Internship Report – Internship Studio

Internship in Machine learning

Mahesh Ravindra Chaudhari

3rd year Department of Engineering Design

Indian Institute of Technology Madras

Sept 2020 – Oct 2020

My contact details

+91 8329847222

chaudharimahesh2000@gmail.com

linkedin.com/in/mahesh-chaudhari-498a26183

❖ **Table** containing all the models and their **mean square error**. **Highlight** the chosen model and give reasoning

Model name / Error name	Linear Regressor	Decision Tree Regressor	Random Forest Regressor	Support Vector Regressor
Mean absolute error	3707.378005824532	2567.5536202185795	3238.281330066957	3707.378005824532
Mean squared error	835663131.1210337	874222779.6089481	574223937.3404809	835663131.1210337
Root mean squared error	28907.83857573986	29567.258574459487	23962.97012768828	28907.83857573986

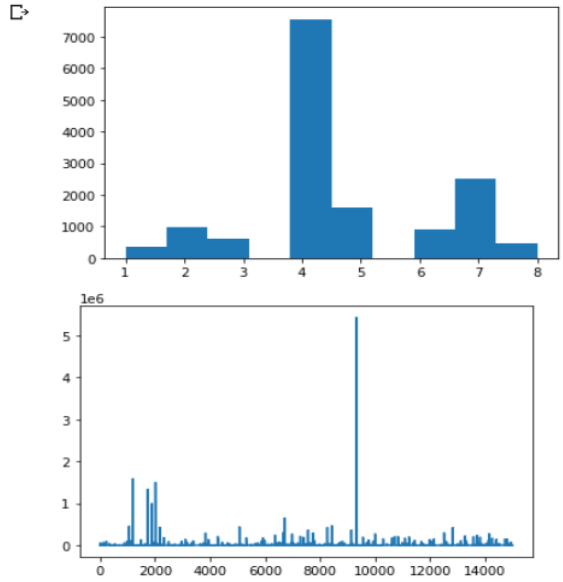
➢ **Chosen Model**
Random Forest Regressor

➢ **Reason behind choosing the model**
When we run our data through all the models and calculate mean errors between Y_test and its corresponding prediction we get minimum mean squared error in Random forest regressor model. Lesser error reflects more accuracy of the model also we know that Random forest is a combination of many decision trees so it will show more accuracy than decision tree and linear regression.

❖ **Plots and graphs**

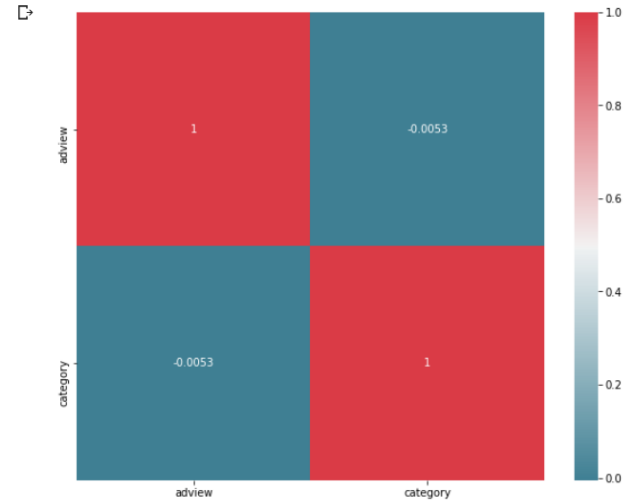
1) Training dataset: 1st one is frequency of videos in each category whereas 2nd one frequency of number of adviews

```
# Visualization
# Individual Plots
plt.hist(data_train["category"])
plt.show()
plt.plot(data_train["adview"])
plt.show()
```



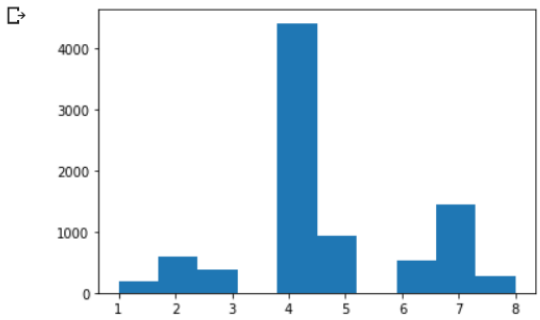
2) Training dataset: Graph of correlation between advise and category after removing outliers

```
# Remove videos with adview greater than 2000000 as outlier
data_train = data_train[data_train["adview"] <2000000]
# Heatmap
import seaborn as sns
f, ax = plt.subplots(figsize=(10, 8))
corr = data_train.corr()
sns.heatmap(corr, mask=np.zeros_like(corr, dtype=np.bool), cmap=sns.diverging_palette(220, 10, as_cmap=True),
            square=True, ax=ax,annot=True)
plt.show()
```



3) Test dataset: Frequency of videos in each category

```
# Visualization
# Individual Plots
plt.hist(data_test["category"])
plt.show()
```



❖ Stepwise explanation of the project

➢ Step 1

First of all we import preinstalled python libraries and packages like numpy, pandas, matplotlib and seaborn for data cleaning and visualisation. Then we import train.csv file using the pandas library as a pandas dataframe, check shape of the dataframe usning .shape command. Values in the category column are given in string format (A to H), so basically we assign them with numbers ranging from 1 to 8.

➢ Step 2

We visualise and study the dataset and its distributions by plotting frequency of videos in each category, frequency of number of adviews, heatmaps.

➢ Step 3

Here in this step we are removing the videos having character 'F' in its columns. Also the given data is in string form so we are converting the given data to numeric form by using *pd.to_numeric* and *LabelEncoder()* function.

➢ Step 4

Convert date of publishing and duration in coded numerical format. We do all this because we need to have all the data in numerical format before putting it any sort of ML model. Column of video ID (vidid) and adview are separated from the man dataset and the remaining dataset is used as an independent variables in the following models.

➢ Step 5

The data is split into training and testing data by using *train_test_split* command. After that data is normalised to make it more concise and to get better results too.

➢ Step 6

All the errors have been put in the function named print_error and below that *Linear Regression model* and *Support Vector Regressor model* have been run to get the errors in the respective models.

- **Step 7**

  Here we have run the data through *Decision Tree Regressor* and *Random Forest Regressor* and checked the errors that we are getting. I also found out that we get least mean squared error in the *Random Forest Regressor* model.

- **Step 8**

  Here we built artificial neural network and trained it in different layers and hyper parameters using *keras.layers*

- **Step 9**

  Here I picked the best model that I could get from the above four model after comparing their mean squared error. Lesser the mean squared error more the accuracy of the model!

- **Step 10**

  Hereafter I have treated the test.csv file the same way as the train.csv file was. I had my best model from the above steps so I used that model to get the predicted values of adview of the test.csv data..!

❖ **Predictions_Submission.csv file link:**

Following Google drive link will redirect to **Predictions_Submission files**

https://drive.google.com/drive/folders/1kzXMqGBUWc7xQLPWs-kQLFSyHB7NUz6H?usp=sharing

Thank you!