Naive Bayes	Test Accuracy	Test F1 Score
Using only Number_Word_Tokens	0.8810483870967742	0.9163120567375886
Using only Number_Word_Types	0.8810483870967742	0.9160739687055477
Using first three features	0.9133064516129032	0.9408528198074277
Using all features	0.9112903225806451	0.9388888888888889
Logistic Regression	Test Accuracy	Test F1 Score
Using only Number_Word_Tokens	0.9294354838709677	0.9532710280373832
Using only Number_Word_Types	0.9173387096774194	0.945260347129506
Using first three features	0.9415322580645161	0.9614873837981408
Using all features	0.9536290322580645	0.9692101740294511

Summary:

For Naïve Bayes, using either Number_Word_Tokens or Number_Word_Types yields very similar performance with about 88% accuracy and 91–92% F1 scores. Looking at the use of the "first three features", it seems that adding more improves both accuracy (91.3%) and F1 score (94.1%), so the combination of vocabulary metrics and spelling execution seems to enhance the classification. Surprisingly however, adding the fourth feature (Average_Word_Length) doesn't lead to a major improvement. In fact, both accuracy (91.1%) and F1 score (93.9%) slightly decrease, suggesting that for Naive Bayes, this extra might not provide useful discriminative information, or it could introduce noise.

With Logistic Regression, it is interesting that using only Number_Word_Tokens leads to a much higher accuracy (92.9%) and F1 score (95.3%) compared to using only Number_Word_Types (91.7% accuracy and 94.5% F1). This suggests that the Logistic Regression model benefits more from the token count feature, which might capture more distinctive patterns in the data. Similarly, using the first three features gives a noticeable boost in accuracy (94.2%) and F1 score (96.1%), therefore, the combined feature set offers improvement over single features alone. In contrast to Naïve Bayes, adding all features improves Logistic Regression's performance to the highest accuracy (95.4%) and F1 score (96.9%), suggesting that Logistic Regression is able to leverage the full set of features effectively without overfitting or noise interference.

It seems that Naive Bayes relied more heavily on conditional probabilities based on feature independence. While this worked well with fewer or independent features, adding features that don't contribute as much might decrease performance, as seen with the all-features scenario. On the other hand, Logistic Regression appears to benefit from a larger feature set, which aligns with its strength in handling multicollinearity and capturing more complex decision boundaries. This is evident from the consistent performance improvement as more features are added.

Overall, Logistic Regression outperforms Naive Bayes across all feature subsets, both in accuracy and F1 scores. This suggests that Logistic Regression is better suited for this classification task, likely because it can model more complex relationships between features and the target label, whereas Naive Bayes is limited by its assumption of feature independence.