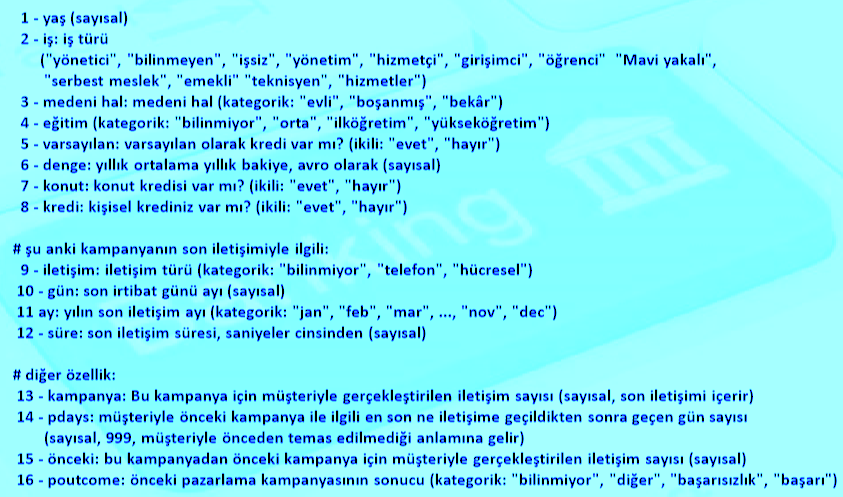
# İstatistiksel Veri Madenciliği Dönem Ödevi

## 1.Giriş

Bir Portekiz bankasının pazarlama kampanyaları ile ilgili telefon görüşmeleri sonucu müşterilerin belirli kriterleri sağlayıp sağlamadıkları ile ilgilenilmektedir. Veri seti Mayıs 2008- Kasım 2010 tarihleri arasını kapsamaktadır. Veri madenciliği yöntemleri kullanılarak önceden var olan bilgiler bilgisayara öğretilecek ve gelecekteki müşterilerin durumu tahmin edilecektir. İlk etapta vadeli mevduata sahip olup olmadığı ile ilgili tahmin yapmayı hedefliyoruz, ikinci hedefimiz ise müşterinin bankada bulundurduğu yıllık ortalama bakiyedir.

## 2.Veri Tanıtımı

Elimizdeki veri setinde toplamda 17 tane değişken bulunmaktadır. Bunlardan 16 tanesi programlara Girdi, 1 tanesi Çıktı olacaktır. Toplamda 45211 tane gözlem değerimiz vardır. Elimizde ki verilerin açıklamaları aşağıda ki gibidir.



Şekil 1 - Veri Girdileri

Verilerin türleri aşağıda ki tabloda ki gibidir.

Say. = Sayısal, Kat. = Kategorik

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Yaş | İş | M.Hal | Eğitim | Varsay. | Denge | Konut | Kredi | İletişim | Gün | Ay | Süre | Kamp. | PDays | Önce. | O.Come |
| Say. | Kat. | Kat. | Kat. | İkili | Say. | İkili | İkili | Kat. | Say. | Kat. | Say. | Say. | Say. | Say. | Kat. |

Şekil 2 - Veri Türleri

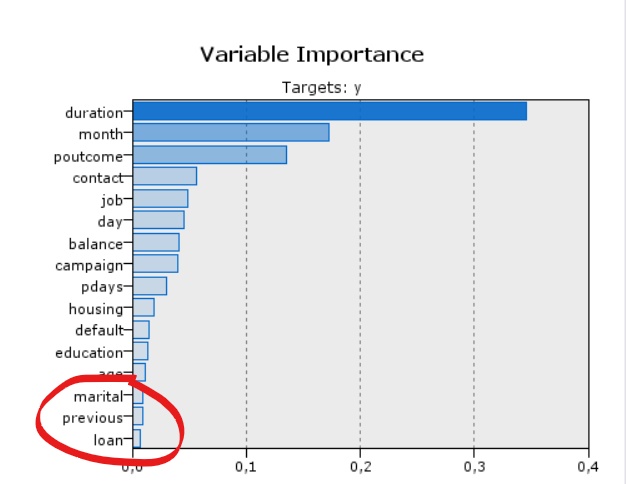
## 3.Veri Önişleme

Veri ön işleme de iki tür işlem vardır. Bunlardan ilki veri kümelerinin birleştirilmesi ve seçilmesidir. Bu aşama bizde zaten hazır durumdadır çünkü çalışma yapacağımız veri kümesi bellidir. İkinci işlem ise verilerin daha yararlı hale getirilmesi aşamasıdır. Bizim üstünde durmamız gereken kısım ikinci kısımdır. Elimizde ki verilerin nasıl elde edildiği veya hangi kaynaklardan toplanarak birleştirildiği gibi bilgiler bizde sınırlı olduğu için veri önişleme süreci için yapabileceğimiz şeylerde sınırlıdır.



Elde ki verilerde varsa gereksiz değişkenlerin filtrelenmesi (Veri Temizleme) veri kümesini sade hale getirmek için gerekli olabilen bir adımdır. Bunun için ileri yönlü ya da geri yönlü sezgisel seçimler yapılabilir. Biz geri yönlü sezgisel seçim yapmak adına tahmin yöntemlerinden MULTIPLE, PRUNE ve QUICK Neural Network yöntemlerinin üçünde de ortak olarak en az önem derecesine sahip 3 tane değişkeni tespit edilip bunları filtreleyerek, filtrelenmemiş tahmin değerleri ile karşılaştırdık.

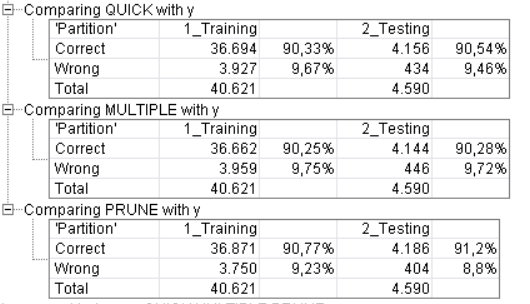
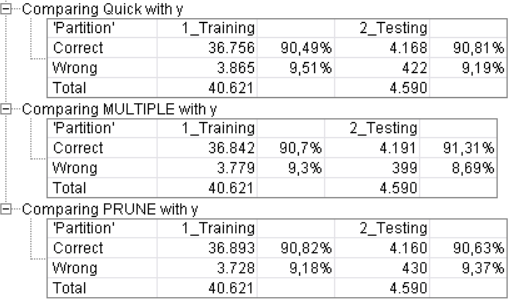




Şekil 3 - Filtreleme

3 değişken içerisinde en çok doğru tahmin yüzdesine sahip MULTIPLE değişkeninin önem sıralaması dikkate alınarak filtrelenmiş değişkenler marital, loan ve previous’dur.

Şimdi elimizde ki verileri filtrelenmemiş tahmin değerleri ile karşılaştıralım.

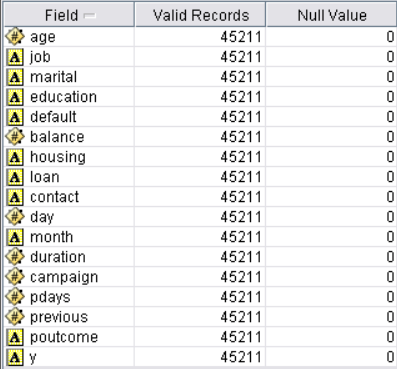


Şekil 4 - Filtrelenmemiş Yüzde Şekil 5 - Filtrelenmiş Yüzde

Grafikler üzerinde görüldüğü üzere PRUNE algoritması filtrelenmiş verilerde yüzde olarak bir artışa sebep olmuştur fakat tersi durumda MULTIPLE algoritması yaklaşık yüzde 1’lik bir değer kaybına sebep olmuştur. Filtrelenmiş verileri tercih etmek istersek PRUNE üzerinden devam edilebilir. Fakat elimizde ki veri sayısının az olmasından ötürü bilgisayarı çok fazla yormayacağından ve diğer algoritmalarda bu 3 değişkenin etkili olabileceği göz önüne alınırsa veri önişleme aşamasında verileri filtrelemeden devam etmeye karar verdik.

Verilerin büyüklüğünden ötürü veriler içerisinde bazen kayıp veriler bulunabilmektedir. Bu kayıp veriler yine veri önişleme yapılarak bazı tekniklerle giderilebilir. Elimizde ki verilerin önsel bilgisinde kayıp veri olmadığını biliyoruz fakat yine de Clementine’ın bize sunduğu Data Audit Nodu ile bunu doğruladık.





Şekil 6 - Boş Değer

Şekil 6 - Boş Değer’de görüldüğü üzere bu veri kümesi içerisinde herhangi bir kayıp veri yoktur. Dolayısıyla bunun için herhangi bir çalışma yapmaya gerek yoktur.

## 4.Önsel Bilgi

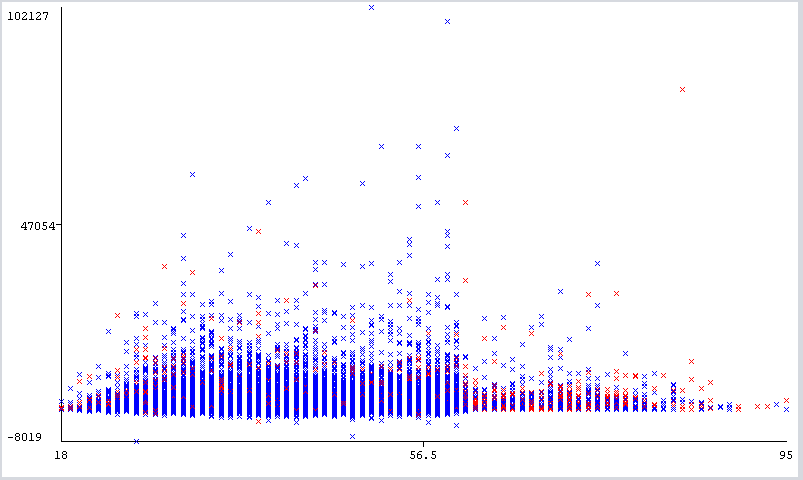
Veriler hakkında bir önsel bilgiye sahip olabilmemiz ve bunu görselleştirebilmemiz için bunu bir bilir kişiye sormamız gerekir. Bizim araştırmamızda bilir kişi olmadığı için hangi değişkenleri görselleştireceğimiz hakkında yapacağımız seçimler kısıtlıdır.

Önsel Bilgi adına veri görselleştirme yaptığımız program Weka’dır.



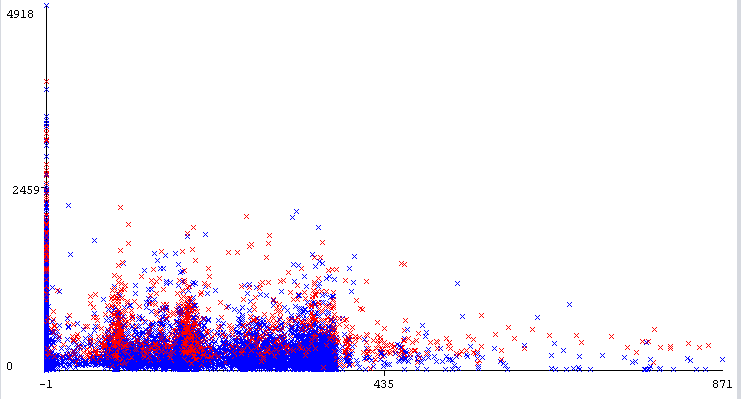
Şekil 7 - Yaş Grafiği

Şekil 7 - Yaş Grafiğinde ve bundan sonra ki bütün önsel bilgi grafiklerinde ana değişken ‘y’ değişkeni olarak belirlenmiştir. Mavi boyalı kısımlar ‘Hayır’, Kırmızı boyalı kısımlar ‘Evet’ i temsil etmektedir. Yaşların dağılım grafiğinde görüldüğü üzere elimizde ki veri kümesinin büyük çoğunluğu 18 ile 60 yaş aralığında bulunmaktadır. Ayrıca mavi boyalı kısımların çok daha fazla yer kapladığı görülmektedir. Buda veri kümesinde elinde vadeli mevduat bulundurmayan kişilerin çoğunlukta olduğu söylenebilir.



Şekil 8 Yaş-Yıllık Bakiye Grafiği

Şekil 8 Yaş-Yıllık Bakiye Grafiği x ekseninde yaş, y ekseninde yıllık ortalama bakiyeyi göstermektedir. Bu grafikte yapabileceğimiz yorum 60 yaşın altında ki kişilerin daha çok yıllık bakiyeye sahip olduğu ve vadeli mevduat açmaya meyilli olmadıkları, 60 yaşın üstündekilerin ise daha az bakiyeye sahip olmalarına karşın daha çok vadeli mevduat açtıklarıdır. Eğer elimizde ki veri kümesinde yıllık ortalama bakiyeye, vadeli mevduatların içinde ki paralarda dahil ise bu grafik 60 yaşın üstündekilerin daha az paraya sahip olsalar bile daha iyi yatırım yaptıklarını göstermektedir.



Şekil 9 Duration-PDays Grafiği

Şekil 9 Duration-PDays Grafiği x ekseninde müşteri ile bir önceki kampanya için en son ne zaman iletişime geçildiği (eğer iletişime geçilmedi ise -1) (PDays), y ekseninde saniyeler cinsinden son iletişim süresini (Duration) göstermektedir. Grafikte dikkat çekici nokta x ekseni -1 olan yani daha önceki kampanyalarda hiç iletişime geçilmemiş kişilerle yapılan telefon görüşmesi süresi çok fazla. Bu süre arama sayısı arttıkça azalmış.

## 5.Makine Öğrenmesi

İlk olarak hedef değişkenimiz ‘y’ değişkeni oldu. Yani müşterinin vadeli mevduatının olup olmadığı ile ilgilendik. Yapmak istediğimiz tahmin sınıflandırıcı bir tahmin yani kullanacağımız algoritmaların hepsi sınıflandırıcı yöntemleri kapsayan algoritmalar oldu.

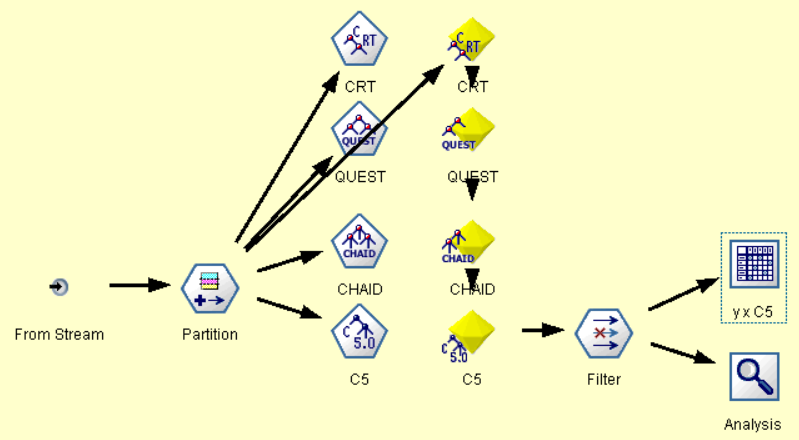
İkinci olarak ise hedef değişkenimiz ‘denge’ (balance) yani müşterinin bankada bulundurduğu yıllık ortalama bakiyedir. Burada kullandığımız tahmin algoritmaları sürekli bir veriyi tahmin etmeye yarayan algoritmalar oldu.

### 5.1 Sınıflandırıcı Tahmin

‘y’ değişkenini tahmin ederken kullandığımız algoritmalar Karar Ağaçları, Bayesçi Ağlar, Sinir Ağları ve k-En Yakın Komşu şeklindedir. Bütün algoritmalarda Partition Nodu ile verinin %90’ı öğrenme %10’u test verisi olarak seçilmiş ve bu hiçbir algoritmada değiştirilmemiştir.

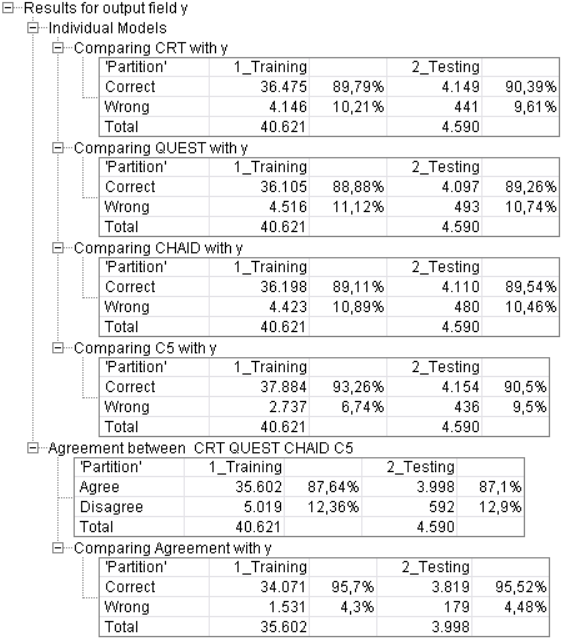
#### 5.1.1 Karar Ağaçları

Karar Ağaçları içerisinde kullandığımız yöntemler Classification ve Regression Tree, Quest, Chaid, C5, Random Forest algoritmalarıdır. Bu algoritmaların aldıkları doğru tahmin yüzdeleri farklı programlarda tekrar tekrar denenmiştir.



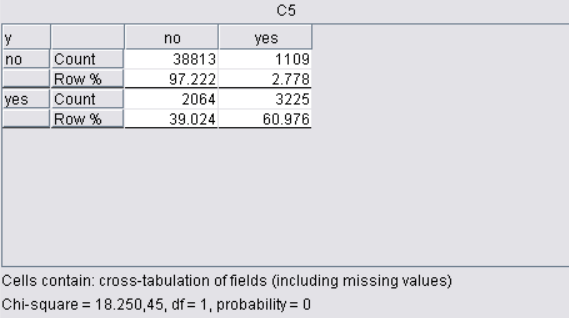
Şekil 10 Karar Ağaçları Clementine

Şekil 10 Karar Ağaçları Clementine’da görüldüğü gibi ilk uygulama Clementine üzerinde yapılmıştır ve doğru tahmin yüzdeleri aşağıda ki gibidir.



Şekil 11 Sınıflandırma Karar Ağaçları Doğru Tahmin Yüzdeleri

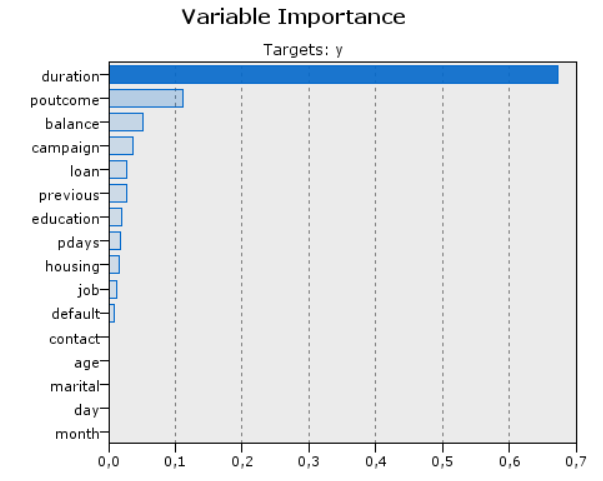
Grafikte görüldüğü üzere test kısmında en yüksek doğru tahmin yüzdesine sahip algoritma C5 algoritmasıdır. Karar ağaçları içerisinde bir algoritma kullanmak istersek C5 algoritmasını kullanacağız.



Şekil 12 C5 Konfüzyon Matrisi

Yukarıda C5 algoritmasının Konfüzyon matrisini görmekteyiz. Matrise göre C5 algoritması vadeli mevduatı olmayan insanları %97, vadeli mevduatı olanları %60 oranında doğru tahmin etmiştir. Buna göre vadeli mevduatı olmayanların daha iyi tahmin edildiğini görmekteyiz.

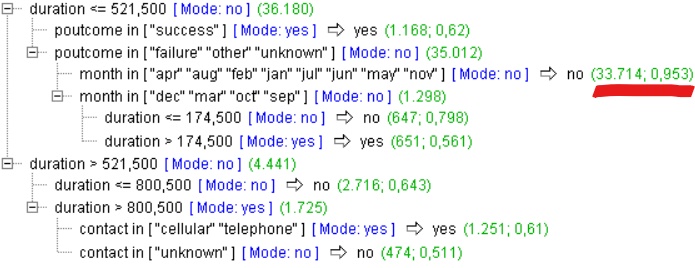
Şimdi C5 algoritmasının hangi değişkenleri önemli olarak gördüğüne bakalım.



Şekil 13 C5 Önem Dereceleri

%67’lik bir yüzde ile Duration yani müşteri ile geçirilen iletişim süresi en önemli değişken olarak belirlenmiştir.

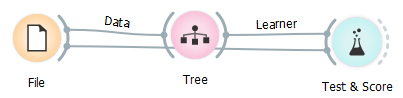
C5 algoritmasının karar ağacını incelediğimiz zaman ağacın büyüklüğünün çok fazla olduğunu görmekteyiz. Sınıflandırmaları birbirinden ayırırken ortaya çok fazla düğüm çıkmış bu da ağacı yorumlamayı zor bir hale getirmektedir. Bu nedenle Karar ağaçları içerisinde doğru tahmin yüzdesi olarak ikinci sırada yer alan CRT algoritmasının ağaç yapısını yorumlamamız daha mantıklıdır.



Şekil 14 CRT Karar Ağacı

Şekil 14 CRT Karar Ağacında altı kırmızı olarak çizilen nokta %95’i hayır olarak sınıflandırılan bir yapraktır. Bu sebeple bu noktanın yorumlanmasını uygun gördük. Buna göre saniyeler cinsinden son iletişim süresi 521’den düşük ve önceki pazarlama kampanyası sonucu başarılı olmayanlar ve bu aramanın mart, eylül, ekim ya da aralık ayı dışında yapılmış olanların %95’lik bir kısmının vadeli mevduatının olmadığını görmekteyiz.

Şimdi karar ağacı algoritmalarından Random Forest algoritmasını Orange üzerinden uygulayalım. Zaten Orange’ın yapmış olduğu tek karar ağacı algoritması Random Forest algoritmasıdır.



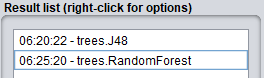
Şekil 15 Orange Karar Ağacı



Şekil 16 Orange Karar Ağacı

Şekil 16 Orange Karar Ağacında görüldüğü üzere CA yani Classification Accuracy %89,2 çıkmıştır. Diğer algoritmaların tahmin yüzdelerine göre daha düşük bir orandır.

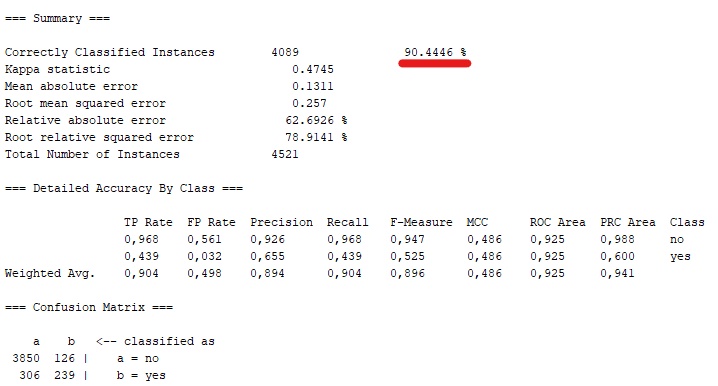
Şimdi Weka üzerinden uygulanmış algoritma yöntemlerine bakalım.



Şekil 17 Weka Karar Ağaçları

İki farklı algoritma denenmiş bunlardan J48 🡪 C 5 algoritmasıdır. Random Forest ise bilgimiz algoritmadan bir farkı yoktur.

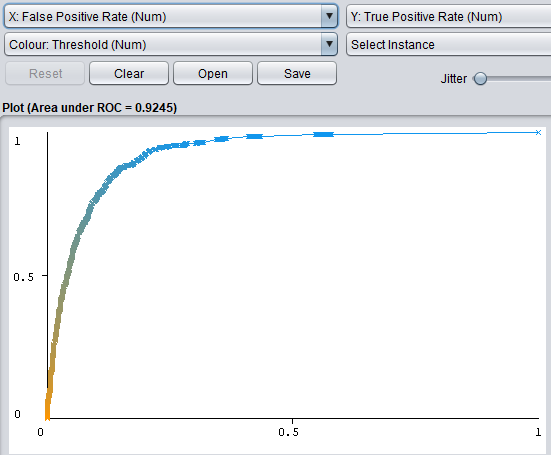
İki algoritma içinde Random Forest %90,44 ile en yüksek doğru tahmin oranına sahiptir.



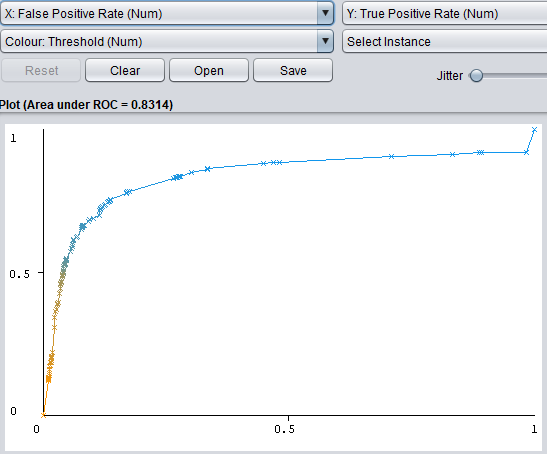
Şekil 18 Weka Random Forest

Random Forest algoritmasının Orange programına göre burada daha iyi bir tahmin yaptığını görmekteyiz.

Clementine’da ki C5 algoritması ile Weka’da ki Random Forest algoritması doğru tahmin yüzdesi olarak birbirlerine çok yakın çıkmıştır. Burada en iyi algoritmanın hangisi olduğuna karar vermek adına C5 ve Random Forest’ın Weka üzerinde ROC eğrilerini inceledik.



Şekil 19 Random Forest ROC Eğrisi



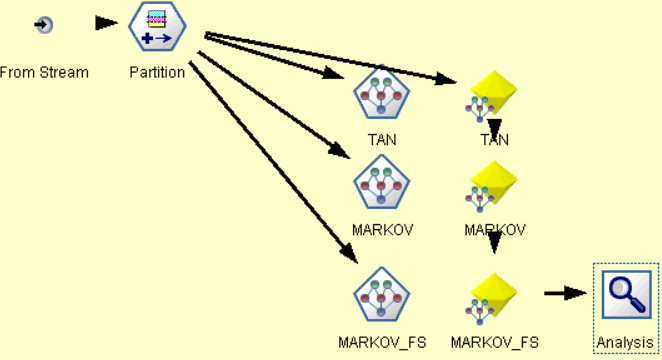
Şekil 20 C5 ROC Eğrisi

Grafikte görüldüğü üzere ROC altında kalan alan (Area under ROC) Random Forest algoritmasında %92.45, C5 algoritmasında %83.14’tür. Bu durumda karar ağaçları içerisinde geleceğe yönelik tahmin için en iyi algoritmanın Random Forest olduğuna karar verilmiştir.

#### 5.1.2 Bayesçi Ağlar

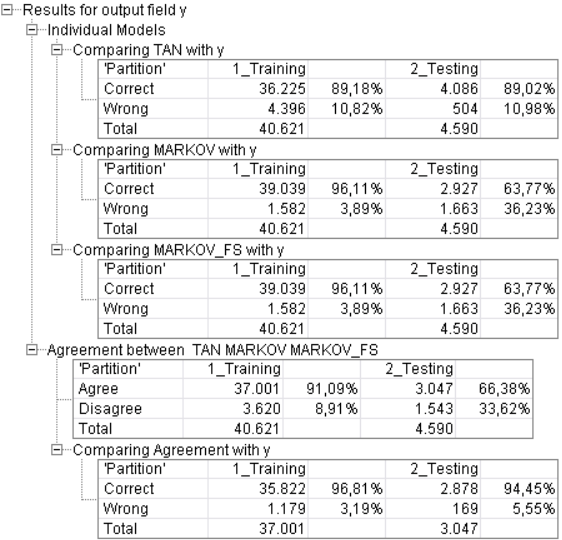
Bayesçi Ağlar kendi yapıları gereği İstatistikte ki Bayesçi Yaklaşımı kullanan algoritmalardır.

Burada kullanacağımız arama algoritmaları (search algorithm) TAN, MARKOV ve MARKOV FUTURE SELECTION’dır.



Şekil 21 Clementine Bayesçi Ağlar

İlk uygulama Clementine üzerinde yapılmış ve doğru tahmin yüzdeleri aşağıda ki gibidir.

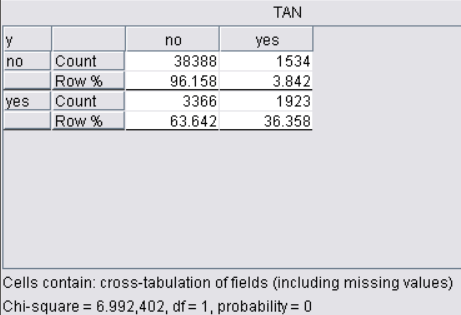


Şekil 22 Bayesçi Ağlar Doğru Tahmin Yüzdeleri

Şekil 22 Bayesçi Ağlar Doğru Tahmin Yüzdelerinde görüldüğü üzere TAN algoritması %89.02 ile en yüksek doğru tahmin oranına sahiptir. Burada MARKOV ve MARKOV\_FS algoritmalarının eğitim yüzdeleri çok yüksek, test yüzdeleri çok düşük çıkmıştır. Bunun sebebinin sistemin bu algoritmaları ezberlemiş olmasından kaynaklandığını söyleyebiliriz.

Burada açık ara farkla TAN algoritmasının önde olduğunu söyleyebiliriz. Dolayısıyla ROC eğrilerini incelemeye gerek görmedik.

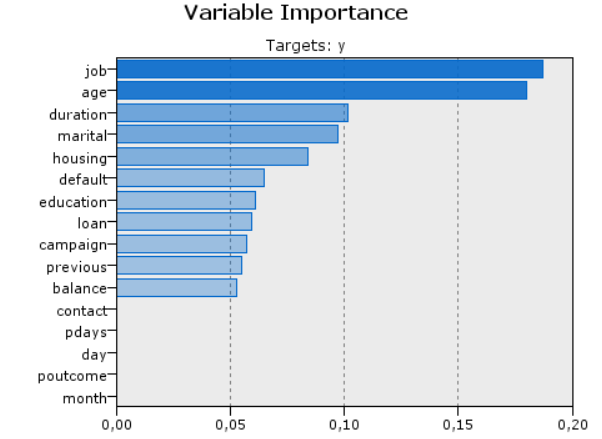
Aşağıda TAN algoritmasının Konfüzyon matrisi görülmektedir.



Şekil 23 TAN Konfüzyon Matrisi

Matriste görüldüğü üzere bu algoritma vadeli mevduatı olan kişileri %36 yani çok kötü bir yüzdeyle doğru tahmin etmiş. Buna karşılık vadeli mevduatı olmayan kişileri %96 gibi çok yüksek bir oranla doğru tahmin etmiştir.

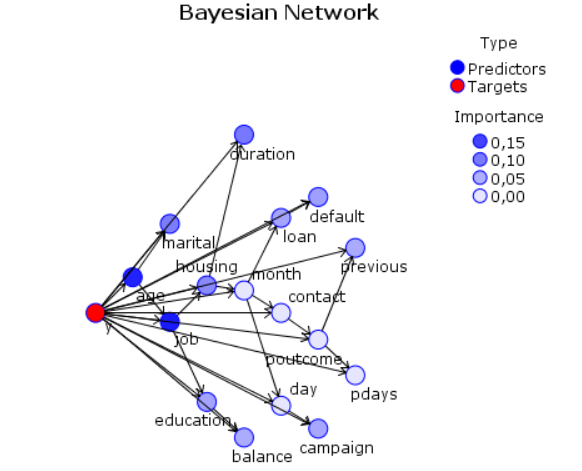
Şimdi TAN algoritmasının hangi değişkenleri önemli olarak gördüğüne ve hangi algoritmalar arasında bağlantı kurduğuna bakalım.



Şekil 24 TAN Algoritması Önem Sıralaması

Şekil 24 TAN Algoritması Önem Sıralamasında görüldüğü üzere en önemli 2 değişken age ve job olarak belirlenmiştir.

Şimdi bu algoritmaların önem sıralamasına göre nasıl bir ağ yapısı oluşturduklarına bakalım. TAN algoritması hedef değişkenin belirlediği önemli değişkenlerin kendi içlerinde de bağlantı kurmasına olanak sağlamaktadır. Kurulan ağ yapısının da bu şekle uygun olmasını beklemekteyiz.



Şekil 25 TAN Algoritması Ağ Yapısı

Şekil 25 TAN Algoritması Ağ Yapısında görüldüğü üzere en önemli 2 değişken age(yaş) ve job(iş) hedef değişkenle ilişkilendirilmiş. Daha sonra daha az olan önemli değişkenler gelmiştir ve bu değişkenler arasında da bağlantı kurulmasına izin verilmiştir.

Şimdi Orange üzerinden Bayesçi Ağ algoritmasını uygulayalım. Orange yalnızca tek bir algoritma yaklaşımına izin vermektedir oda çocukların (Child) kendi aralarında ilişki kurmasına izin vermemektedir.



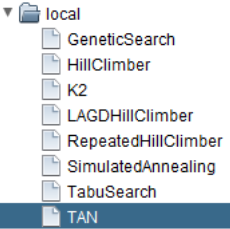
Şekil 26 Orange Bayesçi Ağ



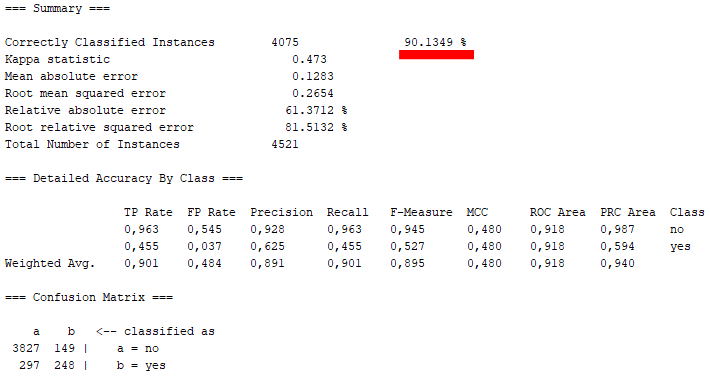
Şekil 27 Orange Bayesçi Ağ

Görüldüğü üzere Classification Accuracy değeri %87,4 çıkmıştır. Bu değer Clementine programında bulduğumuz değeri geçememiştir.

Şimdi Bayesçi Yaklaşım algoritmalarını Weka üzerinde deneyelim. Weka kendi bünyesinde Bayesçi Yaklaşım adına 8 farklı search algorithm (arama algoritması) kullanmaktadır.

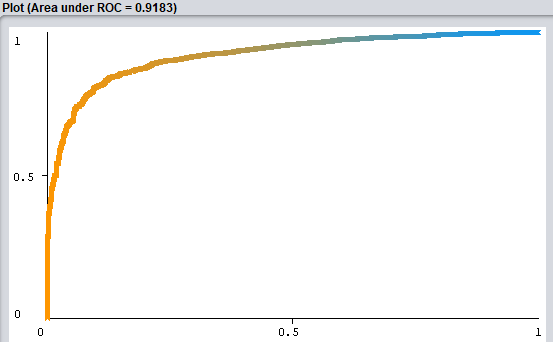


Bu algoritmaların hepsi denenmiş ve tıpkı Clementine üzerinde olduğu gibi TAN algoritması en iyi sonucu vermiştir. Ek olarak Weka, Clementine programına göre daha iyi bir doğru tahmin yüzdesine ulaşmıştır.



Şekil 28 Weka TAN Algoritması

Aynı algoritmanın Weka üzerinde %90,13 ile Clementine’a göre çok daha iyi bir tahmin yaptığını görmekteyiz. Detaylar kısmında bu algoritmanın ROC Eğrisi grafiği altında kalan alanının %91,8 olduğunu görmekteyiz. Fakat biz yine de ROC eğrisi grafiğini Weka üzerinden çizdirelim.



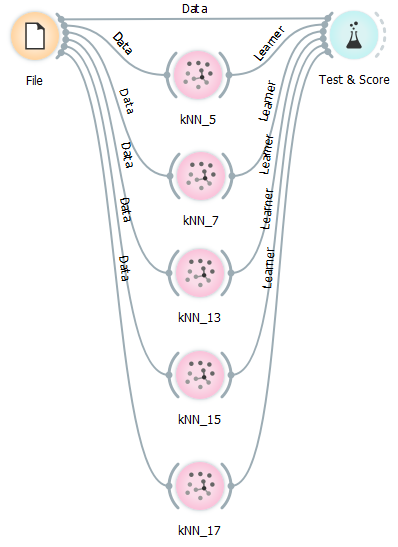
Şekil 29 TAN ROC Eğrisi

Bu bilgiler ışığında Bayesçi Ağlar ile geleceğe yönelik tahmin yapmak istersek kullanacağımız arama algoritması TAN’dır.

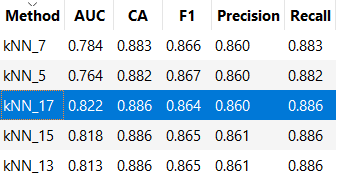
#### 5.1.3 K- En Yakın Komşu Algoritması

K en yakın komşu algoritması temelinde tahmin edilmek istenen değere en yakın olan k tane sınıflandırmaya bakarak kendisinin bunlardan hangisine dahil olduğunu anlamaya çalıştığı bir yaklaşımdır.

Bu algoritma Clementine’da bulunmadığı için Modeler üzerinden denenmiş fakat Execution (Çalışma) süresi diğer programlara göre çok daha uzun sürdüğü görülmüştür. Bu sebeple bu algoritma denemeleri sadece Weka ve Orange üzerinden yapılmıştır. K değerleri 5, 7, 13, 15 ve 17 olarak seçilmiş ve tahminler bu k değerlerine göre yapılmıştır.



Şekil 30 Orange K En Yakın Komşu



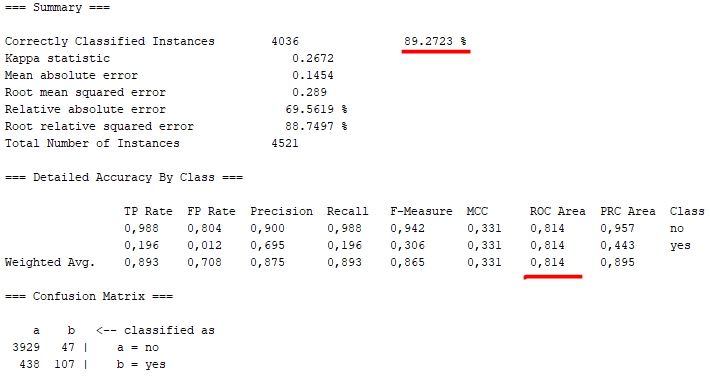
Şekil 31 Orange K En Yakın Komşu

Şekil 31 Orange K En Yakın Komşuda görüldüğü üzere k değeri 15 ve 17 olan değerlerin Classification Accuracy değerleri eşit çıkmıştır.

Fakat kNN\_17’nin ROC eğrisi altında kalan alanı (AUC) %82,2 ile diğer hepsinden yüksek çıkmıştır.

Dolayısıyla en doğru tahminin %88,6 ile Orange üzerinde Knn\_17 olacağını söyleyebiliriz.

Bundan sonra ki adımda Weka üzerinde k değerleri 13, 15 ve 17 denenmiş ve yine Knn\_17 en yüksek değer çıkmıştır. Sonucu aşağıda ki gibidir.



Şekil 32 Weka kNN\_17

Şekil 32 Weka kNN\_17’de görüldüğü üzere doğru tahmin oranı %89,27’dir. ROC Area ise %81,4 olarak hesaplanmıştır.

Weka ve Orange programlarının aynı algoritma üzerinde ki doğru tahmin ve ROC Eğrileri yüzdeleri aşağıda ki gibidir.

|  |  |  |
| --- | --- | --- |
|  | Doğru Tahmin Oranı | ROC Eğrisi |
| Weka | %89,27 | %81,4 |
| Orange | %88,6 | %82,2 |

Şekil 33 K En Yakın Komşu Karşılaştırma

Şekil 33 K En Yakın Komşu Karşılaştırma tablosunda görüldüğü üzere Weka doğru tahmin yüzdesinde, Orange ise ROC Eğrisinde daha yüksek çıkmıştır. Burada ROC Eğrisinin tahminler için daha güçlü olduğunu düşündüğümüzden Orange programında ki Knn\_17 algoritmasının en iyisi olduğuna karar verdik.

#### 5.1.4 Yapay Sinir Ağları

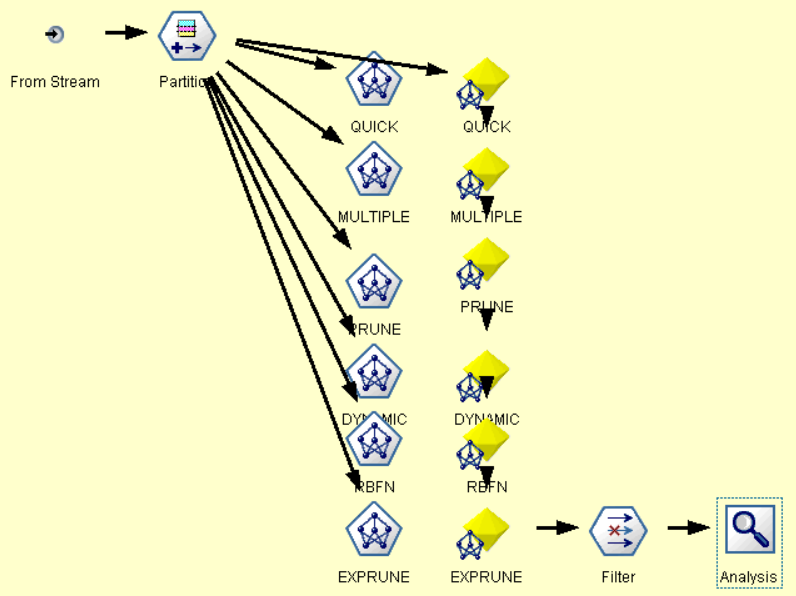
Yapay sinir ağları, tıpkı beynimizde ki gibi giriş ve çıkış katmanlarıyla nöronlar oluşturan ve bunları birbiri arasında ilişkilendirerek öğrenmeyi amaçlayan bir algoritmadır. Daha yeni ve etkili bir yöntem olduğu için çoğu zaman diğer algoritmalara göre daha iyi sonuçlar verebilmektedir.

Clementine bize 6 farklı sinir ağı metodu sunmaktadır. Bunlar QUICK, MULTIPLE, PRUNE, DYNAMIC, RBFN ve EXPRUNE’dur.

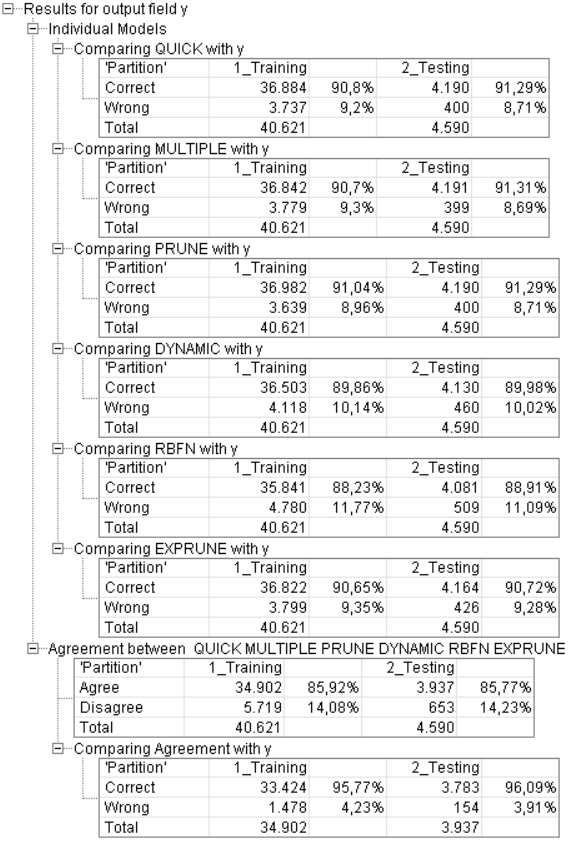
Orange ise bize sinir ağlarını oluştururken kullanılan Aktivasyon Fonksiyonlarını değiştirme olanağı sağlamaktadır. Bunlar LOGISTIC, IDENTITY, TANH ve RELU’dur.

Weka, yukarıda ki faktörlere izin vermemektedir. Yalnızca saklı katmanları (hidden Layer) değiştirmemize olanak vermektedir.

İlk olarak Clementine uygulamasında ki doğru tahmin yüzdelerine bakalım.



Şekil 34 Clementine Sinir Ağları

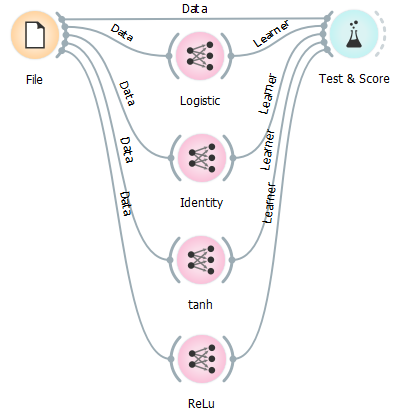


Şekil 35 Clementine Sinir Ağları

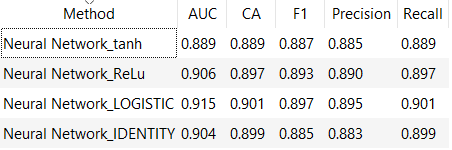
Şekil 35 Clementine Sinir Ağlarında görüldüğü üzere QUICK, MULTIPLE ve PRUNE algoritmalarının testing değerleri birbirlerine çok yakın çıkmıştır.

Clementine, ROC Eğrilerini oluşturamadığı için ikinci bir karşılaştırma yapmaksızın en yüksek doğru tahmin oranına ulaşmış olan MULTIPLE algoritması en iyi algoritma seçilmiştir.

Şimdi Orange üzerinden yapılan uygulamalara bakalım.



Şekil 36 Orange Sinir Ağları



Şekil 37 Orange Sinir Ağları

Şekil 37 Orange Sinir Ağlarında görüldüğü üzere LOGISTIC aktivasyon fonksiyonu hem sınıflandırma yüzdesi olarak hem de ROC eğrisi altında kalan alan olarak en yüksek değere sahiptir. Orange üzerinde bir sınıflandırma tahmini yapmak istersek LOGISTIC Sinir Ağını kullanacağız.

Weka üzerinden yapılan Sinir ağı, Modeler uygulamasının da kullandığı Multilayer Perceptron yani istediğimiz kadar tabaka (Layer) kullanmamıza olanak sağlayan bir yaklaşım kullanmaktadır.

İlk başta saklı tabaka (hidden Layer) sayısı programın otomatik belirlediği sayı olarak alınmış fakat iyi bir sonuç elde edilememiştir. Daha sonra bu tabakaların sayısı 4 ve 6 olarak değiştirilmiştir.



Şekil 38 Weka Saklı Katmanlar

Tabaka sayıları değiştirildikten sonra elde edilen sonuç aşağıda ki gibidir.



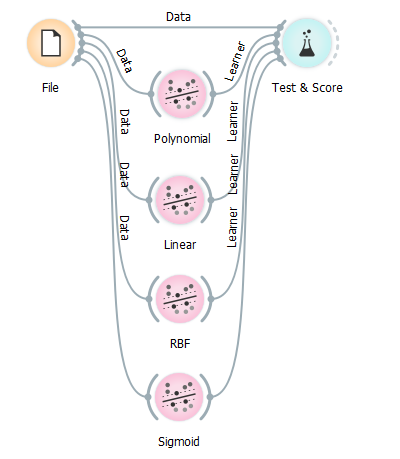
Şekil 39 Weka Sinir Ağları

Görüldüğü üzere Weka bize %90,33’lük bir doğru tahmin oranı sunmuştur.

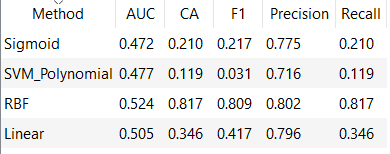
Geleceğe yönelik bir tahmin yapmak istersek bütün programlar içerisinde en yüksek doğru tahmin oranı olan Clementine’ın MULTIPLE algoritması kullanılacaktır.

#### 5.1.5 Destek Vektör Makineleri

Destek vektör makineleri yalnızca Orange üzerinde denenmiştir. Algoritmalarda kullanılan matematiksel fonksiyonlara göre Destek Vektör Makineleri; LINEAR, POLYNOMIAL, RBF ve SIGMOID olmak üzere 4’e ayrılmaktadır.



Şekil 40 Orange SVM



Şekil 41 Orange SVM

Şekil 41’de görüldüğü üzere açık ara farkla Destek Vektör Makineleri için en iyi algoritma RBF’dir.

#### 5.1.6 Sonuç ve Gelecek Tahmini

|  |  |  |
| --- | --- | --- |
| **Yöntem** | **Algoritma** | **Doğru Tahmin Yüzdesi** |
| Karar Ağaçları | Random Forest | %90,44 |
| Bayesçi Ağlar | TAN | %90,13 |
| K- En Yakın Komşu | Knn\_17 | %88,6 |
| Yapay Sinir Ağları | MULTIPLE | %91,31 |
| Destek Vektör Makineleri | RBF | %81,7 |

Şekil 42 Sonuç Tablosu

Şekil 42 Sonuç Tablosunda görüldüğü üzere yapay sinir ağlarının en iyi algoritma olduğuna karar verilmiştir.

Şimdi bu algoritmayı kullanarak daha önce vadeli mevduatının olup olmadığı bilinmeyen bir kişi için bilgisayara tahmin yaptıralım. En iyi algoritma Clementine tarafından bulunmuştu dolayısıyla tahmini de Clementine üzerinden yaptıracağız.



Şekil 43 Gelecek Tahmin Verileri

Şekil 43 Gelecek Tahmin Verilerinde y değişkeni boş bırakılmış ve bilgisayarın MULTIPLE algoritması ile tahmin etmesi sağlanmıştır.



Şekil 44 Tahmin Sonucu

Görüldüğü üzere bilgisayar girilen nitelikler (Attribute) doğrultusunda bu kişinin bir vadeli mevduatının olmayacağını öngörmüştür.

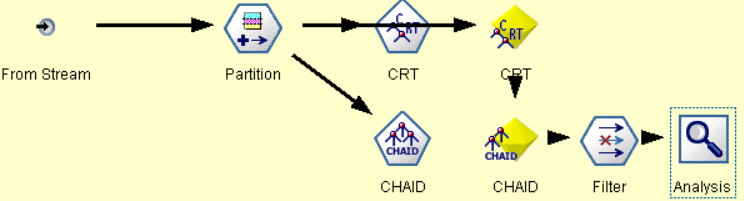
### 5.2 Regresyon Tahmini

Regresyon tahmini elimizdeki sürekli verinin geleceğe yönelik tahmini anlamına gelir. Burada veri kümesinde ki ‘balance’ yani ortalama yıllık bakiyenin tahmini yapılacaktır. Kullanılacak algoritmalar Karar Ağaçları, Sinir Ağları, Doğrusal Regresyon, K En Yakın Komşu ve Destek Vektör Makineleridir. K En Yakın Komşu algoritması yalnızca Orange üzerinde denenmiş geri kalanlar aynı zamanda Clementine’da çalıştırılmıştır.

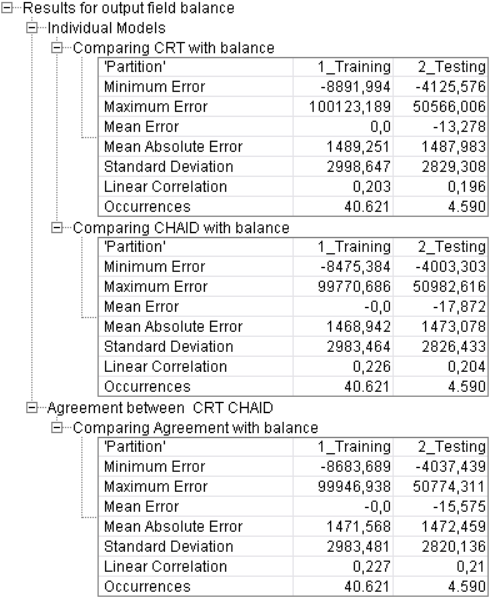
Burada sınıflandırıcı tahmin için kullandığımız ‘y’ değişkenini önemsiz bulduğumuzdan her bir algoritma için bu değişkeni filtreledik.

#### 5.2.1 Karar Ağaçları

Karar ağaçları içerisinde sürekli veriler için kullanabileceğimiz 2 algoritma bulunmaktadır. Bunlar CRT ve CHAID algoritmalarıdır.



Şekil 45 Karar Ağaçları Regresyon Tahmini

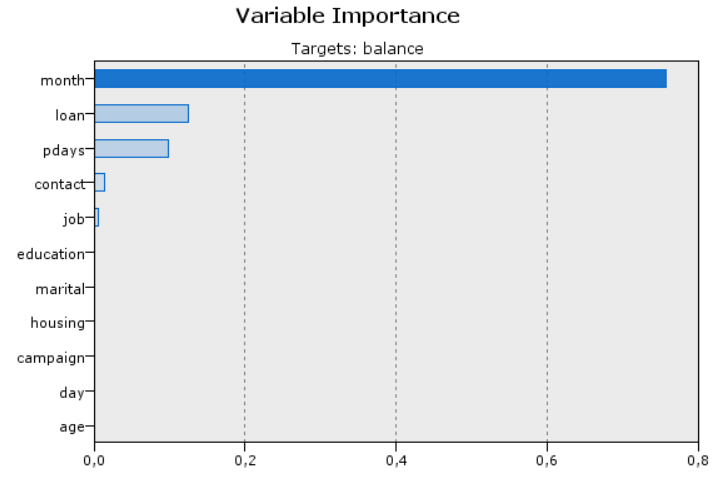


Şekil 46 Karar Ağaçları Regresyon Tahmini

Şekil 46 Karar Ağaçları Regresyon Tahmininde görüldüğü üzere Sınıflandırıcı Tahmine göre çıktımız biraz daha farklıdır. Burada bizim için önemli olan kısım ‘Mean Absolute Error’ yani mutlak ortalama hata kısmıdır. Bu değer bir hata değeri olduğu için olabildiğince küçük olmasını isteriz.

Dolayısıyla CHAID algoritması 1473 ile en küçük hataya sahiptir. Clementine üzerinde karar ağaçlarında bir sürekli veri tahmini yapmak istersek CHAID algoritmasını kullanacağız.

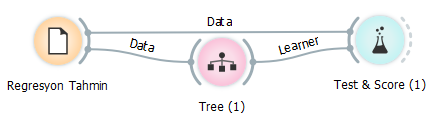
Şimdi bu algoritmanın hangi öznitelikleri en önemli olarak gördüğüne bakalım.



Şekil 47 CHAID Algoritması Önem Sıralaması

Şekil 47 CHAID Algoritması Önem Sıralamasında görüldüğü üzere CHAID algoritması ‘month’ yani ay değişkenini önemli olarak görmüş ve diğer algoritmaları neredeyse hiç gerek duymamıştır.

Şimdi Orange üzerinden Karar Ağacını uygulayalım. Bilindiği üzere Orange karar ağaçları içerisinden yalnızca Random Forest algoritmasını uygulamaktadır.



Şekil 48 Orange Karar Ağaçları



Şekil 49 Orange Karar Ağaçları

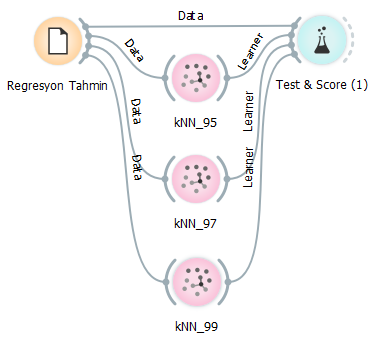
Şekil 49’da bizi ilgilendiren kısım MAE (Mean Absolute Error) kısmıdır. Bu değer 1901 çıkmıştır. Clementine’ın bulduğu sonuçlardan çok daha kötü bir sonuçtur.

Bu bilgiler ışığında Karar ağaçları içerisinde bir Regresyon Tahmini yapmak istersek kullanacağımız algoritma CHAID algoritmasıdır.

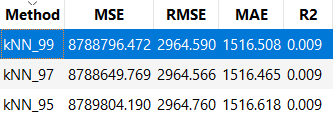
#### *5.2.2 K En Yakın Komşu Algoritması*

Bu algoritma Clementine üzerinde bulunmadığı için bu sefer yalnızca Orange üzerinde denenmiştir. K sayısını ne kadar arttırırsak hatanın o kadar düştüğünü fark ettik. Fakat bir süre sonra hatanın giderek daha az düşmeye başladığını gördük. Bu durumda k sayısını daha fazla arttırmak anlamsız olurdu.

Bu sebeple k sayısının 95, 97 ve 99 olduğu değerler Orange üzerinde gösterilmiştir.



Şekil 50 Orange K En Yakın Komşu

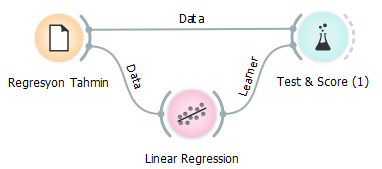


Şekil 51 Orange K En Yakın Komşu

Şekil 51 Orange K En Yakın Komşu tablosunda görüldüğü üzere k sayısını arttırdıkça hatada çok az da olsa bir düşüş görülmektedir. Bu sebeple geleceğe yönelik tahmin yapmak istersek kullanacağımız algoritma Knn\_99 olacaktır.

#### 5.2.3 Doğrusal Regresyon Algoritması

Doğrusal regresyon algoritması, sürekli verileri tahmin etmek için en temel yöntemlerden biridir. Hatta bütün algoritmaların babası denilebilir. Algoritmayı yalnızca Orange üzerinde denedik.



Şekil 52 Orange Doğrusal Regresyon

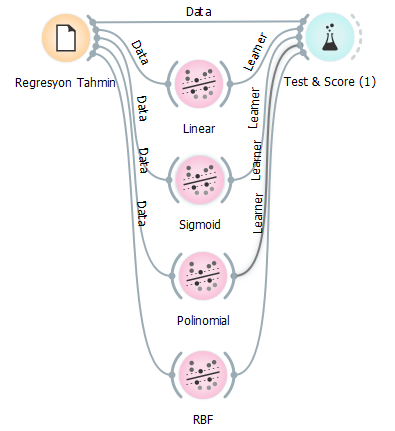
C:\Users\user\Desktop\Ekran Alıntısı.PNG

Şekil 53 Orange Doğrusal Regresyon

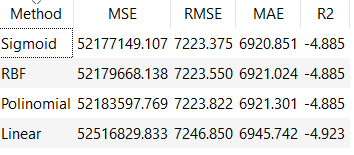
Görüldüğü üzere hata değerimiz 1472 çıkmıştır. İlkel bir yöntem olmasına karşın Orange’ın bulduğu diğer çoğu algoritmadan daha iyi bir sonuç elde ettik.

#### 5.2.4 Destek Vektör Makineleri

Destek Vektör Makineleri, yine sadece Orange üzerinde denenmiştir.



Şekil 54 Orange SVM

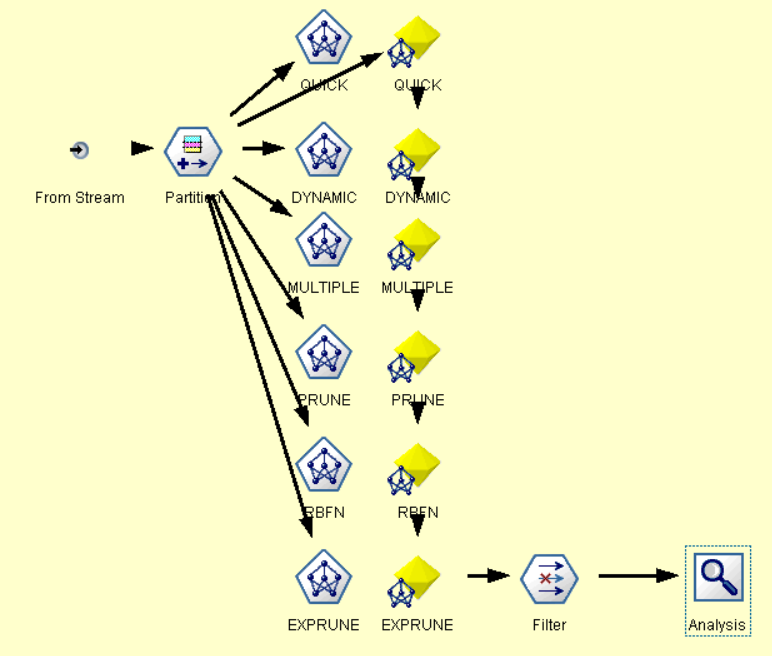


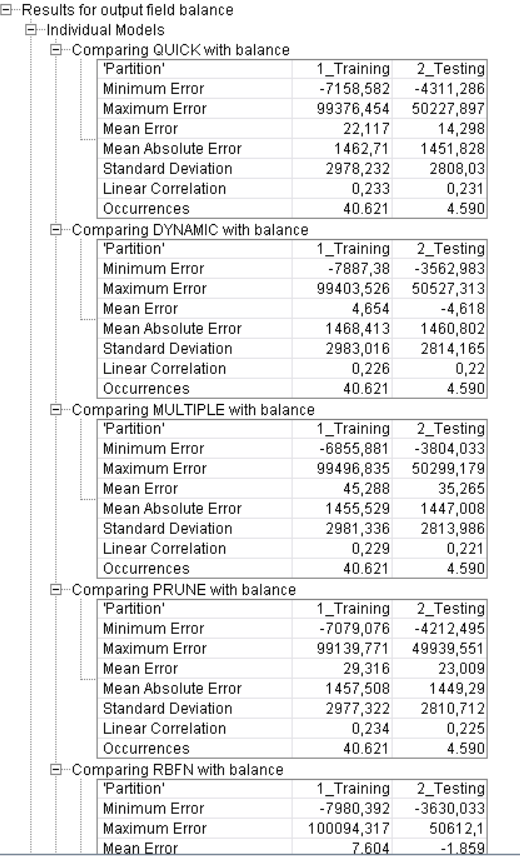
Şekil 55 Orange SVM

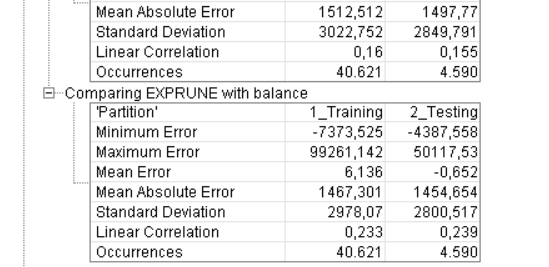
Görüldüğü üzere Destek Vektör Makineleri tıpkı sınıflandırma tahmininde olduğu gibi burada da çok kötü bir sonuç elde etmiştir. Bu sebeple yorumlamaya ihtiyaç duymadık.

#### 5.2.5 Yapay Sinir Ağları

Yapay sinir ağlarının, sınıflandırıcı tahminde olduğu gibi burada da iyi bir tahmin edici olmasını bekliyoruz.



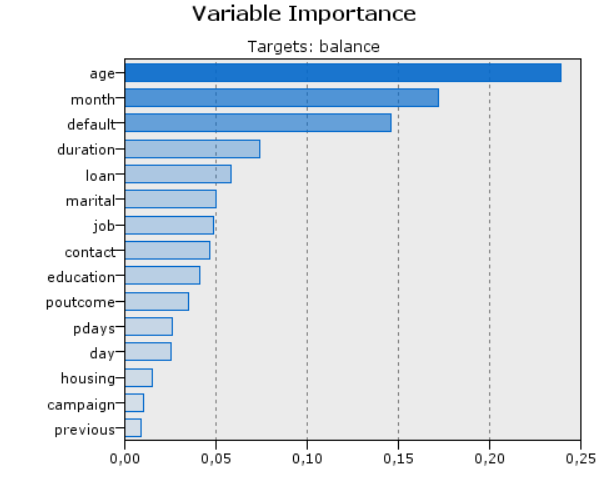




Şekil 56 Clementine Sinir Ağları

Şekil 56 Clementine Sinir Ağlarında denenmiş bütün algoritmaların sonuçları görülmektedir. Çoğu sinir ağı algoritmasının hatası birbirine yakın çıkmıştır. Fakat MULTIPLE algoritması 1446,008 ile birinci sıradadır.

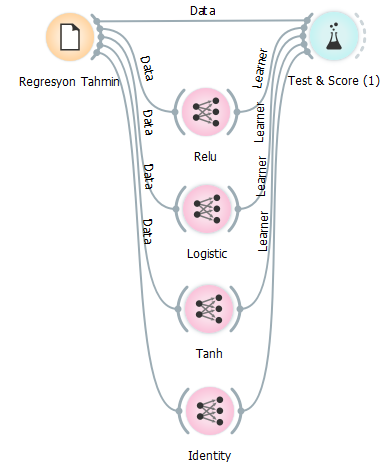
MULTIPLE algoritmasının sürekli bir değerin tahmini için hangi değişkenleri önemli olarak gördüğüne bakalım.

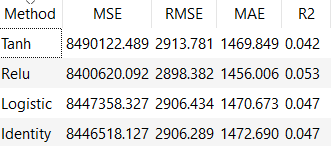


Şekil 57 Sinir Ağları Önem Dereceleri

Şekil 57 Sinir Ağları Önem Derecelerinde görüldüğü gibi yaş değişkeni bizim için en önemli değişkendir. Daha sonra ay ondan sonra da müşterinin kredi çekip çekmediği gelmektedir.

Şimdi Orange üzerinden tahmin yapalım.





Şekil 58 Orange Sinir Ağları

Şekil 58 Orange Sinir Ağlarında en iyi algoritmanın hatası 1456 olan RELU olduğunu görmekteyiz fakat bu hata Clementine’da ki MULTIPLE algoritmasının hatasından daha yüksektir.

Bu nedenle sinir ağları ile geleceğe yönelik bir tahmin yapmak istersek MULTIPLE algoritmasını kullanacağız.

#### 5.2.6 Sonuç ve Gelecek Tahmini

|  |  |  |
| --- | --- | --- |
| **Yöntem** | **Algoritma** | **Mutlak Hata** |
| Karar Ağaçları | CHAID | 1473,08 |
| K En Yakın Komşu | Knn\_99 | 1516,508 |
| Doğrusal Regresyon | Regresyon | 1472,642 |
| Destek Vektör Makineleri | RBF | 6921,024 |
| Yapay Sinir Ağları | MULTIPLE | 1446,008 |

Şekil 59 Sonuç Tablosu

Şekil 59 Sonuç Tablosunda görüldüğü üzere yapay sinir ağlarından MULTIPLE algoritması en düşük hataya sahiptir.

Şimdi bu algoritmayı kullanarak geleceğe yönelik tahmin yapalım.



Şekil 60 Gelecek Tahmin Verileri

Şekil 60 Gelecek Tahmin Verilerinde balance yani hedef değişkenimiz boş bırakılmıştır. Diğer değişkenlere yeni kullanıcı için değerler girilmiştir.



Şekil 61 Tahmin Sonucu

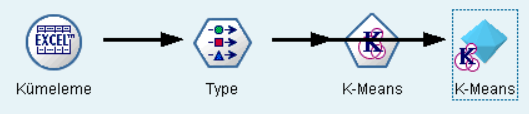
Şekil 61 Tahmin Sonucunda Clementine’da boş olarak bırakılmış değerin tahmin sonucu gözükmektedir.

Girilen özniteliklere sahip bir kişinin yıllık ortalama bakiyesi 897,415 Euro olacaktır.

### 5.3 Kümeleme

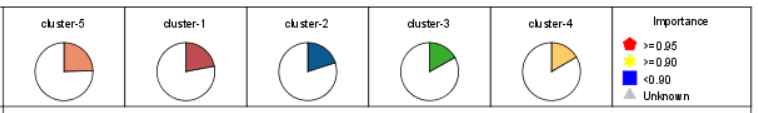
Kümeleme analizinde 3 farklı algoritma vardır. Bunlar K-Means, Kohonen ve TwoStep’tir. 3 yöntemde Clementine’da incelenmiştir.

#### 5.3.1 K-Means



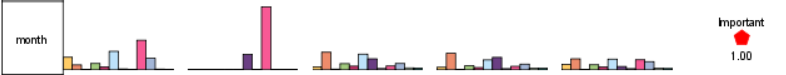
Şekil K-Means Kümeleme

Şekil 62 K-Means Kümeleme resminde görüldüğü gibi veri kümesine 1 adet type nodu bağlanmış ve bu nod aracılığı ile ‘y’ değişkeni dışında ki bütün değişkenler girdi olarak belirtilmiştir.



Şekil K-Means Kümeleme

Yukarıda ki grafikte algoritmanın çalışma sonucu 5 tane küme oluşturduğunu görmekteyiz. Kümelerin değişkenler üzerinde ki etkilerine baktığımızda ay değişkeni dikkatimizi çekti.



Şekil K-Means Kümeleme Sonucu

Grafiklere göre Cluster-1’de verilerin büyük bir kısmının mayıs ayına denk geldiği görülmektedir.

#### 5.3.2 TwoStep

Bu algoritmanın oluşturulma şekli K-Means’de olduğu gibidir. Maksimum 15 minimum 2 tane küme oluşturması istenmiştir.



Şekil TwoStep Kümeleme

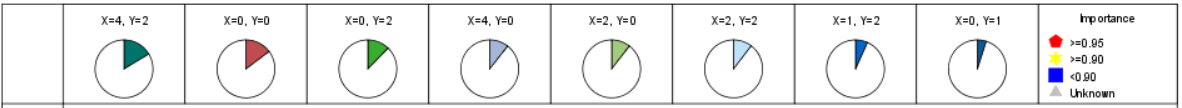
Şekil 65 TwoStep Kümelemede algoritmanın bize 2 tane küme oluşturduğunu görmekteyiz.



Şekil TwoStep Kümeleme

Grafiklerde görüldüğü üzere sarı kutucuğun birinci kümede ikinci kümeye göre daha yüksek çıktığı görülmekte. Bu durumda iletişim türünün hücresel olduğu kişilerin birinci kümede yoğunlaştığını söyleyebiliriz.

#### 5.3.3 Kohonen Ağlar

Bu algoritmanın da oluşturulma şekli K-Means’de olduğu gibidir.

Şekil Kohonen Ağlar Kümeleme

Kohonen ağlar bize 8 farklı küme oluşturmuştur.



Şekil Kohonen Ağlar Kümeleme Sonucu

Şekil 68 Kohonen Ağlar Kümeleme Sonucunda yeşil sütunun(Yükseköğretim) x=0, y=0 noktasında neredeyse %100 etkili olduğu görülmektedir. Yani bu küme eğitim durumu yükseköğretim olan kişiler üzerinde kullanılabilir yorumunu yapabiliriz.

**Şükriye Nur Şencan – 21622152**

**Mert Yanık - 21522134**