

Project : Youtube Adview Prediction

8. Build an artificial neural network and train it with different layers and hyperparameters. Experiment a little. Use keras.

At the end we can play around with neural networks using the keras library which is quite similar to the scikit learn library. Here we can define the model architecture (layers, number of neurons and activation functions), optimisation algorithm (like gradient descent or adam), cost function (mean square error) and give in the dataset. Then the model trains for different epochs (going through dataset once means one epoch) to result in an improved model. We may need to perform hyperparameter tuning (i.e. selecting the best hyperparamters like number of neurons or activation function to yield minimum error).

We need to accept the fact that in some cases neural network may not perform better than other machine learning models.

- ☐ Import keras
- ☐ Define model architecture
- ☐ Define optimisation algorithm and cost function
- ☐ Train the neural network
- ☐ Calculate errors

```
# Artificial Neural Network
import keras
from keras.layers import Dense

ann = keras.models.Sequential([
    Dense(6, activation="relu",
          input_shape=X_train.shape[1:]),
    Dense(6, activation="relu"),
    Dense(1)
])

optimizer=keras.optimizers.Adam()
loss=keras.losses.mean_squared_error
ann.compile(optimizer=optimizer, loss=loss, metrics=["mean_squared_error"])
```

```
history=ann.fit(X_train,y_train,epochs=100)

ann.summary()

print_error(X_test,y_test,ann)
```

9. Pick the best model based on error as well as generalisation.

Now that we have different machine learning models and their respective errors, we can select the one with minimum error or any other selection scheme. It's important to make sure there is no overfitting and the model generalises well. We use the test data we got from splitting the original dataset (which acts as validation set in our case).

- ☐ Compare errors for each model
- ☐ Select the most suitable ones
- ☐ Use F1 Score and RC Curves (Bonus)

10. Save your model and predict on test set.

Saving the models can be done through both keras and scikitlearn. This helps us to load the model again or share it with others. At the end, we use completely unknown data from 'test.csv' to make our final predictions for evaluation from our chosen model in the previous step.

- ☐ Save models
- ☐ Make predictions

```
#Saving Scikitlearn models
import joblib
joblib.dump(decision_tree, "decisiontree_youtubeadview.pkl")

# Saving Keras Artificial Neural Network model
ann.save("ann_youtubeadview.h5")
```