Mphasis
The Next Applied

Drift Velocity in Machine Learning

Whitepaper by Dr. Archisman Majumdar, AVP & Lead - Applied AI, Mphasis NEXT Labs | Keerthana Sundaresan, Intern, Mphasis

Contents

1.

Introduction

Large scale Machine Learning (ML) projects involving continuous model build, train, deploy and retrain are growing in prominence as compared to bespoke ML projects. Therefore, automated, repeatable and scalable best practices are needed to manage them. A common assumption made while training ML models is that the data remains stable. However, the data evolves over time, and this effectively changes the patterns and relations on which the models had initially been trained, causing performance degradation.

In this paper, we highlight the need for rigorous methodologies and best practices across the ML lifecycle to monitor, measure and resolve drift in producing ML models. We will also introduce PACE-ML, which is Mphasis MLOps framework & methodology for automated, continuous end-to-end machine learning.

PACE-ML incorporates workflows, collaboration and tools to improve model selection, reproducibility, versioning, auditability, explainability, packaging, reusability, validation, deployment & monitoring. It also helps in dealing with drift.

The umbrella term model drift is used while monitoring models. This includes data drift – the evolution of data that may have introduced unseen variations and concept drift – the change in the interpretation of data over time. For example, during a pandemic, the drop in taxi demands may be attributed to a city-wide lockdown, explaining the change in data. However, the continued low on car-pooling demands post-lockdown that may occur due to social distancing is a hidden element that would affect the data.

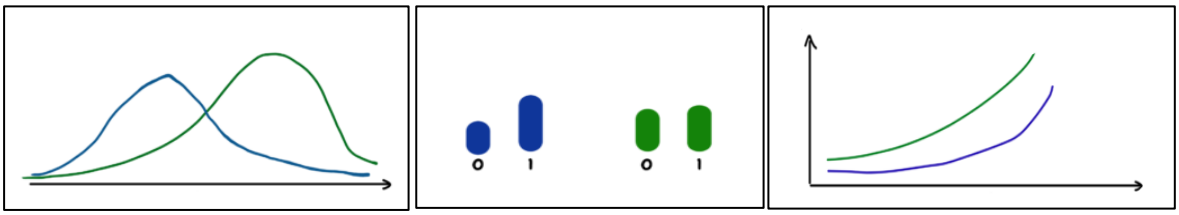

Figure 1. Feature Drift – change in P(X); Label Drift – change in P(Y); Concept Drift – change in P(Y/X)
Source: https://dkopczyk.quantee.co.uk/covariate_shift/

1.1 Concept Drift

Concept drift is defined as the change in relationships between input and output data in the underlying problem over time.

The concepts of interest may depend on some 'hidden context'. For example, temperature data is prone to seasonal changes which may not be clearly mentioned in the data; customers' buying preferences may be influenced by the strength of the economy. The "hidden context" here is not explicitly defined in the data, but has great influence over it, thus altering the interpretation of data.

Formally, concept drift between time point *t* and *u* can be defined as:

$P_t(X,y) \neq P\neg_u(X,y)$

where P_t denotes the joint distribution at time *t* between the set of input variables *X* and target variable *y*. Changes in aspects of this relation are attributed to changes in data. Therefore, prior probabilities of classes P(y) may change and class conditional probabilities P(X|y) may change. As a result, the posterior probabilities of classes P(y|X) may change affecting the model prediction. Such changes may occur with or without incoming data changes, i.e., P(X).

1.2 Understanding Drift

In real-world tasks, drift may occur due to many reasons, such as changes in individual preferences, population shift or the complexity of the environment. It may occur as a combination of multiple changes or as reoccurring concepts. It may also be affected by seasonality or be completely unpredictable by nature.

The data arriving may also have a variety of forms such as sequential, time series, numerical, batches or data streams. The input data can be categorized by task, whether it is used for classification, regression, clustering and so on. The source of the data and its task may also be grouped together in various application areas, such as user modelling or predictive analytics.

2.

Drift Velocity

A clear understanding of the nature of the data, the use case, and the type of drift would be instrumental in decoding the performance of a model in the ML production environment. Obtaining this knowledge in the form of a literature review leads one to conclude that there is a gap in the literature – in terms of arriving at a definition for drift that would encompass the various characteristics of drift.

In ML production models, in addition to detecting the drift, gaining insights on the behavior of drift would prove to be useful in improving the models; for example, knowing the frequency at which drift occurs would enable one to increase or decrease the rate at which drift is checked and then handled. Thus, a drift velocity measure is defined, using certain concept drift characteristics that are categorized into three broad pillars as the components of drift velocity.

3.

Components of Drift Velocity

3.1 Speed of Drift

Drifting time is defined as the number of time steps taken before a new concept replaces an older one. Speed is the inverse of this time; i.e., higher speed is associated with lower number of time steps and vice versa. *Frequency* is an important term related to the speed of drift, and refers to how often drift occurs over a fixed period of time. When new drifts occur within short instances of time, it indicates a high frequency. A low frequency refers to long intervals between drifts.

Based on the speed, drift may be classified as *sudden/abrupt drift*, if concepts switch from one to another within a few steps of each other. It may also be classified as *gradual drift*, if the distributions slowly change over a period of time. A large drifting time is associated with this

type of drift; however, the slow transition period of uncertainty poses a difficulty in terms of detecting the drift. Gradual drift may be further divided into *gradual probabilistic* and *gradual continuous drift*. In the former, two concepts exist together until the probability of sampling from one increases, and slowly replaces the other. In gradual continuous drift, the concept shift occurs through minor modifications at every time step.

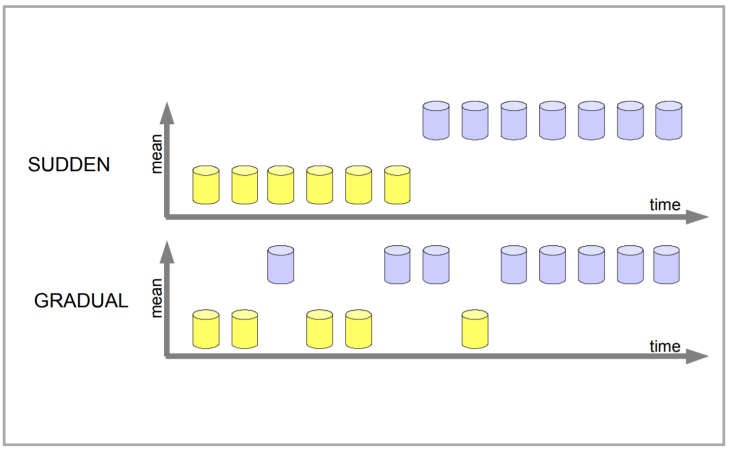

Figure 2. Sudden and Gradual Drift

Source: Bifet, A., Gama, J., Pechenizkiy, M., & Zliobaite, I. (2011). Handling concept drift: Importance, challenges and solutions. PAKDD-2011 Tutorial, Shenzhen, China.

3.2 Severity of Drift

It refers to the extent of changes caused by the drift to a new concept. Based on the severity, drift may be categorized into *local* and *global drift*. Local drift is said to occur if changes only affect some regions of the instance space. In some cases, noise might be confused for local drift, which makes it important to distinguish between the two. Global drift is characterized by changes that affect the entire instance space; the concepts have more noticeable differences.

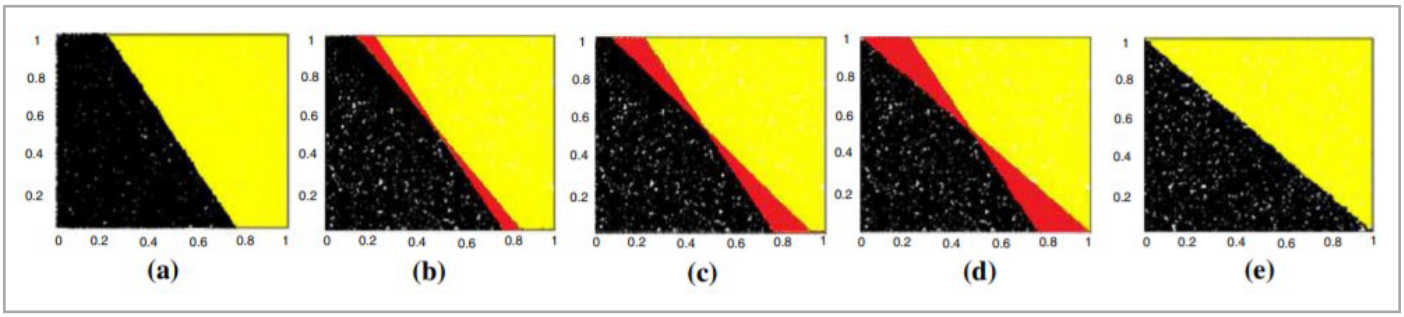

*Figure 3. Gradual continuous local drift: **a** concept1, **b-d** instance space affected by the drift and **e** concept2*

Source: Khamassi, I., Sayed-Mouchaweh, M., Hammami, M., & Ghédira, K. (2018). Discussion and review on evolving data streams and concept drift adapting. Evolving systems, 9(1), 1-23.

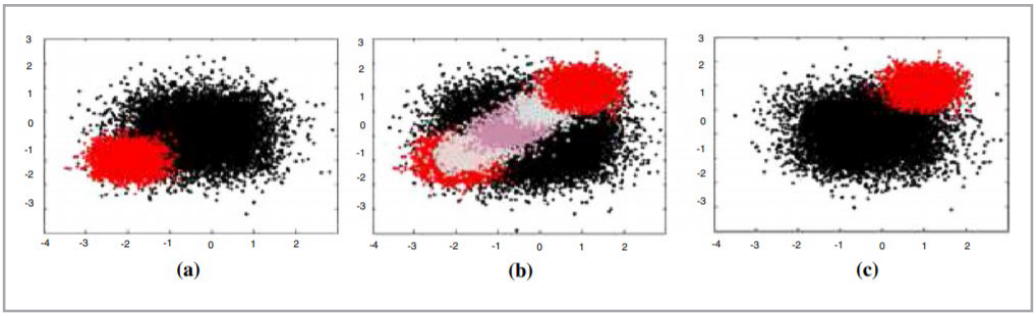

*Figure 4. Gradual continuous global drift: **a** concept1, **b** concept evolution and **c** concept2*

Source: Khamassi, I., Sayed-Mouchaweh, M., Hammami, M., & Ghédira, K. (2018). Discussion and review on evolving data streams and concept drift adapting. Evolving systems, 9(1), 1-23.

3.3 Periodicity

It is the characteristic of drift that occurs due to the reappearance of a pre-existing concept, or a concept similar to an older one. Based on periodicity, drift is divided into *cyclic* and *acyclic, or unordered, drift*. Drift that occurs according to a certain periodicity or due to a seasonable trend is cyclic, while the uncertainty of when a concept might reappear is attributed to an acyclic drift.

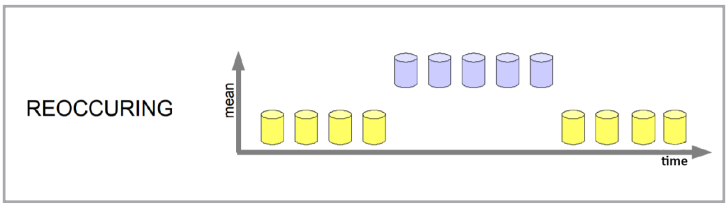

Figure 5. Periodic Drift

Source: Bifet, A., Gama, J., Pechenizkiy, M., & Zliobaite, I. (2011). Handling concept drift: Importance, challenges and solutions. PAKDD-2011 Tutorial, Shenzhen, China.

The *predictability* of drift indicates whether the drift is completely random or follows a pattern. A drift is predictable if it follows a certain set of rules, and unpredictable when its occurrence does not follow any set mechanism. In the figure below, (a) shows the centroid movement as being random, thus considering the drift to be unpredictable. However, (b) shows a linear movement of the centroid, in a predictable manner.

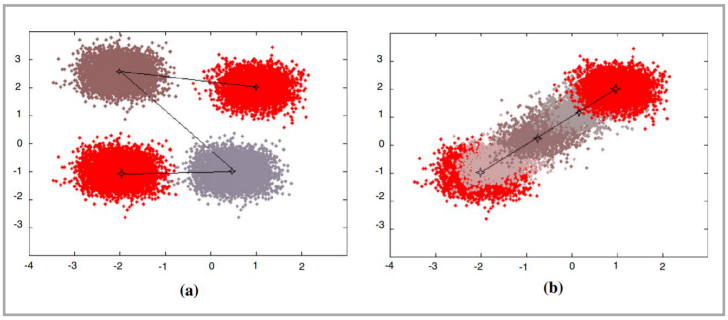

Figure 6. Predictability of Drift

Source: Khamassi, I., Sayed-Mouchaweh, M., Hammami, M., & Ghédira, K. (2018). Discussion and review on evolving data streams and concept drift adapting. Evolving systems, 9(1), 1-23.

4.

Capturing Drift Velocity

Several approaches have been defined to detect drift in applications, whose results allow for measures that would handle the drift and improve the model being monitored. Of these approaches, eight methods that lay the foundation for multiple drift detection models have been selected and analyzed, to understand the algorithms implemented and the inferences that could be gained from each approach. PACE-ML can employ these methods in the model monitoring stage, in order to capture the aforementioned elements of drift velocity.

A base learner (classifier) is initially used to classify the incoming instances. For each instance, the predicted class is compared against the true class label, and the result is used by drift detection methods to indicate if drift has occurred. Following this, the base learner is trained on the instance, and this occurs for every incoming instance.

4.1 Drift Detection Method (DDM)

The classification result of the base learner indicates whether the class has been predicted correctly (True) or incorrectly (False). An online-error-rate is computed using the results of classification; a decrease in the error-rate implies the classification was correct, an increase implies incorrect classification. The DDM makes the assumption that the base learner incorrectly classifies instances if the data distribution is different, which causes drift. Thus, while the distribution is stationary, the error-rate would decrease. The error-rate (p_i) and standard deviation ($s_i=\sqrt{(p_i(1-p_i)/i)}$) are calculated and these values are stored when p_i+s_i reaches its minimum, obtaining p_{min} and s_{min}. A warning level is reached when $p_i+s_i \geq p_{min} + 2 .s_{min}$, indicating a possibility of drift, and a drift level is reached when $p_i + s_i \geq p_{min} + 3 .s_{min}$, indicating a context change. At this point, the base learner, p_{min} and s_{min} are reset. A new base learner is trained on the incoming examples post warning level.

4.2 Early Drift Detection Method (EDDM)

The EDDM is similar to DDM; however, it uses the distance-error-rate as a metric to identify if drift has occurred, and computes the number of examples between two classification errors. The distance increases in the absence of drift, which means the base learner has improved its prediction accuracy. If drift occurs, the base learner has committed more mistakes, and the distance decreases. EDDM is applicable for slow gradual drifts as well as sudden drifts. The average distance between two errors (p_i) and its standard deviation (s_i) are computed. These values are stored when $p_i + 2. s_i$ reaches its maximum value, obtaining p_{max} and s_{max}. A warning level is reached when $(p_i + 2 .s_i) / (p_{max} + 2 .s_{max}) < \alpha\ (=0.95)$, indicating a possibility of drift, and a drift level is reached when $(p_i+2 .s_i) / (p_{max} + 2 .s_{max}) < \beta\ (=0.9)$, indicating a context change. At this point, the base learner, p_{max} and s_{max} are reset. A new base learner is trained on the incoming examples post warning level.

4.3 Page Hinkley Test (PHT)

This is a sequential analysis technique. The observed values, or the actual accuracy of the classifier, and their mean are computed until the current moment. In the event of an incorrect classification, the actual accuracy of the classifier decreases, thus indicating that drift has occurred. The average accuracy up to the current moment decreases as well. The cumulative difference U_T and minimum difference m_T between observed value and previous value are calculated; a higher U_T indicates a greater difference between the observed value and its previous value. When the difference between U_T and m_T is above a particular threshold, a change in distribution is detected.

4.4 ADWIN/ADWIN2 (Adaptive Windowing)

The ADWIN algorithm uses sliding windows of varying size in order to detect drift. A sliding window W contains recently read examples, and the distribution in sub-windows W_0 and W_1 are compared. If there are "distinct enough averages", a change in the distribution is detected and drift is said to have been detected. If drift occurs, the older sub-window W_0 is discarded and W_1 remains in window W. The size of the window is recomputed online based on rate of change observed from data in windows; it grows with no change and shrinks with change detection.

However, ADWIN is computationally expensive, since the algorithm compares all possible sub-windows within one window. ADWIN2 accounts for this by using a variation of exponential histograms and memory parameters. It limits the number of possible sub-windows in a window, and employs buckets for efficient usage of memory.

4.5 Paired Learners (PL)

This method makes use of two learners – a stable learner and reactive learner. The stable learner predicts based on all its experience, while the reactive learner predicts based on a window (of length w) of recent examples. A circular list of bits with length is updated with each classification result – 1 is stored if the stable learner makes an incorrect classification and the reactive learner is correct, and 0 otherwise. If the number of 1s cross a particular threshold, it indicates that the reactive learner, trained on recent examples, is more accurate than the stable learner. The understanding is that the reactive learner outperforms the stable learner when the target concept changes, and thus drift is said to have occurred. The reactive learner then substitutes the stable learner, and the circular list bits are reset to 0. Having a low w value would allow for sudden drifts to be detected easily, as the reactive learner would rapidly beat the performance of the stable one.

4.6 EWMA for Concept Drift Detection (ECDD)

This approach is based on Exponentially Weighted Moving Averages (EWMA), to identify an increase in the mean of a sequence of random variables. In EWMA, the probability of incorrectly classifying instance before the change point and standard deviation of the stream are known beforehand. EWMA with a Bernoulli distribution would imply a probability distribution of a random variable, which takes 1 with success p (correct classification) and 0 with failure q=1-p (incorrect classification). Here, values are computed online based on the accuracy of base learner, together with an estimator of expected time between false positive detections. Two estimations are then compared – one with more weight on recent examples, and another with similar emphasis on recent and old data. If the difference between estimations exceeds a particular threshold, a change is detected.

4.7 Statistical Test of Equal Proportions (STEPD)

This is a test to compare the classification accuracy in recent window with the historical accuracy of examples excluding this window. The assumption made here is if the target concept is stationary, accuracies will not change; a significant decrease in recent accuracy is an indicator of change. The accuracy of the base learner for *W* recent examples is compared to the overall accuracy of the learner. A chi-square test is performed by computing a statistic and its value is compared to the percentile of the standard normal deviation to obtain the significance level. If the value is less than a significance level, drift is said to have occurred.

4.8 Degree of Drift (DoF)

In this method, data is processed in chunks, and the instance in the current batch is used to find the nearest neighbor in the previous batch, after which their labels are compared. A distance map is created by associating the index of the instance in the previous batch and label computed by the nearest neighbor. A metric called the Degree of Drift is computed from the map. The average and standard deviation of all degrees of drift are calculated, and if the current value is away from average by more than s standard deviations, change is detected. Here, s is a parameter of the algorithm. This algorithm is well suited in detecting gradual drifts.

The following table captures the effectiveness of the various algorithms in capturing the various components of the drift velocity.

Algorithm	Speed		Severity
	Abrupt	Gradual	
DDM	✓		
EDDM	✓	✓	
ADWIN/ADWIN2		✓	
PHT	✓		
PL	✓		✓
ECDD		✓	
STEPD			
DoF		✓	

5.

Strategies to Handle Drift Velocity

5.1 Do Nothing

The most common method is to treat the model as a static model; i.e., we assume that there is no drift, thereby creating one “best” model and using it on all input data. This allows us to obtain a baseline model, which can be monitored over time for any reduction in performance accuracy that may occur as a consequence of drift.

5.2 Periodically Refit

Models are periodically retrained in a batch-based manner, using a sliding window of fixed size with the most recent historical data. The size of the window may be altered in order to capture the new relationships between input and output data. This strategy can be a simple but powerful way of addressing drift.

5.3 Incrementally Update

Alternatively, learning models may use a trigger to initiate a model update. This uses an incremental learning approach, where the existing model and a single new sample of data is used in updating the model. While the cost of retraining may become substantial, this will ensure the most recent models are always in production.

5.4 Regionally Adapt

For models which may have been applied in diverse environments, the organization may choose to address drifts locally, thus cleaning a region only if a false alarm is raised based on spatial information, and leaving the other regions unharmed. The entire feature space would be divided into a set of sub-feature spaces, to track and adapt to the drifts.

6.

Conclusion

In this study, we characterized drift along three key dimensions, enumerated the different algorithms for tackling drift, and highlighted four possible strategies for dealing with it. We have explained how capturing drift early and having a robust strategy in place for handling such drifts are essential activities required for putting ML models in production. Further, we have introduced, PACE-ML, which utilizes multiple state-of-the-art drift detection algorithms, for monitoring and measuring such drifts. We believe such tools and frameworks can help organizations reap the maximum benefits from their AI initiatives, even when external contexts may be changing.

7.

References

[1] Žliobaitė, I., Pechenizkiy, M., & Gama, J. (2016). An overview of concept drift applications. In Big data analysis: new algorithms for a new society (pp. 91-114). Springer, Cham.

[2] Webb, G. I., Hyde, R., Cao, H., Nguyen, H. L., & Petitjean, F. (2016). Characterizing concept drift. Data Mining and Knowledge Discovery, 30(4), 964-994.

[3] Khamassi, I., Sayed-Mouchaweh, M., Hammami, M., & Ghédira, K. (2018). Discussion and review on evolving data streams and concept drift adapting. Evolving systems, 9(1), 1-23.

[4] Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2018). Learning under concept drift: A review. IEEE Transactions on Knowledge and Data Engineering, 31(12), 2346-2363.

[5] Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004, September). Learning with drift detection. In Brazilian symposium on artificial intelligence (pp. 286-295). Springer, Berlin, Heidelberg.

[6] Baena-Garcıa, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavalda, R., & Morales-Bueno, R. (2006, September). Early drift detection method. In Fourth international workshop on knowledge discovery from data streams (Vol. 6, pp. 77-86).

[7] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. ACM computing surveys (CSUR), 46(4), 1-37.

[8] Bifet, A., & Gavalda, R. (2007, April). Learning from time-changing data with adaptive windowing. In Proceedings of the 2007 SIAM international conference on data mining (pp. 443-448). Society for Industrial and Applied Mathematics.

[9] Webb, G. I., Lee, L. K., Goethals, B., & Petitjean, F. (2018). Analyzing concept drift and shift from sample data. Data Mining and Knowledge Discovery, 32(5), 1179-1199.

[10] Minku, L. L., White, A. P., & Yao, X. (2009). The impact of diversity on online ensemble learning in the presence of concept drift. IEEE Transactions on knowledge and Data Engineering, 22(5), 730-742.

[11] Losing, V., Hammer, B., & Wersing, H. (2018). Incremental on-line learning: A review and comparison of state of the art algorithms. Neurocomputing, 275, 1261-1274.

[12] Bifet, A., Gama, J., Pechenizkiy, M., & Zliobaite, I. (2011). Handling concept drift: Importance, challenges and solutions. PAKDD-2011 Tutorial, Shenzhen, China.

Authors

Dr. Archisman Majumdar

AVP & Lead - Applied AI, Mphasis NEXT Labs

Dr. Archisman leads a cross-functional team of Data Scientists and consults Fortune 500 companies on AI and ML implementations. He holds a PhD from the Indian Institute of Management Bangalore (IIMB) in the Quantitative Methods and Information Systems area. His areas of expertise are in machine learning, product management, and information systems research.

Keerthana Sundaresan

Intern, Mphasis

Keerthana Sundaresan is an intern with Mphasis. Her domains of interest include AI, ML, Data Science and Cyber Security, and she is working towards a deeper understanding of the fields. In her free time, she learns Bharatanatyam, enjoys photography and writing, and is also a core member of her college magazine and theatre troupe.

About Mphasis

Mphasis (BSE: 526299; NSE: MPHASIS) applies next-generation technology to help enterprises transform businesses globally. Customer centricity is foundational to Mphasis and is reflected in the Mphasis' Front2Back™ Transformation approach. Front2Back™ uses the exponential power of cloud and cognitive to provide hyper-personalized ($C = X2C^{2}_{TM} = 1$) digital experience to clients and their end customers. Mphasis' Service Transformation approach helps 'shrink the core' through the application of digital technologies across legacy environments within an enterprise, enabling businesses to stay ahead in a changing world. Mphasis' core reference architectures and tools, speed and innovation with domain expertise and specialization are key to building strong relationships with marquee clients. To know more, please visit www.mphasis.com

For more information, contact: marketinginfo.m@mphasis.com

USA
460 Park Avenue South
Suite #1101
New York, NY 10016, USA
Tel.: +1 212 686 6655

UK
1 Ropemaker Street, London
EC2Y 9HT, United Kingdom
T : +44 020 7153 1327

INDIA
Bagmane World Technology Center
Marathahalli Ring Road
Doddanakundhi Village
Mahadevapura
Bangalore 560 048, India
Tel.: +91 80 3352 5000

NR 31/08/20 US LETTER BASIL6386

www.mphasis.com