Fraud_Email_MNB.ipynb — Detailed Overview

This notebook implements a machine learning pipeline using the Multinomial Naive Bayes (MNB) algorithm to detect fraudulent emails.

1. Libraries and Data Loading

Imports:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import string, re
```

Loads the email dataset:

```
df = pd.read_csv('fraud_email_.csv')
```

2. Text Cleaning

A cleaning function is defined:

```
def clean_text(text):
    text = text.lower()
    text = re.sub(r'https?://\S+|www\.\S+', '', text)
    text = re.sub(r'<.*?>+', '', text)
```

```
    text = re.sub(r'[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub(r'\n', '', text)
    text = re.sub(r'\w*\d\w*', '', text)
    return text
```

Applied as:

```
df['Email Text'] = df['Email Text'].apply(clean_text)
```

3. Visualization

Email label distribution plotted using seaborn:

```
sns.countplot(df['label'])
```

4. Feature Extraction

Using CountVectorizer to convert text into numeric features:

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer()
X = cv.fit_transform(df['Email Text'])
```

5. Model Training

Splits the data:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, df['label'], test_size=0.2, random_state=42)
```

Trains the model:

```
from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
model.fit(X_train, y_train)
```

6. Evaluation

Evaluates using classification report and confusion matrix:

```
from sklearn.metrics import classification_report, confusion_matrix
y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

7. Prediction Function

Defines a prediction function:

```
def predict_email(text):
    text = clean_text(text)
    text_vectorized = cv.transform([text])
    prediction = model.predict(text_vectorized)
    return 'Scam' if prediction[0] == 1 else 'Not Scam'
```

Used as:

```
predict_email("Congratulations! You've won a $1000 gift card...")
```

Summary:

- Basic ML + NLP pipeline using CountVectorizer + MultinomialNB.

- Cleaned email content.

- Trained classifier and evaluated with standard metrics.

- Deployed a helper function to classify new email texts.

8. Module Descriptions

- pandas: Used for loading and manipulating structured tabular data.

- numpy: Provides numerical operations, though minimally used here.

- matplotlib.pyplot & seaborn: Used for plotting and visualizing data distributions.

- re & string: Used for text preprocessing and regular expressions.

- sklearn.feature_extraction.text.CountVectorizer: Converts text data into a bag-of-words feature matrix.

- sklearn.model_selection.train_test_split: Splits the dataset into training and testing subsets.

- sklearn.naive_bayes.MultinomialNB: The classifier used for modeling text data with multinomial likelihoods.

- sklearn.metrics: Used for performance evaluation including classification report and confusion matrix.

The script combines classical machine learning (Multinomial Naive Bayes) with NLP preprocessing to detect scam emails based on token frequencies.