ML MODELS FOR FORECASTING

Full Name	Category	Description	Benefits	Considerations	Use Cases
Autoregressive Integrated Moving Average (ARIMA)	Statistical	Uses time series data to predict future values based on past values.	Relatively simple to execute. No "black box."	Limited to a single variable. Linear models don't capture nonlinear patterns.	Status quo forecasting — relevant for understaffed teams or brands without data scientists.
Autoregressive Integrated Moving Average with Exogenous Variable (ARIMAX)	Statistical	ARIMA with the flexibility to include additional variables.	Still simple but adds a layer of sophistication.	Linear models don't capture nonlinear patterns.	Brands that want to keep it simple but have additional data that might increase forecast accuracy.
Prophet	Statistical	Easy-by-design approach that builds on time series approaches by adding seasonality elements.	Still simple but adds a layer of sophistication. Open-source.	Can perform worse than more basic options when seasonality is not the primary driver of outcomes.	Brands looking for moderately advanced self-serve options. Products that are primarily impacted by seasonal trends.
Exponential smoothing	Statistical	Cousin of ARIMA that weights recent observations more highly.	Flexible enough to add trends or seasonality.	Not suitable for complex trend patterns. Can be tough to balance accuracy when adding trends/seasonality.	Brands with a small data science team that want to go beyond the basics.
Adaptive Boosting (AdaBoost)	Ensemble	Combines multiple "weak" predictors to form a strong predictor.	Good protection against overfitting.	Sensitive to noisy data and outliers.	Brands with a moderate data science team that have the advanced skills required to manage hyperparameters.

Full Name	Category	Description	Benefits	Considerations	Use Cases
Gradient Boosting (GB)	Ensemble	Evolution of AdaBoost that is more flexible in its use.	Typically outperforms comparable models. Interpretable (little to no "black box").	Susceptible to overfitting on training data. Computation-heavy approach.	Brands with a moderate data science team that have the advanced skills required to manage hyperparameters.
Random Forest (RF)	Ensemble	Combines multiple "decision trees."	Uncorrelation between trees improves accuracy. Can handle categorical and numerical data.	Low interpretability ("black box").	Brands with a moderate data science team but that don't have the time or skills required to closely monitor or tune models.
Multi-Layer Perceptron (MLP)	Neural Network	Connects inputs and outputs in a non-linear way.	Works well with large data sets. Works with non-linear data.	Requires hands-on fine-tuning and extensive training. Computation-heavy approach.	Brands with large data science teams who want easier-to-train models + are willing to accept worse performance with sequential data.
Recurrent Neural Network (RNN)	Neural Network	Uses memory to connect past inputs with current inputs/ outputs.	Good at learning short-term relationships between inputs.	Challenges retaining relevant information over long sequences. Long training time. Difficult to learn complex patterns.	Brands with large data science teams willing to accept the increased difficulty of training.
Temporal Convolutional Network (TCN)	Neural Network	Applies convolutional tactics to sequential data.	Can compute outputs in parallel, which speeds things up. Avoids common problems experienced with RNNs.	Requires large amounts of labeled data.	Brands with large data science teams willing to accept the increased difficulty of training.
Transformers	Neural Network	One of the most cutting-edge options. Maps relationships between sequential data to uncover context.	Doesn't require labeled training data. Can compute outputs in parallel, which speeds things up.	Incredibly computation-heavy approach. Don't inherently prioritize short-term patterns.	You work for a major tech company like Google or Nvidia. Or you have access to a pre-trained transformer model that can be tailored for your needs.