Fraud prevention case Snappt:

1. Preliminary Dataset Analysis:
After briefly exploring the fraud dataset, these are some of the key findings. First, the label (clean vs edited) has a slight imbalance, with only 20% of the dataset appearing to be identified as fraud. Usually this makes it more difficult to train a performing model, indicating a quite fraudulent population. There are very few unlabeled rows (4). This will be dropped for model training. Most features show a continuous distribution with the exception of color, which shows only integer values. This makes sense given the document was probably digitized. Edge noise and alpha channel density show a bimodal distribution that when splitting them by label shows each mode corresponding to a class (edited/clean).
Similarly with income but not as pronounced. There appears to be no duplicate data, some IDs are missing and some numerical columns have nan. Also, while looking at scatter plots (edge_noise vs alpha_channel_density) data appears linearly separable, suggesting a simple model could be enough (e.g. Logistic Regression, small neural net). Also, grayscale and text density are highly correlated so one of the features will be ignored for now.

2. Baseline model:
As mentioned in the data analysis, datasets look almost linearly separable. So, the first model as a baseline will be a Logistic Regression and testing using 20% of data as the dataset is quite small. A simple model using the mentioned features (noise + density), almost perfect metrics are achieved. ROC AUC=1, Precision=0.975 Recall=1 on Train set. ROC AUC=1, Precision=1 Recall=1 on Train set. Only misclassifying 2 non-fraudulent applications.

3. Marginal improvements on the model:
To try to improve the already performant model 2 experiments were run: One neural network with 4 layers (with GPU) and then XGBoost with hyperparameter tuning and 5 fold KV. In this case both models achieved a perfect score on train and test set in terms of recall and precision. XGBoost model is chosen for simplicity and speed, and it does not require GPU for similar performance. The final model is XGBOOSTClassifier(max_depth=7, n_estimators=1000). Looking at decision boundaries, the EDITED set looks much better isolated than linear regression, taking into consideration that the group has less variance than the clean group. Given small datasets, other models such as SVM could be used, but it would not scale if the dataset continued to grow. Making it slower for training and inference.

4. Deployment:
Simple API endpoint that can receive all features (in case needed in the future). Host on a small cloud instance with autoscaling if traffic spikes. No GPU. For logging, consider latency, and recall metrics, specially if fraud is much more expansive than something like churn (precision). Also, monitor main statistics on input features and distribution of labels on a periodic basis to check for datadrift (shifting, scaling, etc). Applicants could get more sophisticated making detection harder. Also, the amount of fraud/total could get much smaller, making the dataset much more unbalanced. Deploy cyclical retrains
and triggered retrains based on significant datadrift detection with statistical tests. Feedback from stakeholders is critical here. Also monitor for overfitting as the dataset grows. If another model is already in place, the deployment of the new one should be gradual,
maybe using A/B testing groups.
Finally, some fields, such as income are highly sensitive, so sanitizing is super
important.