Machine Learning Model Hyper-Parameter Tuning Log Book:

PS Link :
https://peerabduljabbar.notion.site/DS-2-1d08a908222680d389d0d4a520048a86#1d08a908222680419360cb06c289558c

Authors:

- Marvin Remiigius J
- Aman Vishwakarma

Current Date: 12/04/2025

Day 1:

Understanding the problem statement we decided that, we need a brief understanding of machine learning models, because only after that we can understand the tuning part of it

So first few hours were well spent on understanding the dataset, exploring it, knowing what kind of data were being used

We downloaded the data (Insurance fraud.csv) from kaggle, and we cleaned it up ourselves

At the end of day 1 we created a rough notebook file, showcasing how we are gonna approach the problem statement

End of day one, we sat on a meeting, deciding the changes and improvements that needed to be done in this project

Improvement needed:

1. Break into sections (good notebook)
2. EDA plots, data cleaning explanations (just include necessary one and include some graphs)

Include ROC AUC curve for model comaprison
Button

If time allows,

- Important parameters for each 5 models.
- try to understand one or two more models

To fix 1 error: Press Tab

Day 2:

On this day we were full fledged focused on creating the project file, using front end streamlit

The part we really struggled was to find the best hyper tuning parameters for the five different Models:

Below shows are the versions of the parameters we went through before finally deciding on the one that worked the best

Version 1:

```
param_grids = {
    "Logistic Regression": {'C': [0.001, 0.01, 0.1, 1, 10, 100], 'solver': ['liblinear', 'lbfgs']},
    "Decision Tree": {'max_depth': [3, 5, 10, 20, None], 'min_samples_split': [2, 5, 10]},
    "Random Forest": {'n_estimators': [50, 100, 200], 'max_depth': [5, 10, 20, None], 'min_samples_split': [2, 5]},
    "Gradient Boosting": {'n_estimators': [50, 100, 200], 'learning_rate': [0.01, 0.05, 0.1, 0.2], 'max_depth': [3, 5, 10]},
    "SVM": {'C': [0.01, 0.1, 1, 10], 'kernel': ['linear', 'rbf', 'poly'], 'gamma': ['scale', 'auto']}
}
```

Version 2:

```
param_grids = {
    "Logistic Regression": {'C': [0.001, 0.01, 0.1, 1, 10, 100], 'solver': ['liblinear']},
    "Decision Tree": {'max_depth': [3, 5,], 'min_samples_split': [2, 5]},
    "Random Forest": {'n_estimators': [50, 100, 200], 'max_depth': [5, 10, 20, None], 'min_samples_split': [2, 5]},
    "Gradient Boosting": {'n_estimators': [50, 100, 200], 'learning_rate': [0.01, 0.05, 0.1, 0.2], 'max_depth': [3, 5, 10]},
    "SVM": {'C': [0.01, 0.1, 1, 10], 'kernel': ['linear', 'rbf', 'poly'], 'gamma': ['scale', 'auto']}
}
```

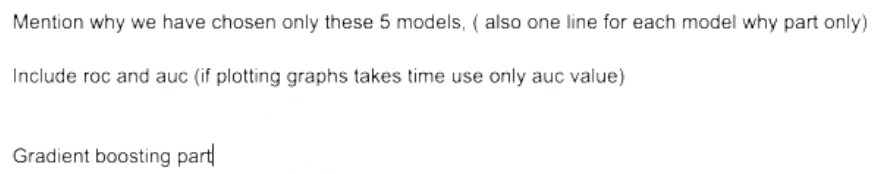
Version 3:

```
param_grids = {
    "Logistic Regression": {
        'C': [0.001, 0.01, 0.1, 1, 10, 100],
        'solver': ['liblinear', 'lbfgs']
    },
    "Decision Tree": {
        'max_depth': [3, 5, 10, 20, None],
        'min_samples_split': [2, 5, 10]
    },
    "Random Forest": {
        'n_estimators': [100, 300, 500],
        'max_depth': [10, 20, 30, 50, None],
        'min_samples_split': [2, 5, 10],
        'min_samples_leaf': [1, 2, 4],
        'max_features': ['sqrt', 'log2', None],
        'bootstrap': [True, False]
    },

    "Gradient Boosting": {
        'n_estimators': [100, 200, 300, 500],
        'learning_rate': [0.01, 0.05, 0.1, 0.2],
        'max_depth': [3, 5, 7, 10],
        'subsample': [0.6, 0.8, 1.0],
        'min_samples_split': [2, 5, 10],
        'min_samples_leaf': [1, 2, 4],
        'max_features': ['sqrt', 'log2']
    },

    "SVM": {
        'C': [0.01, 0.1, 1, 10],
        'kernel': ['linear', 'rbf', 'poly'],
        'gamma': ['scale', 'auto']
    }
}
```

Version 4(Final Version):

```
param_grids = {
    "Logistic Regression": {
        'C': [0.01, 0.1, 1],
        'solver': ['liblinear']
    },
    "Decision Tree": {
        'max_depth': [3, 5, 10, None],
        'min_samples_split': [2, 5,]
    },
    "Random Forest": {
        'n_estimators': [50, 100],
        'max_depth': [5, 10],
        'min_samples_split': [2, 5]
    },
    "Gradient Boosting": {
        'n_estimators': [100, 200],
        'learning_rate': [0.01, 0.05, 0.1],
        'max_depth': [3, 5, 7],
        'subsample': [0.8, 1.0],
        'min_samples_split': [2, 10]
    },
    "SVM": {
        'C': [0.1, 1],
        'kernel': ['linear', 'rbf'],
        'gamma': ['scale']
    }
}
```

Around the middle, of day 2 we decided to sit for another meet discussing further improvements that needs to be done in the project:



We added our personal touch of ROC AUC Curve which shows how well our models are performing, allowing us to know if the model is performing accurately or its bluffing

After doing lot of tweaking and coding, we finally pushed the project files to my github repo, coming to the end of this 48hrs hackathon

Drive Link we used to share all the needed files:
https://drive.google.com/drive/folders/1WhCvy9XBVNcysvnpq-Lcxvp6kM6HLRnc?usp=sharing