**“How Ensemble Learning Boosted My Spam Classifier’s Accuracy to 98%”**

**How Ensemble Learning Boosted My Spam Classifier’s Accuracy to 98%**

*CarolNyanjui*  
*3 min read · Just now*

**The Hidden War Against Spam: How Machine Learning Detects Scams**

Every day, millions of messages attempt to scam unsuspecting users—offering fake lottery winnings, posing as banks, or pushing fraudulent links. While many are easy to ignore, behind the scenes, **machine learning** plays a crucial role in detecting and stopping them.

As part of a recent data science project, I explored how ensemble learning—a technique that combines multiple models—can **supercharge SMS spam detection**. The result? A smart filter with **98.2% accuracy** that consistently outperforms individual models.

**🗂️ Understanding the Dataset**

To simulate a real-world SMS filtering task, I analyzed a dataset containing thousands of SMS messages labeled as:

* ham (legitimate)
* spam (fraudulent)

Key insights:

* **87%** of messages were ham, **13%** were spam.
* Spam messages tended to be longer and packed with emotional triggers like **“urgent,” “free,”** or **“win.”**

**🧹 Preprocessing & Feature Engineering**

Cleaning the raw text was vital for meaningful model training:

* Removed punctuation, digits, stopwords
* Lemmatized tokens (e.g., “won” from “winning”)
* Extracted features using **TF-IDF**
* Engineered new features:
  + Message\_Length
  + Digit\_Count
  + Word\_Count

These steps allowed models to focus on the true semantics of spam messages.

**🤖 Model Training**

I trained three classifiers:

* **🔹 Logistic Regression** – simple and fast, great for linearly separable data.
* **🔹 Multinomial Naive Bayes** – lightweight and ideal for text classification.
* **🔹 Random Forest** – powerful and robust, great for capturing nonlinearities.

Each model performed well individually—but I knew we could do better.

**🧠 Enter: Ensemble Learning**

Instead of choosing one "best" model, I combined all three using a **Voting Classifier** with soft voting (based on predicted probabilities). This allows the model to **leverage the unique strengths of each algorithm** while minimizing weaknesses.

VotingClassifier(estimators=[

('lr', LogisticRegression()),

('nb', MultinomialNB()),

('rf', RandomForestClassifier())

], voting='soft')

**📊 Results Summary**

| **Model** | **Accuracy** | **AUC** |
| --- | --- | --- |
| Logistic Regression | ~97.5% | 0.99 |
| Naive Bayes | ~96.8% | 0.98 |
| Random Forest | ~96.9% | 0.97 |
| ✅ **Ensemble** | **98.2%** | **0.995** |

**ROC Curves** further confirmed the ensemble model’s superiority—it consistently achieved the highest AUC, signaling better overall performance.

**🎯 Conclusion**

This project proved that **ensemble learning is a game changer** for spam detection. By fusing diverse models, we created a classifier that is more resilient, balanced, and accurate than any single model.

**🔮 What’s Next for Spam Detection?**

While ensemble models work impressively well, the spam war is evolving. Future directions may include:

* Integrating **deep learning** models (e.g., LSTM, BERT)
* Real-time filtering in production systems
* Explainability tools like SHAP for model transparency

So the next time your inbox filters out a suspicious SMS, remember: it’s likely thanks to machine learning working quietly behind the scenes—just like this ensemble.

*Stay curious, stay skeptical, and let AI do the heavy lifting in the hidden war against spam.*  
🧠💌

Would you like this formatted as a downloadable .pdf or .md file for publishing?