

T.C.
FIRAT ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ

Projenin Raporlanması

Manisa Şehiri Temiz Hava Tahmini

7.Hafta - Rapor

Proje Ekibi

Erdoğan DAĞLI

Haziran 2020

Bu hafta yapmış olduğum

Araştırmalarım :

- Modelleme yapabilmek için literatür araştırması yaptım.
- Daha öncede kurslarda da denemiş olduğum modellemeleri denedim
- Bunlara ek olarak datasetimi uygun hale getirmek için birkaç araştırma yaptım.

Yaptıklarım :

- Geçen haftaki yapmış olduğum feature metriklerini belirlemiştim.
- Bu belirlenen metriklere uygun bir şekilde sonuç sütunumu ikilik sistemde almayı ve modellememde buna uygun bir şekilde tanımlamayı hedefledim.
- Ortalama hava kalitesini hesaplarken 5'den 0 doğru kalite sıralaması yapmıştım.
 - Bunların 3 den düşün olanları 0 yani kalitesiz olarak tanımladım
 - Geri kalan 2 den büyük olanları ise 1 yani kaliteli olarak tanımladım.
- Korelasyon grafiği , standartlaştırma, normalizasyon gibi kavramlardan da yararlanmaya çalıştım.
- En son işlem olarak ise kodlamalarımı github a committedim.

Kod bloklarım:

```
from sklearn.model_selection import train_test_split, cross_val_score, cross_val_predict
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import StandardScaler, RobustScaler
#df=makeStandartion()
df = addFeature()
label_encoder = LabelEncoder().fit(df.binarySonuc)
labels = label_encoder.transform(df.binarySonuc)
classes = list(label_encoder.classes_)

y = df.drop("binarySonuc", axis=1)
X = labels
nb_features = 21
nb_classes = len(classes)

from sklearn.utils import shuffle
X, y = shuffle(X, y)
X = np.asarray(X).astype(np.float32)

###
#Normalization
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler().fit(df.binarySonuc)
X = scaler.fit_transform(df.binarySonuc)
###Variladition
from sklearn.model_selection import train_test_split
X_train, X_valid, y_train, y_valid = train_test_split(df, labels, test_size=0.3)
###Category Part
from tensorflow.keras.utils import to_categorical
y_train = to_categorical(y_train)
y_valid = to_categorical(y_valid)
```

Şekil 1 : Modelleme Öncesi Yapılan Hazırlıklar

```

#Model
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Activation,SimpleRNN,Dropout,MaxPooling1D,Flatten,BatchNormalization

model = Sequential()
model.add(SimpleRNN(2048,input_shape=(nb_features,1)))
model.add(Activation("relu"))
model.add(Dropout(0.25))
model.add(BatchNormalization())
model.add(Flatten())
model.add(Dense(2048,activation="relu"))
model.add(Dense(1024,activation="relu"))
model.add(Dense(nb_classes,activation="sigmoid"))
model.summary()

###
#Compile and Training
from tensorflow.keras.optimizers import SGD
opr = SGD(lr=1e-3,decay = 1e-5,momentum=0.9 , nesterov = True)
model.compile(optimizer=opr, loss='binary_crossentropy', metrics=['accuracy'])
score = model.fit(X_train,y_train,epochs=50,validation_data=(X_valid,y_valid))

```

Şekil 2 : Modelleme Kısmı