# Machine Learning meets Databases

## Ioannis Papapanagiotou – ipapapa@ncsu.edu

Engineering Manager, Cloud Database Engineering, Netflix

At Netflix, we embrace personalization and algorithmically adapt many aspects of our member experience, including the rows we select for the homepage, the titles we select for those rows, the galleries we display, the messages we send, and even the images you see. We collect billions of data points from impressions, viewing history, actions, as well a number of metadata from movies that our studios produce, to generate a personalization story for each user. With a catalog spanning thousands of titles and a diverse member base spanning over a hundred million accounts, recommending the titles that are just right for each member is crucial. We have therefore developed a software architecture that can deliver an enhanced experience and support rapid innovation. Our architecture handles large volumes of existing data, is responsive to user interactions, and makes it easy to experiment with new recommendation approaches. At the same time, Netflix as a cloud-native enterprise runs all of its infrastructure on the public cloud.

To achieve these goals, we run online and offline processing services. For example, real-time model scoring, real-time stream processing as well as offline services using data lakes in Amazon S3. We also leverage pub/sub-frameworks like Apache Kafka, we transform data and store them in one of the largest Apache Cassandra deployments or in our caching layers which are comprised of a globally replicated in-house Memcached deployment called EVCache. Hence, there are multiple data sources depending on the ML use cases and their respective SLAs. Real-time, latency sensitive, and with high refresh rate ML use cases tend to be served by our caching infrastructure. We prefer to store structured data, like viewing history, in Cassandra because we can horizontally scale the writers per second, apply customer filters (user, device, subtitle, episode, season, actor etc) given the columnar format and can use tunable consistency to tradeoff performance vs data consistency. We also use Apache Parquet to leverage efficient columnar data representation that aids in reduced I/O for large-scale reads.

In this talk, we are going to cover how the Netflix Machine Learning infrastructure leverages the data infrastructure and the technology choices we have made to offer personalization to close to 150M subscribers across the world. More specifically, we are going to look at the scalability choices that influences the format and the schema of the data being used for offline feature generation and model training, and how the data is accessed through latency-sensitive services for online model scoring. Our story does not end here, as we are growing across many directions (subscribers, catalog, countries etc.) we must consider efficiencies at a large scale.

Resources:

- Artwork Personalization
- Distributed Time Travel for Feature Generation
- Learning a Personalized Homepage