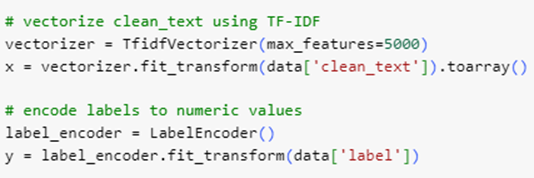
## Main Task 1: AI Detection Development

1. Feature Engineering

Select relevant features and create new ones if required.

Brief explanation:



This code snippet performs two crucial machine learning preprocessing steps:

TfidfVectorizer: Converts cleaned text data into numerical vectors using TF-IDF (term frequency-inverse document frequency). This technique prioritises document terms that are rare in the dataset but common in the document. Max\_features=5000 restricts vocabulary to the top 5,000 terms for computational efficiency.

Label Encoding: The LabelEncoder maps each unique label to an integer, converting categorical labels into numeric values. Machine learning models, which process target variables with numeric inputs, depend on this encoding.

By machine-reading raw text and category labels, these steps prepare the dataset for training.

1. Model Building

Train multiple machine learning models like RF, SVM, Logistic Regression, etc.



This code is used for implementing and training four different machine learning models: Random Forest, SVM, Logistic Regression, and Artificial Neural Network (ANN). Here's how and where these models can be used:

1. Random Forest

Its best for classification tasks with structured/tabular data. Use cases like fraud detection, medical diagnosis, and customer churn prediction since it handles missing values and outliers well and Reduces overfitting through ensemble learning.

RandomForestClassifiers are trained on X\_train and y\_train using 100 decision trees (n\_estimators=100).

Random Forest, an ensemble learning method, aggregates decision tree outputs to improve performance and reduce overfitting.

1. SVM Model:

The SVM model, which is trained using X\_train and y\_train, uses a linear kernel to classify (kernel='linear').

To evaluate forecast confidence, enable probability outputs (probability=True).

SVM finds the optimum hyperplane to classify data points accurately.

1. logistic regression

A logistic regression model with max\_iter=1000 is initialised and trained using training data to ensure convergence.

For binary or multi-class classification, logistic regression estimates categorical outcome likelihood.

1. ANN

Create an ANN model using Keras' Sequential API. Input layer neurones are triggered using ReLU (activation='relu') and adjusted for input feature size (input\_dim=X\_train.shape[1]).

Two hidden layers with dropout regularisation (Dropout(0.5)) to reduce overfitting: 64 neurones (activation='relu') and 32 neurones.

A multi-class output layer uses neurones equal to class count (len(label\_encoder.classes\_)) and softmax activation (activation='softmax') for probability distributions.

Using the Adam optimiser and sparse categorical cross-entropy as the loss function, a model with integer-encoded multi-class labels is developed.

Using training and validation data spanning 10 epochs, the model is trained with a batch size of 32.

Machine learning techniques such as tree-based ensemble approaches (Random Forest), linear separation (SVM), statistical modelling (Logistic Regression), and deep learning are implemented in this code. To compare performance, each technique is trained on the same dataset.

## Main Task 2: Web Application Development

1. Form Development

Create input forms for user data submission.

## Main Task 3: AWS Deployment and Security

1. AWS VPC Configuration

Create and configure a Virtual Private Cloud (VPC) for network isolation.

1. Dashboard Integration

Integrate a real-time dashboard for threat visualization.

## Main Task 4: Documentation and Reporting

1. Technical Documentation

Create documentation for AI model development and AWS integration.

1. Presentation Preparation

Create a final presentation to showcase the project outcomes (all members).

1. Report Writing

Draft and finalize the project report.