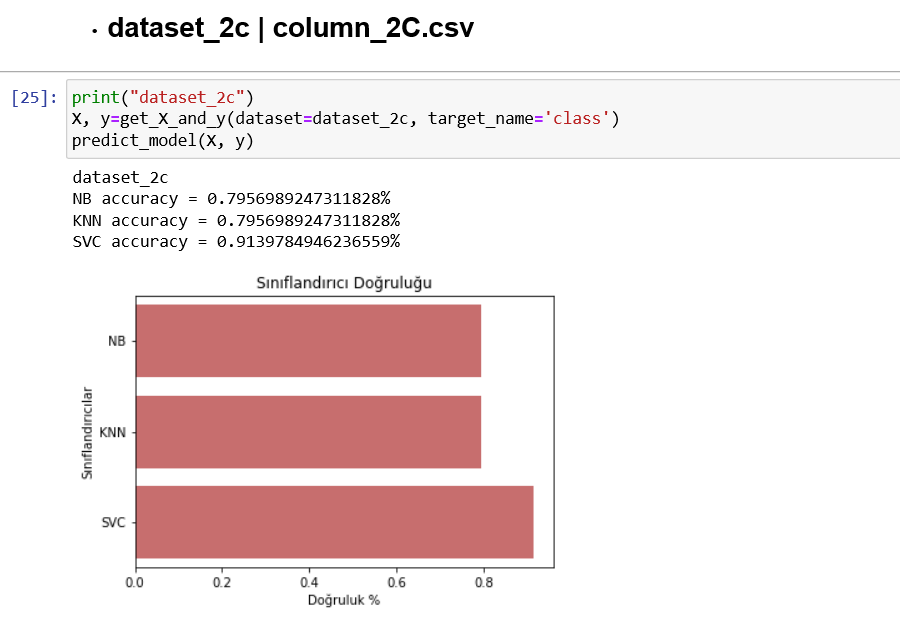


* Daha sonra 3 sınıflandırıcının değerleri gösterildi. Değerler görselleştirildi ve nasıl sonuç verdikleri görüldü.

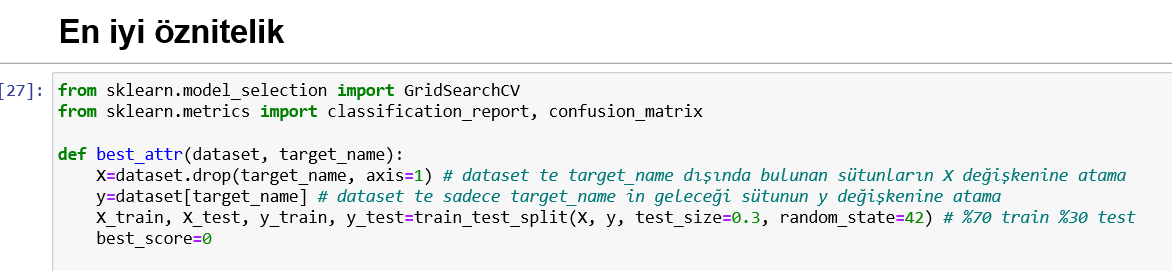


* Sonuçlara göre “ SVC “ nin daha iyi sonuç verdiği görüldü.

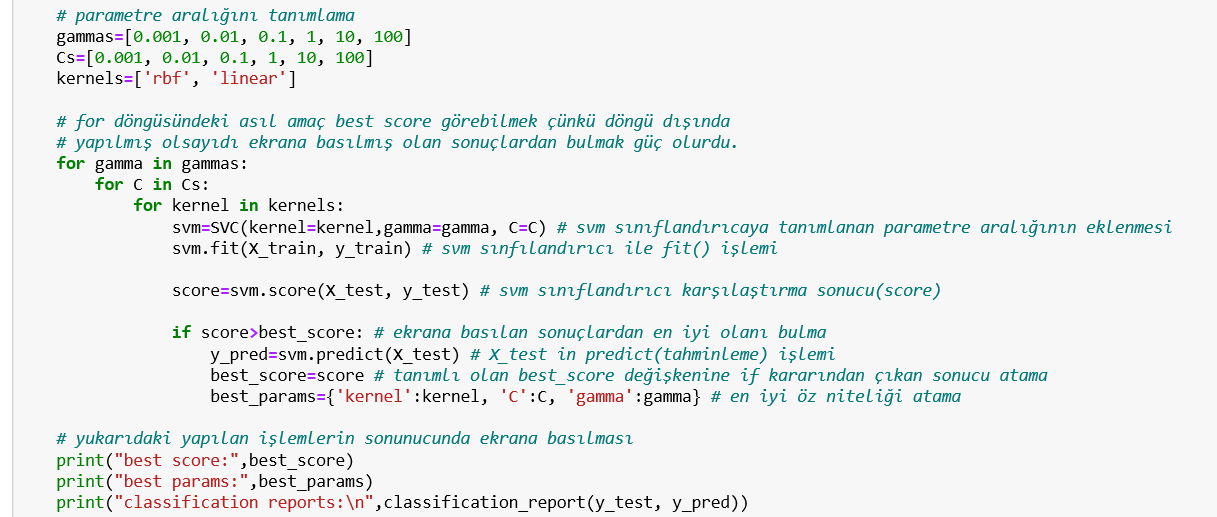




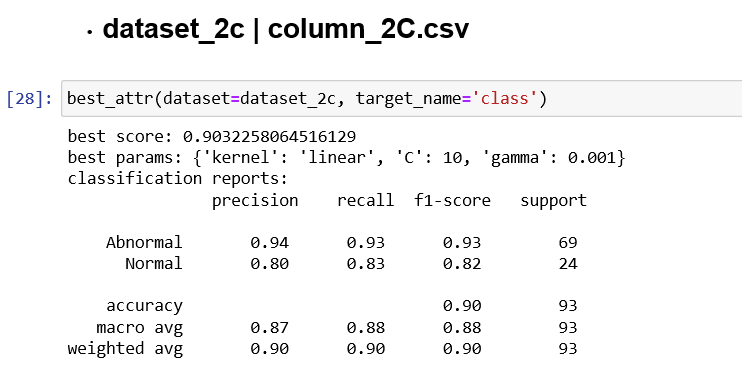
* Son olarak , en iyi öz nitelik hangisi ona bakıldı. Bunun için %70 train %30 test uygulaması uygulandı.



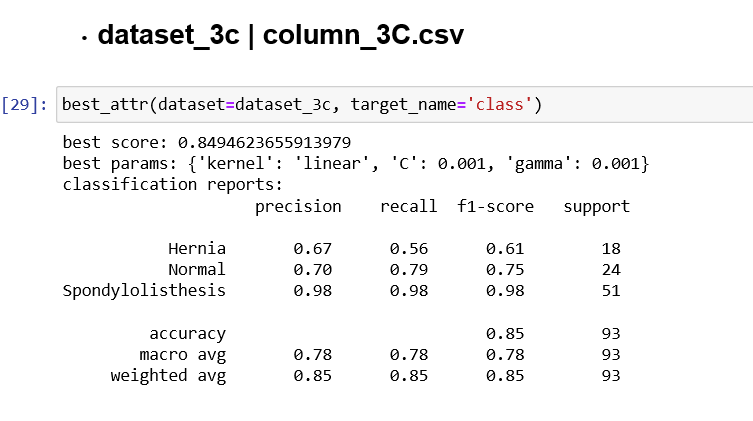
* For döngüsü ile veriler içinde gezinti yapıldı ve en iyi skoru verecek olan öz niteliğe bakıldı.



* Column\_2c veri seti için en iyi öznitelik ;

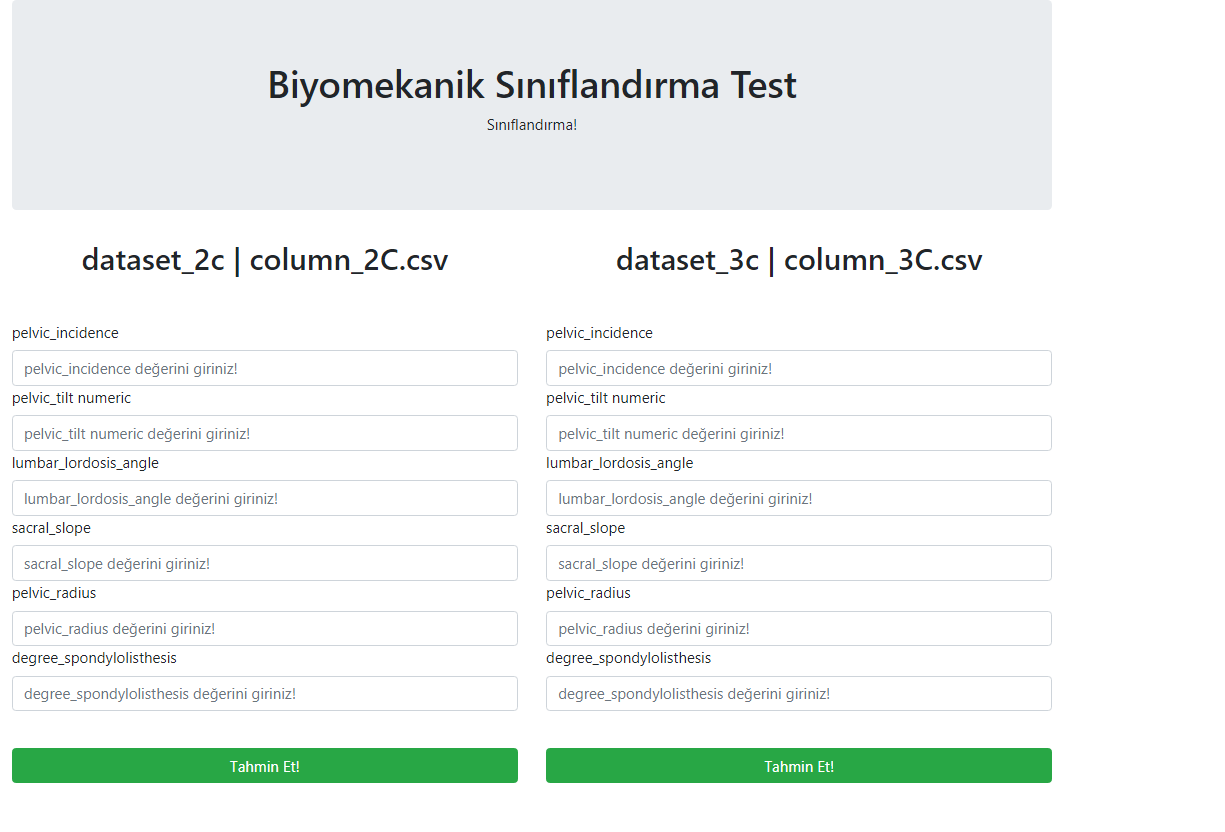


* Column\_3c veri seti için en iyi öznitelik ;



**PROJE İÇİN VEB UYGULAMASI**

* Proje için daha kolay değerlendirme için web uygulaması tasarlandı.
* Web uygulamasında veri setleri içerisindeki niteliklere göre , sonucu hasta yada normal döndüren bir uygulama tasarlandı.
* Kod çalıştırmayla uğraşmak yerine direkt olarak değer girip sonuç alınması için tasarlandı.
* Web uygulaması Visual Studio üzerinde yazıldı.
* Web uygulaması tam manasıyla çalışmamakta , bazı durumlar için hasta sonucu vermesi beklenirken normal sonucu vermektedir..
* Uygulamanın daha da geliştirilmesi gereklidir.
* Projeye ekstra olması için tasarlanmıştır.
* Uygulama web üzerinde çalışmaktadır.
* Uygulama için python kütüphaneleri gereklidir.
* Cmd üzerinden dosya içine girilip install -r requirements komutuyla yüklenip çalıştırılabilir.
* Web uygulaması arayüzü şu şekildedir :



* Değerler girildikten sonra 2 dataset içinde tahmin et butonu bulunmaktadır. Tıklanıldığı takdirde sonuç verecektir.
* Tüm değerleri ayrı ayrı girmek yerine tek satırda kopyala yapıştır yapabilmek için yukarıda ki görseldeki her bir nitelik değeri satırlarını birbirine bağlayarak tek satırda işlem görülebilir.
* Bunun için web uygulaması geliştirilmelidir.
* <https://biyomekanik-test.herokuapp.com/> Uygulama test bağlantısı

**KAYNAKÇA ( 1/4 )**

* <https://www.kaggle.com/serkanpeldek/alt-zelli-e-g-re-hastalar-n-s-n-fland-r-lmas>
* <https://www.geeksforgeeks.org/svm-hyperparameter-tuning-using-gridsearchcv-ml/>
* <https://pandas.pydata.org/pandas-docs/version/0.15.0/visualization.html>
* <https://forum.yazbel.com/t/python-kutuphaneler-pip/2539>
* <https://medium.com/@amine.yesilyurt/python-k%C3%BCt%C3%BCphaneleri-e59fe08cc276>
* <https://tr.bitdegree.org/tutorial/python-kutuphaneleri/#API_ve_Python_Goz_Onunde_Bulundurulmasi_Gereken_En_Iyi_Kutuphaneler>
* <https://tr.bitdegree.org/tutorial/python-kutuphaneleri/>
* <https://yazilimportal.com/python-ile-flask-framework-%C3%BCzerinde-%C3%A7al%C4%B1%C5%9Fmak-1c1c483c25e7>

**KAYNAKÇA ( 2/4 )**

* <https://www.buraksenyurt.com/post/Flask-ile-Basit-Bir-Web-Uygulamas%C4%B1-Gelistirmek>
* <https://medium.com/@awesome_nyn/python-ile-flask-microframework-kullanarak-nas%C4%B1l-web-projesi-olu%C5%9Fturulur-fbca456e7c71>
* <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>
* <https://www.python-course.eu/k_nearest_neighbor_classifier.php>
* <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn>
* <https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html>
* <https://scikit-learn.org/0.15/modules/generated/sklearn.naive_bayes.GaussianNB.html>

**KAYNAKÇA ( 3/4 )**

* <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>
* <https://scikit-learn.org/stable/modules/svm.html>
* <https://pythonprogramming.net/linear-svc-example-scikit-learn-svm-python/>
* <https://swcarpentry.github.io/python-novice-gapminder/09-plotting/>
* <https://matplotlib.org/tutorials/introductory/pyplot.html>
* <https://medium.com/datarunner/matplotlibkutuphanesi-1-99087692102b>
* <https://medium.com/datarunner/python-i%CC%87le-veri-g%C3%B6rselle%C5%9Ftirme-matplotlib-k%C3%BCt%C3%BCphanesi-2-8d1cffef75e8>
* <https://www.veribilimiokulu.com/seaborn-ile-veri-gorsellestirmesi/>

**KAYNAKÇA ( 4/4 )**

* <https://medium.com/datarunner/python-i%CC%87le-temel-veri-analizi-c51b29158f30>
* <https://medium.com/deep-learning-turkiye/python-ile-veri-bilimine-dal%C4%B1%C5%9F-3f069260ebda>
* <https://www.kaggle.com/mert34/python-le-temel-veri-analizi>
* <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html>
* <https://www.kaggle.com/cesartrevisan/scikit-learn-and-gridsearchcv>

**…OKUDUĞUNUZ İÇİN TEŞEKKÜR EDERİZ…**