PREDICTIVE OPTIMIZATION ANALYTICS CAT 1

Vehicle Insurance Fraud Detection

Vehicle insurance fraud involves conspiring to file false or exaggerated claims related to property damage or personal injuries following an accident. Common examples include staged accidents, where fraudsters deliberately "orchestrate" collisions; the use of phantom passengers, where individuals do not present at the accident scene falsely claim to have suffered severe injuries; and exaggerated personal injury claims, where minor injuries are portrayed as serious to maximize compensation.

You have been assigned the task of developing a predictive model optimized to identify potential fraudulent cases. [Click here to access the dataset you will be using.]

NB. **Use F1_score to evaluate your performance across all models or Alternative Use Smote to sort the class imbalance then use Accuracy, which ever choice of your metrics ensure consistency across the work.**

Follow the steps below

Step 1: Data Preprocessing (7 Marks) (Choice which steps are more applicable to you)

Prepare your dataset for modeling by performing the following tasks. For each step you **decide to use**, provide comments and explanations for your choices and outputs:

a. Data Cleaning

- Handle missing values, duplicates, or outliers.
- Remove or impute missing data and identify inconsistencies in your dataset.

b. Feature Engineering and Extraction

- Derive new features from existing ones to enhance predictive power.
- Extract meaningful information, such as interaction terms or domain-specific indicators.

c. Feature Selection

- Select the most relevant features using techniques such as correlation analysis, Recursive Feature Elimination (RFE), or tree-based feature importance.
- Reduce dimensionality to improve model performance.

d. Feature Encoding

- Convert categorical variables into numerical representations using one-hot encoding, label encoding, or ordinal encoding, depending on the variable type.

e. Feature Scaling

- Standardize or normalize features to bring them to a similar scale, especially for models sensitive to feature magnitude (e.g., ANN).

Step 2: Train a Decision Tree Classifier(3Mrk)

- Train a Decision Tree Classifier on the preprocessed data.
- Evaluate its performance using metrics like the F1 score or accuracy (if using SMOTE).
- Check for overfitting by comparing training and validation performance.

Step 3: Optimize Decision Tree Hyperparameters(4Mrk)

- Use a validation curve to identify the optimal values for **max_depth** and **min_samples_leaf.**
- Retrain the Decision Tree using the optimal parameters.
- Compare the updated model's performance to the initial model.
- Comment on whether the optimization improved performance

Step 4: Train Ensemble Models(7Mrk)

- Use at least two ensemble techniques we have learnt in class
- Evaluate the performance of each ensemble model and compare it to the Decision Tree model.
- Check for overfitting by analyzing performance on training and validation/test datasets.
- Provide an explanation for the observed performance.

Step 5: Train an Artificial Neural Network (ANN)(5Mrk)

- Develop an ANN model with an architecture of your choice.
- Experiment with the number of layers, neurons, activation functions, and optimizers to improve performance.
- Retrain the ANN model and compare its results with previous model

Step 6: Tune the Best-Performing Model (3Mrk)

- Select the best-performing model from the previous steps.
- Use **RandomizedSearchCV** to fine-tune its hyperparameters.
- Evaluate whether the tuning improves the model's performance and provide comments.
- State you best model and its score either – F1 Score or Accuracy (If used SMOTE)

Step 7: Save Your Best Model (1 Mrk)

- Save the best-performing model to your local machine for future use.
- This model will be deployed later during the deployment class.