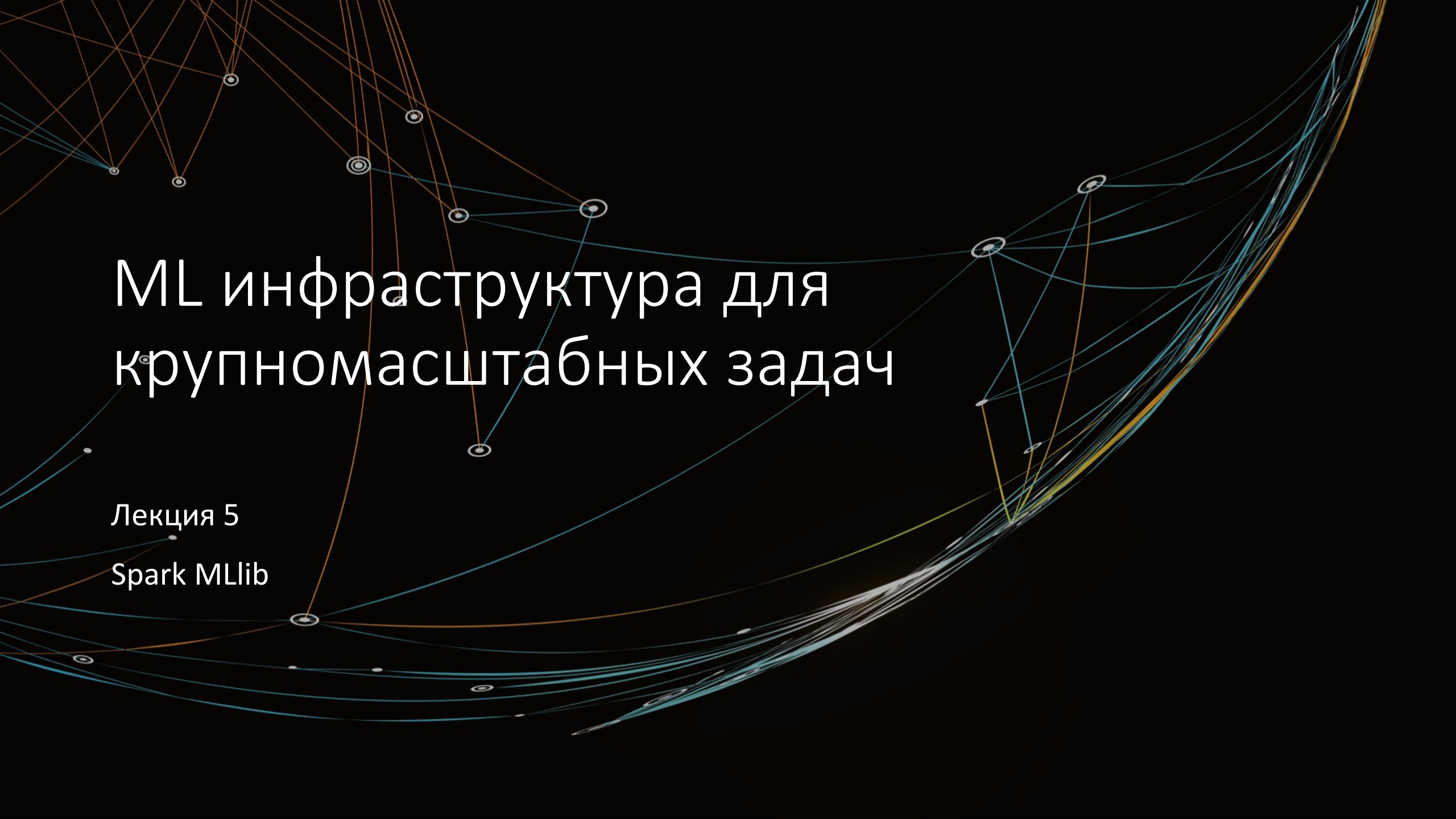

ML инфраструктура для крупномасштабных задач

Лекция 5

Spark MLlib

О чем поговорим

- DataFrame-based or RDD-based
- Что есть в MLlib

Mllib: DataFrame vs RDD

Начиная с версии Spark 2.0 RDD версия Mllib (spark.mllib) находится в стадии поддерживания, но основной является версия на основе DataFrame (spark.ml).

Что сейчас происходит с двумя версиями:

- Mllib на основе RDD до сих пор поддерживается на уровне bag-fixes;
- В Mllib на основе RDD не добавляется ничего нового;
- Mllib на основе DataFrame активно развивается, в версиях 2.x основной задачей было догнать функционал RDD версии.

Mllib: DataFrame vs RDD

А зачем DataFrame в ML?

- DataFrame предоставляет более user-friendly API в отличии от RDD;
- Преимущества DataFrame используются для построения моделей и оптимизации;
- DataFrame позволили сделать единое API для всех моделей и различных языков программирования;
- Появились Pipelines.

Состав MLlib

- ML алгоритмы: регрессия, классификация, кластеризация и коллаборативная фильтрация;
- Работа с фичами: преобразования, снижения размерности и прочее;
- Pipelines: инструменты для преобразования данных, обучения моделей и подбора параметров;
- Persistence: сохранение и загрузка алгоритмов, моделей и пайплайнов;
- Полезные вещи: линейная алгебра, статистика и прочее.

Mllib: вектора и матрицы

- Vector – основной тип данных для Mllib, по сути похож на numpy array;
- SparseVector;
- Matrix;
- SparseMatrix.

MIlib: работа с фичами

- Разбиение непрерывных переменных на бакеты
- Countvectorizer
- Tf-idf
- Hashing
- Index to string
- OHE
- Разные scalers
- N-grams
- PCA
- Stopwords
- word2vec

Mllib: модели

- SVC
- Linear\Logistic Regression
- Decision Tree
- Random Forest
- GBT
- NaiveBayes
- Factorixation Machine
- ALS
- Isotonic Regression
- Разные виды Kmeans
- GaussianMixtire
- Разные виды LDA

Mllib: тюнинг моделей

Тюнинг идет по сетке через ParamGridBuilder - аналог GridSearch с двумя вариантами:

- CrossValidation
- TrainValidationSplit

+

Отдельные классы для оценки качества моделей, базовый Evaluator

Pipeline

DataFrame: ML API использует DataFrame из Spark SQL как dataset, в котором может храниться множество различных типов данных. Например, признаки, метки классов, предсказания.

Transformer: Transformer это алгоритм, который может преобразовать один DataFrame в другой DataFrame. Например, ML model это Transformer, который преобразовывает DataFrame с признаками в другой DataFrame с предсказаниями.

Estimator: Estimator это алгоритм, который может быть обучен на DataFrame, чтобы создать Transformer. Например, алгоритм обучения это - Estimator, который обучается на DataFrame и создает модель.

Pipeline: Pipeline соединяет в цепочку несколько Transformers и Estimators вместе, чтобы задать ML workflow.

Parameter: Все Transformers и Estimators теперь имеют общее API для спецификации параметров.

Сохранение

Через класс с приставкой Model можно загружать сохраненные модели

```
from pyspark.ml.classification import LogisticRegression, LogisticRegressionModel

lr = LogisticRegression(featuresCol='features', labelCol='Is_Lead', predictionCol='prediction',
maxIter=100, probabilityCol='proba')
lr = lr.fit(transformed_data)

lr.save('logreg_model’)

lr2 = LogisticRegressionModel.load('logreg_model')
```