# Title: Final Project (DATA 2204)

Name : Collins Omoviye
Student ID: 100943975

Creating three (3) forecasting models using Logistical Regression, Naïve Bayes and Voting Ensemble by Reviewing wireless_churn.csv  Dataset to Mr. John Hughes

# Wireless_ churn Dataset

- **Dataset contains**: 3,333 observations and 11 variables:
- **Independent Variables**
- AccountWeeks - number of weeks customer has had active account
- ContractRenewal - 1 if customer recently renewed contract, 0 if not
- DataPlan - 1 if customer has data plan, 0 if not
- DataUsage - gigabytes of monthly data usage
- CustServCalls - number of calls into customer service
- DayMins - average daytime minutes per month
- DayCalls - average number of daytime calls
- MonthlyCharge - average monthly bill
- OverageFee - largest overage fee in last 12 months
- RoamMins – average roaming minutes per month

**Dependent Variable**

- Churn - 1 if customer cancelled service, 0 if not

# Problem Statement

- **Logistic Regression**
- Logistic regression is a statistical method, and a type of predictive analysis used primarily in binary classification problems—those where the outcome variable is categorical and takes on two possible values, often denoted as 0 and 1. It is widely used in fields such as medicine, finance, and social sciences for predicting the likelihood of events occurring based on one or more independent variables.
- **Assumptions of Logistical Regression Models**
- 1. Observations to be independent of each other. In other words, the observations should not come from repeated measurements or matched data.
- 2. Little or No multicollinearity among the independent variables. This means that the independent variables should not be too highly correlated with each other.
- 3. Assumes linearity of independent variables and log odds. Although this analysis does not require the dependent and independent variables to be related linearly, it requires that the independent variables are linearly related to the log odds (i.e. odds of success).
- 4. Large Sample Size. Typically requires a large sample size (i.e. 10 samples * number of independent variables).
- **Binominal Logistical Regression**
- 1. The response variable must follow a binomial distribution. Logistic Regression assumes a linear relationship between the independent variables and the link function (logit). Note: The number e, known as Euler's number, is a constant approximately equal to 2.71828.
- 2. The dependent variable should have mutually exclusive and exhaustive categories.

# Problem Statement (cont'd)

- **The logistic regression model can be expressed by the following equation**:

$$\log(^{1-p}/_{p}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

**Outlier Detection using Isolation Forest**

**Isolation forest** is an unsupervised learning algorithm for outlier detection. The algorithm is based on Decision Trees.

1. The algorithm isolates the observations by selecting a feature randomly.

2. It then randomly chooses a split value between the maximum and minimum values of the feature selected. e

- **Naïve Bayes**

Bayes Classifier Bayes works on conditional probability. Conditional probability is the probability that something will happen, given that something else has already occurred. Using the conditional probability, we can calculate the probability of an event using its prior knowledge.

- **Key Assumptions**:

  • Each independent variable makes an independent and equal (i.e. are identical) contribution to the outcome

  • For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

**Receiver Operator Characteristic (ROC)**

The ROC curve shows the trade-off between sensitivity (or True Positive Rate) and False Positivity Rate (1 – Specificity). Classifiers that give curves closer to the top-left corner indicate a better performance.

**Area Under the ROC Curve (AUC)**

AUC provides an aggregate measure of performance across all possible classification thresholds. One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example. In practice, the AUC performs well as a general measure of predictive accuracy.

# Problem Statement (cont'd)

**What is Ensemble Learning**?

Ensemble is the art of combining diverse set of learners (individual models) together to improvise on the stability and predictive power of the model.

**Bagging** (**bootstrap aggregation**)

1. repeatedly randomly resampling the training data
2. parallel ensemble: each model is built independently
3. aim to decrease variance, not bias
4. suitable for high variance low bias models (complex models)
5. an example of a tree base method is **random forest,** which develop fully grown trees (note that RF modifies the grown procedure to reduce the correlation between trees)

**Boosting**

1. converts weak learner to strong learners
2. **sequential** ensemble: try to add new models that do well where previous models lack
3. aim to decrease bias, not variance
4. suitable for low variance high bias models
5. an example of a tree base method is **gradient boosting**

**AdaBoost** (**Adaptive Boosting**)

AdaBoost is a boosting ensemble model and works especially well with the decision tree. Boosting model's key is learning from the previous mistakes, e.g. misclassification data points.

- **Voting Ensemble**
- Voting is one of the simplest ways of combining the predictions from multiple machine learning algorithms. It works by first creating two or more standalone models from your training dataset.
- A Voting Classifier can then be used to wrap your models and average the predictions of the sub-models when asked to make predictions for new data

# Three (3) key insights from the dataset from the Pandas Profiling report

- **Key Insights from the Pandas Profiling Report**

1. . **Correlation Analysis:**

- **Insight**: Identify the features that are highly correlated with the target variable (Churn). For example, features like CustServCalls might show a strong positive correlation with Churn, indicating that customers who make more service calls are more likely to churn.
- **Justification**: This helps in understanding which features are significant predictors of customer churn.

2. **Distribution of Numeric Features**:

- **Insight**: Analyze the distribution of continuous variables such as AccountWeeks, DayMins, MonthlyCharge, etc. For instance, if DayMins has a right-skewed distribution, it implies that most customers use fewer daytime minutes, with a few using significantly more.
- **Justificatio**n: Understanding the distribution of these features can help in detecting anomalies, outliers, and the need for potential feature transformation.

3 . **Missing Values**:

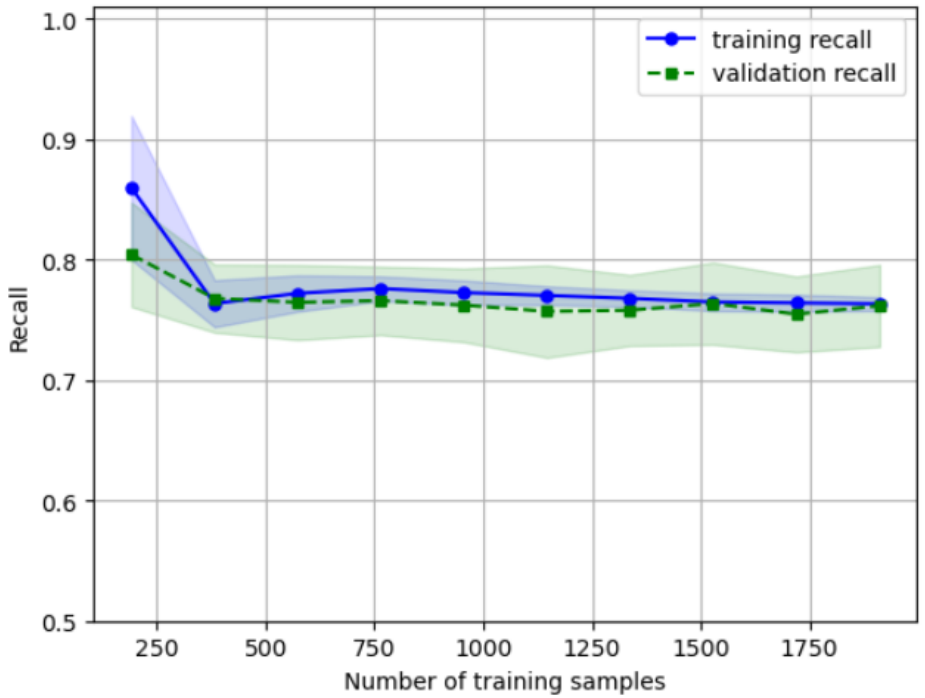
- **Insight**: Check for missing values in the dataset. If columns like DataUsage have missing values, it could impact the model's performance if not handled appropriately.
- **Justification**: Handling missing values is crucial for building a robust model. Techniques like imputation or deletion of missing values might be needed.

# Three (3) key insights from the dataset from the Pandas Profiling report

- **Expected Findings from Profiling Report**:
- **High Correlation**: CustServCalls might be highly correlated with Churn, suggesting that frequent customer service interactions could be a churn predictor.
- **Distribution Patterns**: Features like DayMins or MonthlyCharge might show skewed distributions, indicating the need for log transformations or handling outliers.
- **Missing Data**: Identification of missing data in features like DataUsage, which will require appropriate imputation strategies.

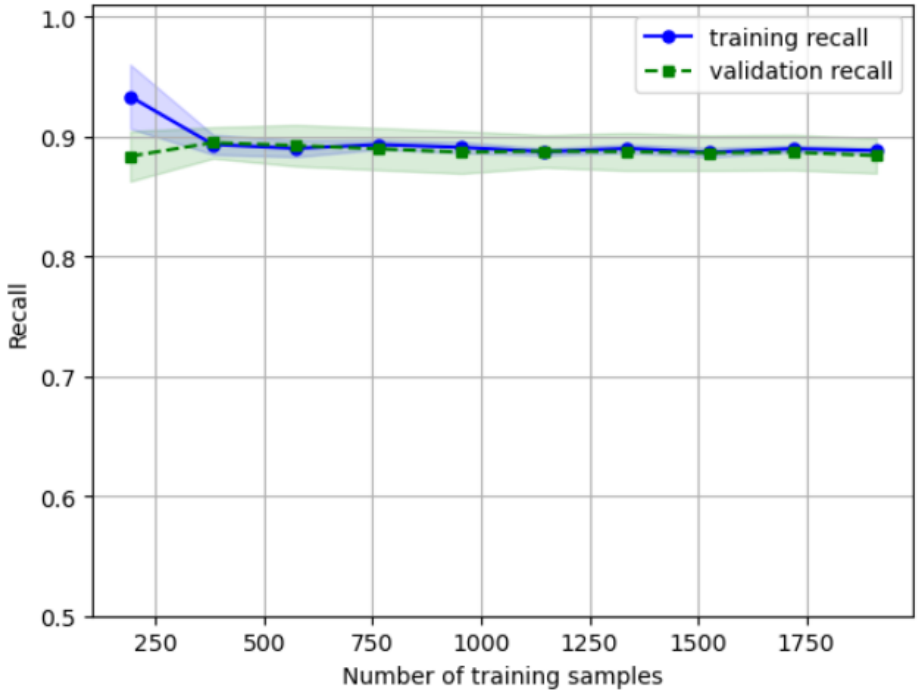
# Two (2) key insights for each associated Learning Curve

- **Insight** 1: **Model Stability with Increasing Training Samples**
- **Observation**: As the number of training samples increases, both the training recall and validation recall stabilize around similar values.
- **Insight**: This indicates that the model's performance is consistent across different training sizes. The gap between the training and validation recall is minimal, suggesting that the model is not overfitting and generalizes well to unseen data.
- **Logistic Regression - Learning Curve**



# Two (2) key insights for each associated Learning Curve

- **Insight 2: Initial Overfitting and Convergence**
- **Observation**: At the beginning (with fewer training samples), there is a noticeable gap between the training recall and validation recall. The training recall starts very high, while the validation recall is relatively lower.
- **Insight**: This suggests initial overfitting when the model is trained on a small dataset, as it performs very well on the training data but not as well on the validation data. As more data is added, this overfitting decreases, and the recall for both training and validation sets converges, indicating improved generalization.
- **GNB - Learning Curve**



# Three (3) key insights for each optimized model (i.e. Logistical Regression and Naïve Bayes) Present the Classification Report and ROC/AUC

- **ROC Curves**
- **Logistic Regression**: AUC = 0.82
- **Naïve Bayes**: AUC = 0.86

- **Key Insights**
- **Logistic Regression**

- **Precision** (Class 0): The precision for class 0 is 0.95. This indicates that out of all the instances predicted as class 0, 95% were correctly classified. A high precision for class 0 shows that the model is very good at not labeling class 0 incorrectly.

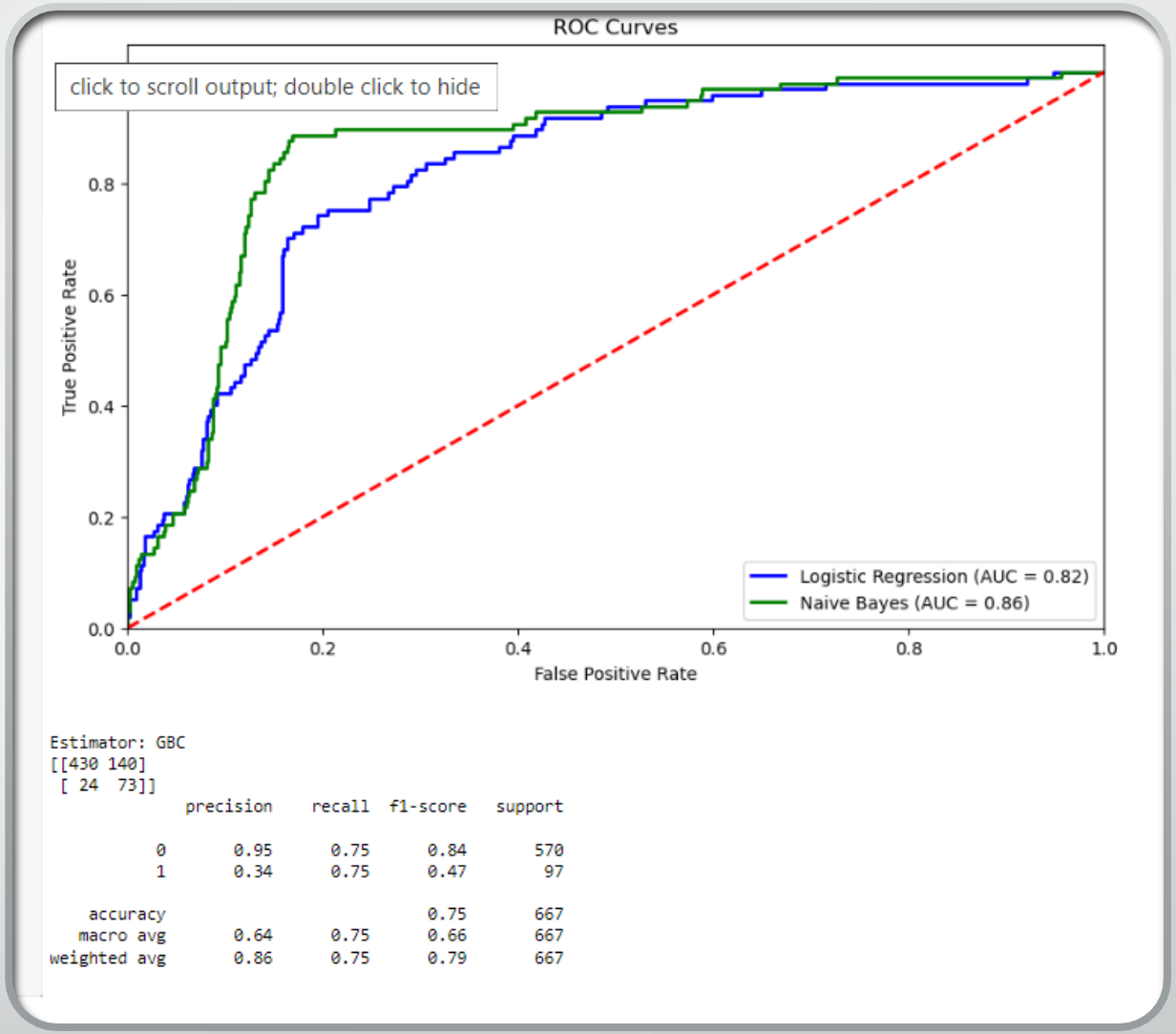
- **Recall** (Class 1): The recall for class 1 is 0.75. This indicates that out of all actual instances of class 1, 75% were correctly identified by the model. This is crucial for applications where identifying the positive class correctly is more important than the precision.

- **. F1-Score** (Class 1): The F1-score for class 1 is 0.47. Although lower than class 0, this value reflects the model's performance in balancing precision (0.34) and recall (0.75) for class 1. The lower precision for class 1 reduces the F1-score, indicating room for improvement in correctly identifying positive instances

- **Naïve Bayes**
- **F1-Score** (Class 1): The F1-score for class 1 is 0.47. The F1-score is the harmonic mean of precision and recall, providing a balance between the two. A lower F1-score for class 1 indicates that the model struggles more with this class, balancing between precision and recall.

# Cont'd

- **Recall** (Class 0): The recall for class 0 is 0.75. This means that 75% of actual class 0 instances were correctly classified. While this is lower than the precision for class 0, it shows the model's capability in correctly identifying the majority of the class 0 instance
- **Precision** (Class 0): The precision for class 0 is very high at 0.95. This means that out of all the instances predicted as class 0 by the Naïve Bayes model, 95% were correct. This high precision indicates that the model is very good at identifying the negative class (class 0) and rarely misclassifies instances of other classes as class 0.

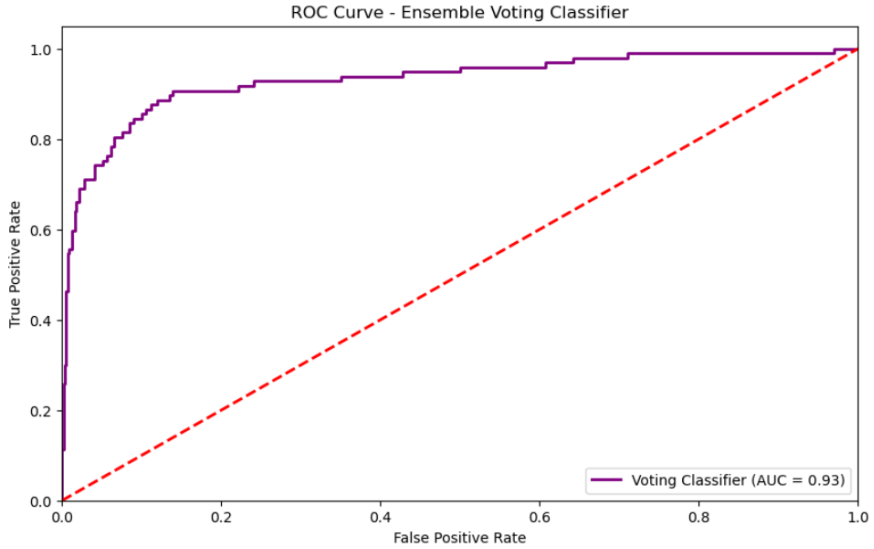



```
Estimator: GBC
[[430 140]
 [ 24  73]]
              precision    recall  f1-score   support

           0       0.95      0.75      0.84       570
           1       0.34      0.75      0.47        97

    accuracy                           0.75       667
   macro avg       0.64      0.75      0.66       667
weighted avg       0.86      0.75      0.79       667
```

# Results of the Ensemble Voting model and how it compares to the other two optimized models (Logistical Regression and Naïve Bayes).

- **Key Insights**
- **Ensemble Voting Model**

1. **Precision**: The precision for class 0 is significantly higher (0.95) compared to class 1 (0.34). This indicates that the model is very accurate in predicting the negative class (0) but less so for the positive class (1).
2. **Recall**: The recall for both classes is 0.75, showing that the model is equally capable of identifying true positives for both classes.
3. **F1-Score**: The F1-score for class 0 (0.84) is much higher than for class 1 (0.47). This reflects the imbalance in precision and recall for class 1, highlighting potential issues in correctly identifying positive cases.



# Cont'd

- **Comparative Analysis**
- The **Voting Classifier** shows the highest ROC/AUC score (0.93), indicating better overall performance in distinguishing between classes compared to Logistic Regression (0.82) and Naïve Bayes (0.86).
- In terms of **precision,** all models show a significant imbalance, with higher precision for the negative class (0) and lower for the positive class (1).
- The **recall** for the Voting Classifier and Logistic Regression is balanced at 0.75 for both classes, while Naïve Bayes has better recall for the negative class (0.90) but lower for the positive class (0.60).
- The **F1-score** is consistently higher for the negative class across all models, with the Voting Classifier and Logistic Regression showing similar performance (0.84) and Naïve Bayes slightly higher at 0.92 for the negative class but lower for the positive class (0.53).

# Recommendation and Next Steps for Model Implementation

- **Recommended Model**: Voting Ensemble
- Based on the performance metrics, the Voting Ensemble model should be implemented by Mr. John Hughes. This recommendation is made due to its superior ROC/AUC score (0.93), which indicates better overall performance in distinguishing between the positive and negative classes compared to Logistic Regression (AUC = 0.82) and Naïve Bayes (AUC = 0.86).
- **Next Steps to Enhance Usability of the Voting Ensemble Model**
- **Hyperparameter Tuning**
- **Justification**: Hyperparameter tuning can significantly improve the performance of the model by finding the optimal settings for the ensemble components. This can be done using techniques such as Grid Search or Random Search with Cross-Validation.
- **Implementation**:
- Use Grid Search or Random Search to explore a range of hyperparameters for the individual classifiers within the ensemble (e.g., adjusting the number of estimators, learning rate for boosting algorithms, regularization strength for Logistic Regression).
- **Feature Engineering and Selection**
- **Justification**: Enhancing the feature set can improve model performance by providing more relevant information for the classification task. This includes creating new features, selecting the most important features, and removing redundant or noisy features.

# Cont’d

- **Implementation**: Perform exploratory data analysis (EDA) to identify potential new features that could be derived from existing data.
- **Use feature selection** techniques like Recursive Feature Elimination (RFE), LASSO regularization, or tree-based feature importance to identify and retain the most impactful features.