

Analyzing risk for Insurance Companies

By -

Aditya Waghlikar

Heli Vora

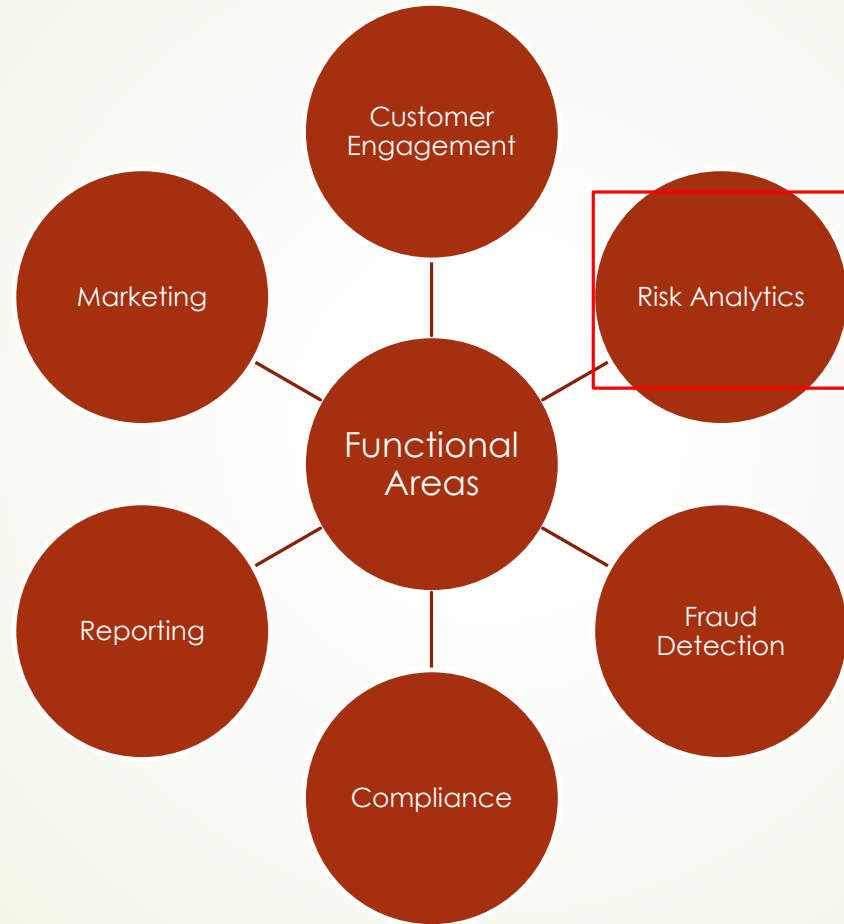
Subodh Ghuge

Sushree Sucharita

Contents

- Overview of the Insurance Industry
- Project Scope
- Why NoSQL
- Why MongoDB
- Demo and Explanation
- Challenges Faced
- Lessons Learnt
- Future Scope
- Questions

Overview Of The Insurance Industry



Different Data Stores can be used for the different applications (Polyglot Persistence)

Project Scope

- Insurance companies need to keep a track of different factors to evaluate whether to provide insurance to a certain customer and to determine the premium. These factors are:
 - Age
 - Medical History
 - Employment History
 - Family Background
 - Natural Calamities
- The aim of the project is explore MongoDB to store this humongous data and evaluate the risk based on these factors
- We got our Data from **Kaggle Prudential Life Insurance** challenge



Fun Fact – “**MongoDB**” is derived from the word “**humongous**” which means enormous

Why NoSQL?

Challenges with Relational Databases	How NoSQL can be a solution	Impact/Results
No single view of a customer	Can help Aggregate data at customer level	Unified customer view, better user experience
Customer information change rapidly; Modification of schema to accommodate these changes is difficult	Schema-less databases provides flexibility	Can store different information for different customers; increased efficiency in risk modelling
Data is very huge; scaling up the system could be expensive	Scale – out using sharding; different parts of data can be put on different nodes	Data is sharded based on locations; Speed of retrieval increases, load on server is balanced

Why MongoDB?

- Data is stored in form of documents; A collection can have large number of documents and they can hold non-uniform data
- Format of data can be different. Rich GeoJSON structures can be used to store location data

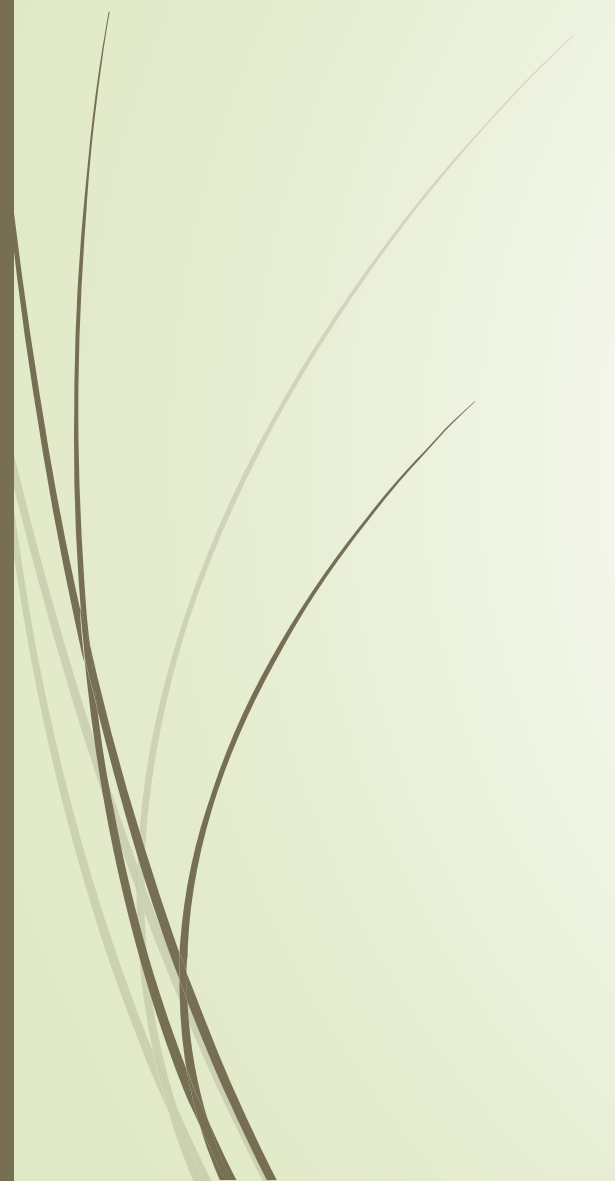
```
{
  customer_id : "1",
  first_name : "Matt",
  last_name: "Kram",
  medical_history : ["Arthritis", "Diabetes"],
  city : "Princeton"
}
```

```
{
  customer_id: "2",
  first_name : "Cara",
  Age : "27",
  Is_Employeed : "Yes",
  location: {
    type: "Point",
    coordinates: [-73.856077, 40.848447]
  }
}
```

Continued..

- Rich Querying Language; Based on JSON
- Aggregate functions provided help perform mathematical/analytical operations
- Map Reduce operations facilitate aggregation of data across shards
- Replica sets prevent loss of data due to unforeseen circumstances thereby ensuring consistency
- **Geospatial queries** assist in real time location based analytics that help model risk due to natural hazards
- RMS, a provider of catastrophe risk modeling software company, has selected MongoDB as the data store for **RMS(one)**. The company's new MongoDB-based risk analytics solution will enable insurance companies to plan new growth strategies and manage their capital more effectively.

Demo and Explanation



Challenges Faced

- Integrating Java, MongoDB and Gradle
- Data Cleaning and Feature Scaling
- Interface between Source Data and Application Logic
- Modularizing the code
- Creating custom libraries for modelling customer risk

Lessons Learnt

- Understand functionality of the Insurance Industry
- Develop understanding regarding NoSQL databases
- Integrating Eclipse and MongoDB
- Types of data MongoDB can store (JSON, GeoSON)
- Features provided by MongoDB – Consistency, Map Reduce
- Detailed knowledge regarding functionalities provided by MongoDB queries (Aggregation, Geospatial)

Future Scope

- Including other factors like Criminal History, Educational Background etc. while modelling the risk
- Utilizing data from the RMS(one) data store to accurately model risks due to natural hazards
- Integrating other functional areas of the Insurance Industry and making a complete system

Citations

- <https://docs.mongodb.com/manual/mongo/>
- <https://www.slideshare.net/mongodb/how-insurance-companies-use-mongodb>
- <https://www.mongodb.com/webinar/insurance-comp-july-2014>

Questions?

THANK YOU