COMPREHENSIVE UNDERSTANDING OF THE J&K AGRISTACK LAND RECORD DIGITIZATION CHALLENGE

THE BIG PICTURE: WHAT IS AGRISTACK AND WHY DOES IT MATTER

AgriStack is India's national digital infrastructure for agriculture. Think of it as the "Aadhaar for farming" - a foundational layer that connects farmers, their land, and government services. It requires two critical building blocks: a Farmer ID (unique digital identifier for each farmer linked to Aadhaar and land holdings) and a Farm/Plot ID (unique geo-referenced identifier for each land parcel). Without clean, structured land records, neither can be created. This blocks the entire downstream ecosystem of digital agriculture services, targeted subsidies, crop insurance, credit access, and data-driven policy in Jammu & Kashmir.

THE CORE PROBLEM: WHY J&K'S LAND RECORDS ARE A NIGHTMARE

Jammu & Kashmir's land records exist primarily as Jamabandi documents - official government records maintained by the Revenue Department. Here is what makes them uniquely challenging:

First, the language barrier. All records are written in Urdu, which uses Arabic script and reads right-to-left. This immediately makes standard OCR and text processing tools ineffective.

Second, the format chaos. Records exist in three states: physical paper documents, scanned PDFs with inconsistent quality, and PDF outputs from a legacy LRIS (Land Record Information System) whose backend database is inaccessible. The government cannot simply query a database - they have to work with what they can see on the front-end.

Third, the historical complexity. J&K has undergone significant land reforms and redistribution over decades, creating fragmented ownership patterns. You have multiple co-owners for single parcels, farmers owning non-contiguous land across different villages, and disputed categories like refugee/muhajireen land where ownership is unclear.

Fourth, the scale. There are over 6,000 villages in J&K, each with 50 to 500 pages of land records. That is potentially 3 million pages. At the ~8 crore (80 million) page estimate mentioned in the problem statement, manual digitization would take years and introduce massive errors.

UNDERSTANDING THE JAMABANDI DOCUMENT STRUCTURE

A Jamabandi is a snapshot of land ownership at a specific point in time. The documents you have been given are from 2017-18. Each document follows a 12-column tabular format that reads from right to left (columns 1 through 12):

Column 1 is Number Khevat. A revenue village is divided into Khevats (subdivisions). This is the broadest grouping.

Column 2 is Number Khata. Within each Khevat, holdings are given Khata (account) numbers. This is the primary row delimiter - each Khata should become a logical grouping in your output.

Column 3 is Nam Tarf Ya Patti Meh Nam Number Dar. This identifies the Nambardar, a government official who represents a group of landowners.

Column 4 is Nam Malik Meh Ahval. This contains the old/historical owners of the Khevat. The total land holdings of these old owners should add up to the entire Khevat's surveyed land.

Column 5 is Nam Kashtakar Meh Ahval. THIS IS THE CRITICAL COLUMN. It contains the details of current cultivators/title holders. This is where you need to extract Name, Parentage, Caste, Residential Village, and ownership type.

Column 6 is Vasayil Abapashi. The means of irrigation.

Column 7 is Number Khasra V Nam Khet. The survey number of the land parcel. It has two sub-parts: hal (current survey number) and sabik (previous survey number). Each individual khasra number represents a distinct land parcel.

Column 8 is Raqba Bakiyad Kisam. The area and type of land. It has sub-parts: kanal (a unit of area), marala (a smaller unit), and kisam zamin (type of land like nahri, gora nahri, hil nahri).

Column 9 is Lagan Jo Mujariya Ada Karta Hai Meh Sharah Va Tadad. Description and amount of tax paid.

Column 10 is Mutalba B Tashari Maal Va Sewai. Cess levy on goods and services.

Column 11 is Havala Intakal. Mutation reference number - this tracks transactions that have occurred on the land like sales, gifts, inheritance.

Column 12 is Kaifiyat. Remarks.

WHY THE TRANSLITERATED VERSION WAS PROVIDED

The Atmapur.pdf file is the original land record in Urdu script. When you try to extract text from it, you get right-to-left Arabic script that is extremely difficult to parse programmatically.

The TransliteradVersion_Village_Gujral_-_Jamabandi.pdf is the same data but converted to Roman script (English alphabet) while preserving the phonetic sounds. For example, the Urdu "کاشت سہید و سنگھ پسر اتر سنگھ" becomes "kasht sahid v singh pisar attar singh" in transliteration.

This transliterated version is your PRIMARY INPUT for building the solution. It maintains all the information but in a format that standard text processing tools can handle. The government has already done this transliteration work for you - your job is to build the extraction and structuring pipeline.

THE SPECIFIC EXTRACTION CHALLENGES THEY WANT SOLVED

Challenge 1: Row Splitting Based on Khata Number. Each unique Khata number in column 2 should define a row boundary. Currently, the PDF layout makes rows span visually across multiple lines, causing confusion about where one record ends and another begins.

Challenge 2: Parsing Column 5 (Nam Kashtakar Meh Ahval). This is the most complex extraction. You need to split this single text field into five structured columns:

Name - the string that follows "kasht". Example: in "kasht sahid v singh pisar attar singh kaum sukh sakindeh gair morosi", the name is "sahid v singh".

Parentage - the string that follows relationship markers. The markers are: pisar/pisaran (son of/sons of), dukhtar/dukhtaran (daughter of/daughters of), zoja (wife of), byuh (widow of). Example: "pisar attar singh" means son of attar singh.

Caste - the string that follows "kaum". Example: "kaum sukh" means caste is sukh.

Residential Village - the string that follows "sakin" or "sakindeh". Sakindeh specifically means resident of the same village.

Remarks/Ownership Type - remaining descriptors like "gair morosi" (legal non-heirs), "alati" (temporary), "morosi" (hereditary).

Challenge 3: Splitting Khasra Numbers Into Separate Rows. When a farmer owns multiple land parcels, they appear together in the PDF with multiple khasra numbers. In the example from the data description, you see khasra numbers 162 and 166 listed together for the same farmer. Each of these needs to become its own row in the Excel output, with the farmer information duplicated.

The row marked "kitta" is a SUM row - it aggregates the total area across all parcels for that farmer. This needs to be identified and handled appropriately (either kept as a summary row or excluded from the individual parcel listing).

Challenge 4: Column Border Misalignment. The PDF table structure does not have consistent column borders. When extracted, text from different columns can bleed into each other. Your solution needs to handle this spatial parsing intelligently.

Challenge 5: Eliminating Duplicate Khata Numbers. There are cases where the same person and land parcel appear in multiple cells due to duplicacy in Khata numbers. These duplicates need to be detected and consolidated.

UNDERSTANDING THE DATA FILES YOU HAVE

1. Atmapur.pdf (4.2 MB, 101 pages): This is an original land record in Urdu. It demonstrates what the raw source looks like. You cannot easily process this directly.
2. TransliteradVersion_Village_Gujral_-_Jamabandi.pdf (752 KB, 209 pages): This is the transliterated version of a different village (Gujral). This is your workable input format. The header shows "2017-18 : sal | Jammu :zla | Jammu West :thsil | Gujral :babata mouza | jamabandi" meaning it is the 2017-18 Jamabandi for Village Gujral in Jammu West Tehsil of Jammu District.
3. Data_Description_for_Jammu___Kashmir_Land_Records.pdf: This is the instruction manual. It explains the objective, data requirements, column meanings, and key asks. This is your Bible for this hackathon.
4. 1__Scheme_-*Land_Record_xlsx*-_Output_Expected.csv: This shows the expected output schema. The columns map to the 12 Jamabandi columns, with notes indicating where transformations are needed (comma delimiters for column 5 splitting, row splits for columns 7-8).

THE KEY TERM MAPPINGS YOU MUST IMPLEMENT

These Urdu terms (now in Roman transliteration) are your parsing signals:

Kasht = cultivator (indicates start of name) Pisar = son of Pisaran = sons of Dukhtar = daughter of Dukhtaran = daughters of Zoja = wife of Byuh = widow of Kaum = caste Namalum = unknown Sakin = resident Sakindeh = resident of the same village Bayaan = seller Mushtari = buyer Wahib = gifter Mohoob allya = receiver of gift Morosi = hereditary/heirs Gair morosi = non-heirs

---

WHAT YOUR SOLUTION NEEDS TO DELIVER

For the first submission (due December 15), you need:

1. A working prototype that takes the transliterated PDF, extracts the data, and outputs a structured Excel file matching the schema.

2. Python code (preferred) implementing the extraction logic.
3. A 5-7 page policy brief explaining the problem context, your technical approach/algorithm, and proposed solution.

For the final submission, you will need to add a robust implementation plan and scale-up strategy.

THE EVALUATION CRITERIA TO OPTIMIZE FOR

Scalability (25%): Can your solution handle 6000+ villages with 50-500 pages each? Is it automated enough to run without human intervention?

Innovativeness (25%): Are you just doing basic PDF parsing, or are you using intelligent NLP, machine learning, or clever algorithmic approaches to handle the complexity?

Impact (20%): Does your solution actually solve the core problem of creating a farmer and farm registry?

Feasibility (20%): Is your solution technically sound, cost-effective, and implementable without extensive dependencies?

Presentation (10%): Is your policy brief clear, well-structured, and convincing?

STRATEGIC RECOMMENDATIONS FOR YOUR APPROACH

Given your RAG and LLM background, consider these angles:

The core task is essentially Information Extraction from semi-structured documents. You could approach this as a Named Entity Recognition problem where your entities are Name, Parentage, Caste, Village, and Ownership Type, with the Urdu terms serving as entity markers.

For PDF parsing, consider using libraries like pdfplumber, camelot, or tabula-py that handle table extraction. The key insight is that even though the PDF looks like a table, the underlying structure may be positional (based on x-y coordinates) rather than semantic.

For the Column 5 parsing, a rule-based regex approach using the term mappings will likely work well since the language is formulaic. Something like: "kasht ([^p]+) pisar ([^k]+) kaum ([^s]+) sakindeh (.+)" could capture the major components.

For handling multiple khasra numbers, you need to identify rows where column 7 has multiple numbers (potentially separated by newlines or spaces in the extracted text), then programmatically duplicate the farmer information for each khasra while assigning the corresponding area from column 8.

The "kitta" keyword identifies sum rows which aggregate multiple parcels. You can use this as a signal to know when one farmer's complete parcel list ends.

This is a genuinely impactful problem. If solved well, it unblocks digital agriculture for an entire Union Territory and could serve as a template for other states with similar challenges. The technical complexity is moderate but the scale and edge cases make it interesting. Your background in document processing, RAG architectures, and production AI systems positions you well for this.