……………………………………………...……………………………………………...……………………………………………...……………………………………………………………...………………

**Name:** Akanksha Nannure

**Email address:** akankshaappliedai@gmail.com

**Contact number:** 8790233087

**Anydesk address:**

**Years of Work Experience:** 2

**Date: 30-10-2022**

……………………………………………...……………………………………………...……………………………………………...……………………………………………………………...………………

**Self Case Study -1:** DON’T OVERFIT II

“After you have completed the document, please submit it in the classroom in the pdf format.”

Please check this video before you get started: <https://www.youtube.com/watch?time_continue=1&v=LBGU1_JO3kg>

# **Overview**

\*\*\* Write an overview of the case study that you are working on. ***(MINIMUM 200 words)*** \*\*\*

1. We have 20,000 rows of continuous variables out of which 250 rows are used for training and remaining 19750 rows are used for testing. We need to build best model which doesn’t overfit on train data. Overfitting happens when model memorizes the data patterns in the training dataset but fails to generalize to unseen examples. The dataset itself is small out of which training sample is very little . So building a model without overfitting is really challenging.

# 

# **Research-Papers/Solutions/Architectures/Kernels**

\*\*\* Mention the urls of existing research-papers/solutions/kernels on your problem statement and in your own words write a detailed summary for each one of them. If needed you can include images or explain with your own diagrams. it is mandatory to write a brief description about that paper. Without understanding of the resource please don’t mention it\*\*\*

1. <https://www.kaggle.com/competitions/dont-overfit-ii/data>

This is the kaggle competition that we are working on which has problem statement and data . I have gone through the discussions where they were mostly discussing about scoring ,leaderboard . The code section has some submissions where people followed different approaches to get best model. The common pattern was EDA was done on dataset and balancing dataset, adding new features to data and trying multiple models for classification to get the best model.

1. <https://medium.com/analytics-vidhya/kaggle-competition-dont-overfit-ii-74cf2d9deed5>

This blog describes the solution for the don’t overfit kaggle problem. In the blog he described the hyper parameter tuning and found that Logistic regression, SGD are working better for this data .Standardization of data is done. He has also tried building a basic deep learning MLP model and tried different architectures which gave 80% kaggle score. He also tried using most important features, different permutations of features for modeling . Also tried using combination of different models and done some tuning to get best model. Logistic regression + SVC gave best results for this data. He also performed feature engineering, oversampling technique(SMOTE),dimensionality reduction,calibaration.

# 

# **First Cut Approach**

\*\*\* Explain in steps about how you want to approach this problem and the initial experiments that you want to do. ***(MINIMUM 200 words)*** \*\*\*

\*\*\* When you are doing the basic EDA and building the First Cut Approach you should not refer any blogs or papers \*\*\*

1. The first step would be to do EDA to understand more about data.
2. Need to check if dataset is imbalanced if so need to try different techniques to balance data.
3. Try standardization, normalization techniques and check if they are improving metrics (accuracy , auc roc score etc).
4. Build a basic model and check metric values.
5. Need to try with important features, permutation of features,feature selection techniques and check if its improving model.
6. Try adding new features(Feature engineering) and check if its improving model.
7. Need to try different models and use gridsearch cv or randomsearchcv to find best parameters and get the best model.
8. Compare different models.
9. Try to plot model comparisons ,results to check if model is overfitting.
10. For every experiment getting statistics or plot can help us understand steps more better. Calibrate the model which adds more to the model.

**Notes when you build your final notebook**:

1. You should not train any model either it can be a ML model or DL model or Countvectorizer or even simple StandardScalar
2. You should not read train data files
3. The function1 takes only one argument “X” (a single data points i.e 1\*d feature) and the inside the function you will preprocess data point similar to the process you did while you featurize your train data
   1. Ex: consider you are doing taxi demand prediction case study (problem definition: given a time and location predict the number of pickups that can happen)
   2. so in your final notebook, you need to pass only those two values
   3. def final(X):

preprocess data i.e data cleaning, filling missing values etc

compute features based on this X

use pre trained model

return predicted outputs

final([time, location])

* 1. in the instructions, we have mentioned two functions one with original values and one without it
  2. final([time, location]) # in this function you need to return the predictions, no need to compute the metric
  3. final(set of [time, location] values, corresponding Y values) # when you pass the Y values, we can compute the error metric(Y, y\_predict)

1. After you have preprocessed the data point you will featurize it, with the help of trained vectorizers or methods you have followed for your train data
2. Assume this function is like you are productionizing the best model you have built, you need to measure the time for predicting and report the time. Make sure you keep the time as low as possible
3. Check this live session: <https://www.appliedaicourse.com/lecture/11/applied-machine-learning-online-course/4148/hands-on-live-session-deploy-an-ml-model-using-apis-on-aws/5/module-5-feature-engineering-productionization-and-deployment-of-ml-models>