dotData

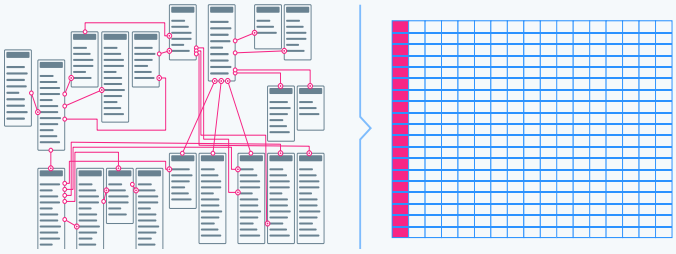
Revolutionize AI Development on Python Through Augmented Feature Discovery

AI/ML models are only as good as the data (features) you use. Feature engineering is critical in discovering patterns hidden in your data but needs interdisciplinary skills like data engineering, statistics, and domain knowledge. The feature discovery process is iterative in nature and the complex relationship between domain expertise and technical skill makes feature engineering a difficult task even for experienced data scientists.

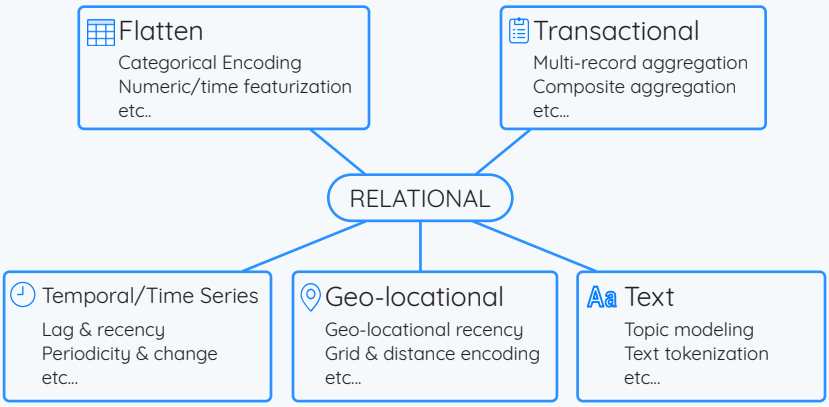
> *Coming up with features is difficult, time-consuming, and requires expert knowledge. Applied machine learning is basically feature engineering* - **Andrew Ng**

Deliver Deeper Insights and Better ML Models

"Features" determine the success or failure of your ML models. In machine learning theory, Feature Engineering involves transforming explanatory variables to extract higher-order statistics. In practical machine learning projects, however, multiple tables with complex relationships mean new use-cases require data consolidation and manipulation to build feature tables.



dotData's Automated Feature Engineering (AutoFE) augments your feature discovery by surfacing multi-modal patterns from enterprise data. dotData supports traditional techniques like numerical histogram flagging, one-hot or target encoding, or weekday/weekend flagging but also identifies multi-modal patterns like seasonality in temporal transactions or geo-based topic distribution in geo-stamped text data. Leverage dotData to bring new insights and ideas to help you build better ML models.



Make ML Models Explainable, Actionable, and Accountable

Mathematically complex features often overwhelm business users. Even if your features are highly correlated to prediction targets, your models might be ignored if business users don't understand how to take preventive measures. Add transparent and explainable features to your models to gain the insights needed to take preventive measures – driven by ML models.

dotData discovers "white-box" features and makes them accessible in Python for easy model customization:

- **Feature Explanations:**
  A natural language expression of each feature allowing data scientists to intuitively understand the meaning of each feature.
- **Feature Blueprints:**
  A lineage diagram that visually explains the feature generation process for each feature
- **Feature Scores:**
  A quantitative measure of the validity and statistical relevance of each feature to provide objective evidence

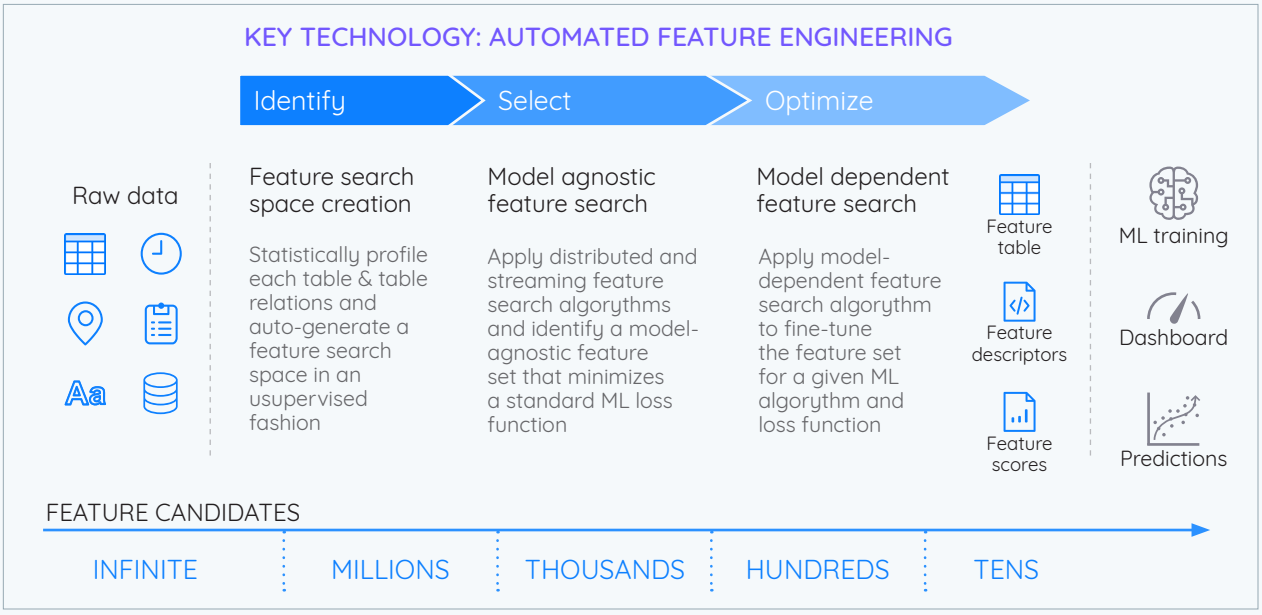
How Does Automated Feature Engineering Work?

dotData's unique and proprietary feature engineering algorithm was invented after 10+ years of research at NEC Laboratories, from which dotData was spun out. It consists of multiple stages:

- **Feature Identification:**
  Identify up-to millions of feature hypotheses by profiling the input tables and their relationships.

- **Feature selection:**
  Select up-to hundreds of model-agnostic feature hypotheses by minimizing a ML loss function for a given prediction problem

- **Feature optimization:**
  Optimize feature hypotheses for a specific ML algorithm, narrowing the resulting feature set to (typically) around 100 features that are finely tuned to a given ML algorithm



Feature Engineering: Technology Highlights

- **Advanced regularization theory:**
  dotData's AutoFE algorithm is based on advanced regularization and sparse learning theory to help mitigate feature overfitting and collinearity, minimizing one of the biggest challenges with large feature spaces.

- **Combinatorial features across multiple tables:**
  dotData's AutoFE technology combines attributes in multiple tables - even with different cardinality - discovering new patterns by leveraging the rich and diverse types of data sources.

- **Built-in Data & Feature Cleansing:**
  dotData automatically identifies illegal data values, systematic or statistical missing values, outliers, categorical value canonicalization, record duplication and more to maximize the quality of features

- **"Leaky Feature" Prevention:**
  "Leaky features" happen when temporal (time-stamp) data might refer to a future date point in place of a historical date. dotData lets you configure data lead time and prediction lead times to automatically prevent this type of data leakage

How Our Clients Use dotData

- SMBC scaled their AI practice from 2,000 features per year to 2,000,000+ - without adding resources while improving model performance by 30%
- Payment processor sticky.io developed an AI model in 90 days to help recover $8M per month in declined transactions
- A national Accountable Healthcare Organization (ACO) found $1.4M in monthly savings by proactively engaging “at risk” patients
- A global insurance company boosted policy add-ons by 2.5X and to achieved 10X faster development cycles with AI
- A global retail chain increased coupon usage rates by 15% across 10,000+ locations and was able to run 3X more campaigns

- **Auto-determination of temporal/geo-locational aggregation:** dotData’s AutoFE automatically profiles different aggregation ranges to determine optimal range values in temporal and geo-location data. This is especially useful when analyzing use-cases where aggregation ranges may not be obvious - like “customers who recently visited the website.” In these scenarios, dotData will determine the optimal “recency” range.
- **Distributed and stream feature selection:** dotData’s AutoFE scales to handle billions of raw records (rows) as well as millions of columns (feature hypotheses) employing a unique distributed and stream feature generation and selection techniques.

Seamlessly Integrated into Your Python Environment

dotData Py provides flexible deployment options regardless of your situation. Deploy on a clustered Python environment to seamlessly scale with your AI/ML development needs or on Databricks with seamless integration, including with the Databricks Feature Store - for maximum flexibility in combining manually built features with automatically generated ones.

For maximum portability and deployment, dotData Py Lite provides a container-based model that can be easily deployed on any environment that supports Docker containers - even a laptop.

	dotData Py	dotData Py on Databricks	dotData Py Lite (Containerized)
Resource Units (RU)	1 Core & 8GB of RAM		1 Core & 2GB of RAM
License Unit	1 Node = 7 Resource Units		1 License = 1 Resource Unit
Base Platform	Hadoop & YARN / Amazon EMR	Databricks	Docker & Docker Compose
Data Scalability	Scale out via node count		Scale up via container size
Infrastructure	AWS or Azure	AWS or Azure	AWS/Azure/On-Premise
Model Production	Supported		

dotData

2121 S. El Camino Real #B100 San Mateo, CA 94403
www.dotdata.com contact@dotdata.com