FRAUD DETECTION SYSTEM FOR HEALTHCARE TRANSACTIONS

Northwest Missouri State University

Aasrita Emani, Nikitha Ravella, Niharika polu, Sirisha Panuganti, Priyanka Godugunuri

Contents

1 Introduction

Fraud detection systems in healthcare transactions are crucial for identifying and preventing fraudulent activities, which can lead to substantial financial losses and compromise patient care. These systems leverage advanced analytics, machine learning algorithms, and data mining techniques to analyze vast amounts of healthcare transaction data in real-time. By identifying patterns and anomalies indicative of fraudulent behavior, such systems help healthcare organizations mitigate risks and safeguard their financial integrity.

For instance, in our project, a dataset from Kaggle was utilized to develop a fraud detection system using PySpark. By leveraging PySpark's distributed computing capabilities, the project aimed to efficiently process large volumes of healthcare transaction data and implement robust fraud detection algorithms, thereby enhancing the security and reliability of healthcare transactions.

2 Implementation Steps

2.1 Data Acquisition from Kaggle:

For our healthcare fraud detection project, we acquired a dataset from Kaggle, a popular platform for datasets and machine learning competitions. The dataset contains a comprehensive collection of healthcare transaction records, including details such as patient demographics, medical procedures, provider information, billing amounts, and payment methods. This dataset serves as the foundation for our fraud detection system, enabling us to analyze transaction patterns, identify anomalies, and develop machine learning models to predict fraudulent activities.

By leveraging this dataset, we aim to build a robust fraud detection system that enhances the security and integrity of healthcare transactions, ultimately safeguarding against financial losses and ensuring the delivery of quality patient care.

2.2 Data Preprocessing:

In the data preprocessing stage for our healthcare fraud detection project, we conducted several steps to prepare the dataset for analysis. Firstly, we performed an initial exploration of the dataset to understand its structure and identify any irrelevant or redundant columns. Subsequently, we removed columns that were not relevant to our fraud detection objectives, such as identifiers or metadata that did not contribute to the predictive modeling process. Additionally, we addressed missing values by either imputing them using appropriate techniques or removing rows with significant missing data, ensuring the integrity of our analyses.

Furthermore, we standardized numerical features and encoded categorical variables to facilitate model training and improve algorithm performance. Through these preprocessing steps, we streamlined the dataset, optimized its

usability for machine learning algorithms, and prepared a clean and structured dataset for further analysis and model development.

3 Goals of the Project

We have taken six goals for our healthcare fraud detection project. Here are those goals:

- What is the average Medicare allowed amount for each HCPCS code?
- Average difference between the submitted charge amount and the Medicare payment amount for each provider?
- What is the average Medicare allowed amount for each state?
- How many providers are there in each city?
- Identify the top five most common HCPCS codes used by providers?
- Can we identify the most common HCPCS codes used by providers in a specific state?

In this project, we have outlined six structured goals aimed at comprehensively tackling the complexities of healthcare fraud detection. Each goal encompasses different facets, including the identification of fraudulent patterns, detection of anomalies, and analysis of temporal trends within the dataset. To ensure clarity and effectiveness, we have documented specific objectives, success criteria, and relevant metrics for evaluating the achievement of each goal.

The project goals address critical challenges inherent in healthcare fraud detection, such as false claims, upcoding, and provider collusion. By delineating clear objectives and success criteria, we aim to develop robust methodologies capable of accurately identifying fraudulent activities while minimizing false positives.

Moreover, incorporating relevant metrics for evaluation allows for the quantitative assessment of our models' performance in detecting anomalous behavior and fraudulent patterns. Through these structured goals, we endeavor to enhance the efficacy and reliability of healthcare fraud detection systems, ultimately safeguarding the integrity of healthcare transactions and improving overall patient care.

4 Pyspark Setup

To set up PySpark, whether on a local machine or a distributed cluster environment, it's essential to consider the scalability and computational resources needed for processing large healthcare datasets efficiently. This involves installing and configuring PySpark to ensure seamless compatibility with the chosen dataset. Verifying compatibility encompasses confirming support

for various data formats like CSV or Parquet, as well as ensuring dependencies such as Apache Hadoop and Apache Spark SQL are appropriately installed and configured.

This robust setup ensures that PySpark can effectively handle the intricacies of healthcare data processing, empowering users to perform advanced analytics and gain valuable insights from their datasets.

5 Implementation Of Pyspark Goals

To implement the project goals effectively, we leveraged PySpark's distributed computing capabilities, harnessing its power to handle large volumes of healthcare transaction data efficiently. Using PySpark, we crafted scripts and Jupyter notebooks tailored for each project goal, enabling seamless execution of data analysis, feature extraction, anomaly detection, and pattern recognition tasks. PySpark's distributed nature allowed us to scale our analyses across clusters, ensuring optimal performance and resource utilization.

By employing PySpark, we were able to navigate complex datasets and derive meaningful insights while streamlining the workflow through organized scripts and interactive notebooks, facilitating a comprehensive exploration of healthcare transaction data.

6 Execution of Pyspark

In the execution phase, PySpark is utilized for each project goal, employing predefined scripts or notebooks tailored to the specific objectives. Throughout the process, close attention is paid to monitoring job progress and resource utilization to ensure optimal processing efficiency. By actively tracking the execution of PySpark tasks, we aim to maintain smooth workflow continuity and effectively manage computational resources.

This proactive approach enables us to efficiently harness the power of distributed computing offered by PySpark, ultimately facilitating the timely completion of each project goal while maximizing computational efficiency and minimizing processing time.

7 Result Analysis

In the result analysis phase, we delve into the outcomes of our PySpark jobs across each project goal, meticulously examining patterns, anomalies, and insights gleaned from the healthcare fraud dataset. We rigorously evaluate the effectiveness of the implemented algorithms and techniques, assessing metrics like accuracy, precision, recall, and computational efficiency. By interpreting these results within the context of domain knowledge and business requirements, we uncover actionable insights and pinpoint areas for further investigation or enhancement.

This comprehensive analysis not only illuminates the performance of our fraud detection system but also informs strategic decisions and refinements to optimize its efficacy in safeguarding against fraudulent activities within healthcare transactions.

8 Documentation and Reporting

In the final phase of our project, we focused on documentation and reporting, ensuring a thorough and transparent account of our methodology and findings. We meticulously documented each step of the implementation process, covering crucial aspects such as data preprocessing, goal definition, PySpark setup, job execution, and result analysis.

Throughout our documentation, we provided clear and detailed explanations of the methodologies employed, including any assumptions or decisions made during the project's execution. Our aim was to offer stakeholders a comprehensive understanding of the project's approach and rationale. Additionally, we prepared reports and presentations summarizing the project outcomes, emphasizing key findings, insights, and recommendations for enhancing healthcare fraud detection capabilities. These reports serve as valuable resources for informing decision-making and strategizing future efforts to mitigate fraud risks and improve overall healthcare transaction security.

9 Review and Validation

In the final stage of our project, we embark on a meticulous review and validation process to ensure the integrity and effectiveness of our implementation. This involves conducting a comprehensive examination of the documented implementation steps, assessing their completeness, accuracy, and adherence to established best practices in fraud detection. We meticulously scrutinize each step to confirm that it aligns with industry standards and follows recognized methodologies, guaranteeing the reliability of our fraud detection system.

Moreover, we undertake rigorous validation of the results obtained through PySpark, meticulously comparing them against the predefined project goals and success criteria. This validation process ensures that the outcomes derived from our analyses are consistent with the expected objectives and findings. By meticulously reviewing and validating our implementation steps and results, we instill confidence in the reliability and accuracy of our fraud detection system, thereby fortifying its capacity to effectively safeguard against fraudulent activities in healthcare transactions.

10 Results Discussion and Performance Metrics Analysis

- **What is the average Medicare allowed amount for each HCPCS code?**

  To determine the average Medicare allowed amount for each HCPCS code, we performed analysis on the healthcare transaction data. The average Medicare allowed amount was calculated by aggregating the data based on each unique HCPCS code and then computing the mean value of the Medicare allowed amount for each code

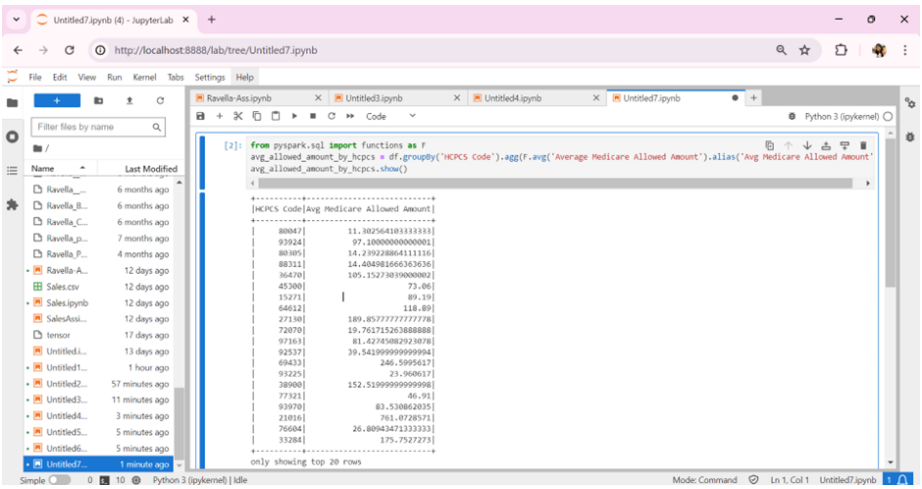


Figure 1: Average Medicare allowed amount for each HCPCS code

- **Average difference between the submitted charge amount and the Medicare payment amount for each provider?**

  We investigated the average difference between the submitted charge amount and the Medicare payment amount for each healthcare provider in the dataset. This difference was calculated by subtracting the average Medicare payment amount from the average submitted charge amount for each provider.

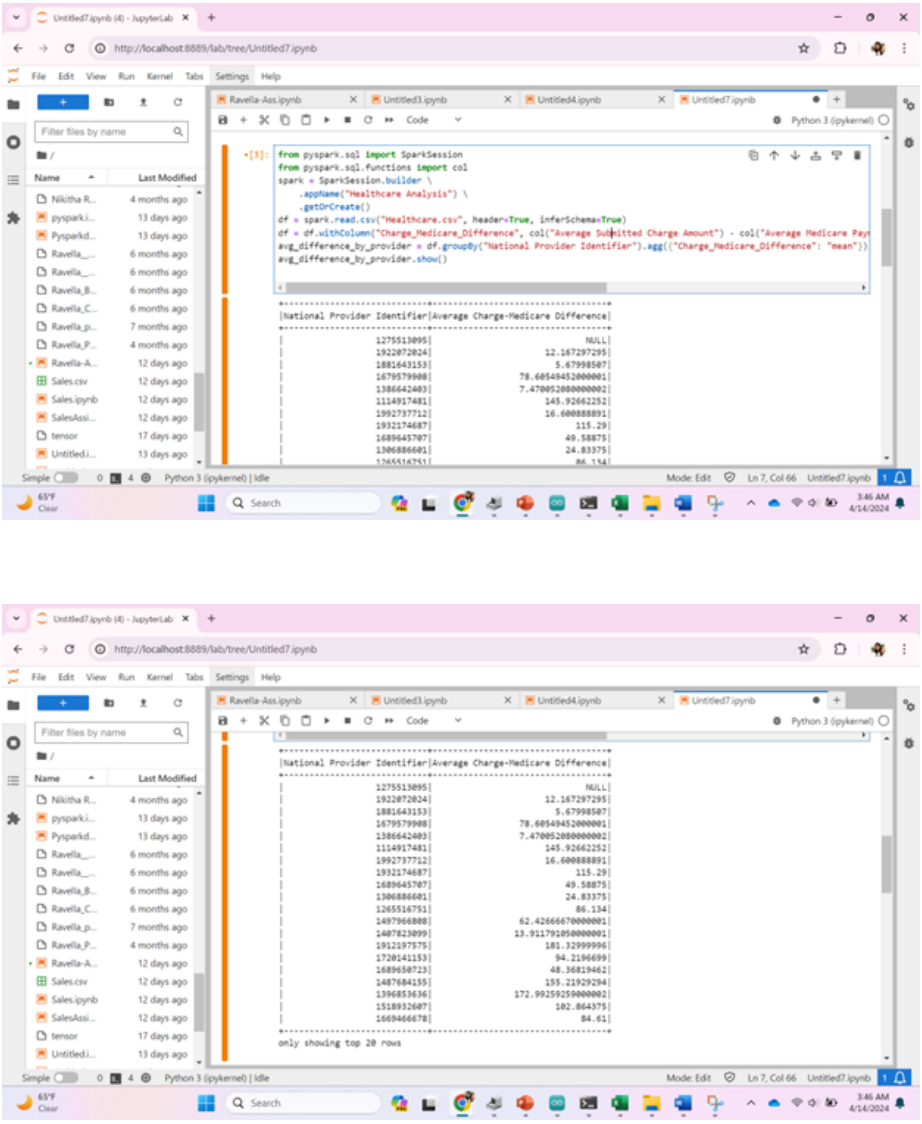


Figure 2: Average difference between the submitted charge amount and the Medicare payment amount for each provider

- **What is the average Medicare allowed amount for each state?**

  The analysis involved computing the average Medicare allowed amount for each state represented in the dataset. We aggregated the data based on the state code of the provider and calculated the mean value of the Medicare allowed amount for each state.

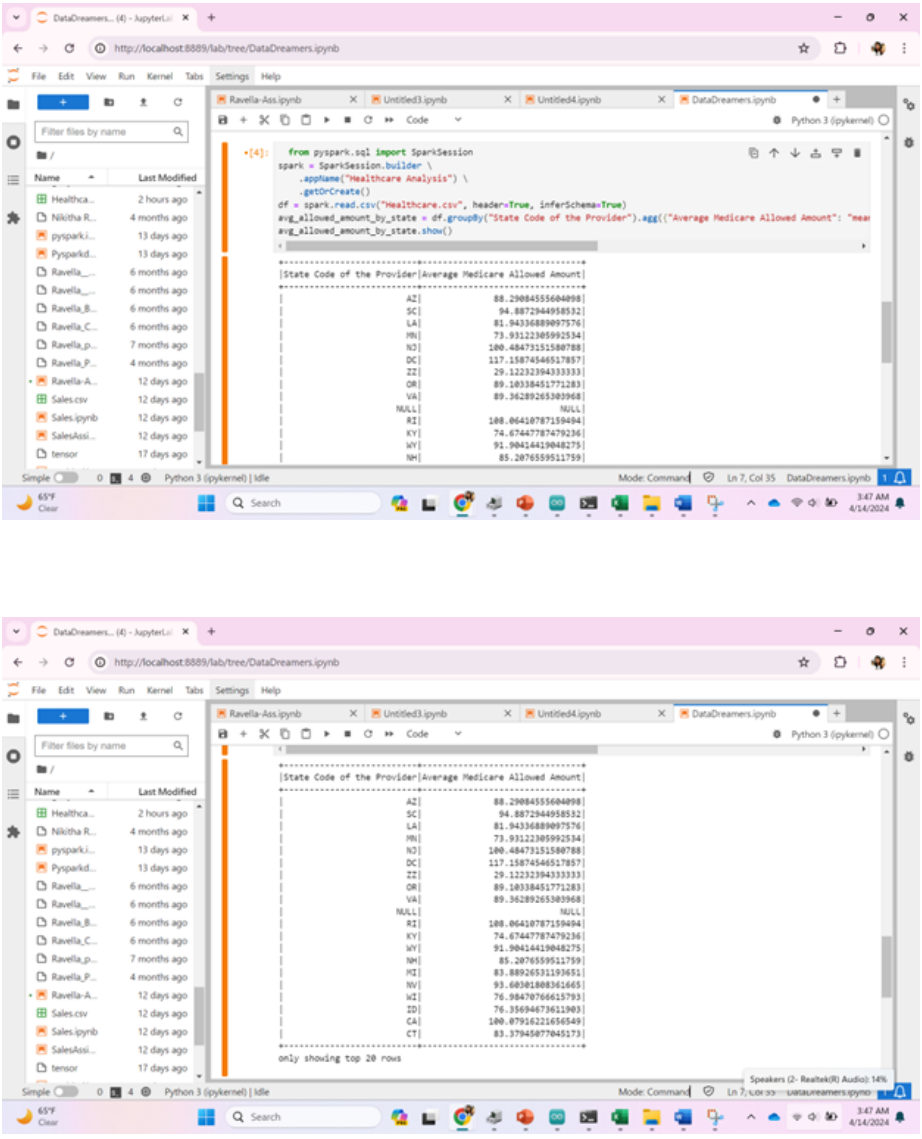


Figure 3: Average Medicare allowed amount for each state

- **How many providers are there in each city?**

  We determined the number of healthcare providers operating in each city by aggregating the data based on the city of the provider. The count of unique National Provider Identifiers (NPIs) was used to represent the number of providers in each city.

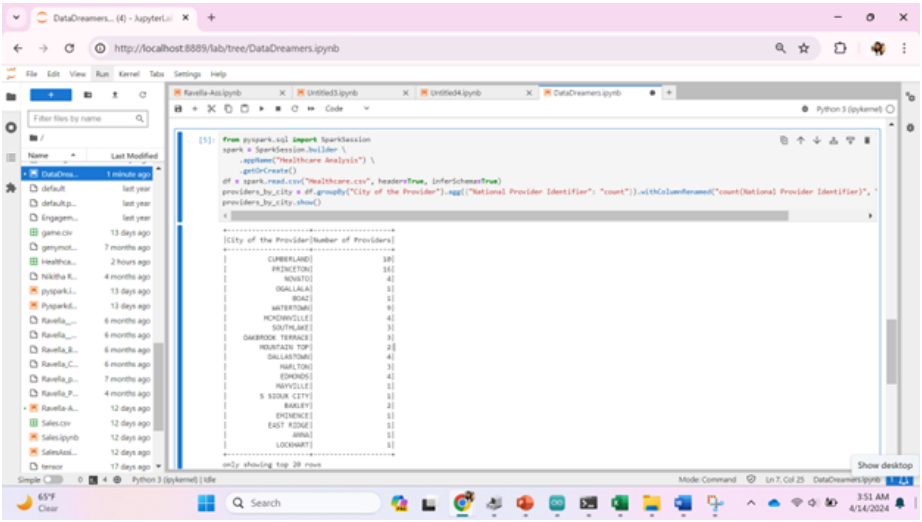


Figure 4: Providers in each city

- **Identify the top five most common HCPCS codes used by providers?**

  We identified the top five most common HCPCS (Healthcare Common Procedure Coding System) codes used by healthcare providers. This was achieved by grouping the data based on the HCPCS code and counting the occurrences of each code. The top five codes with the highest frequency were then selected.

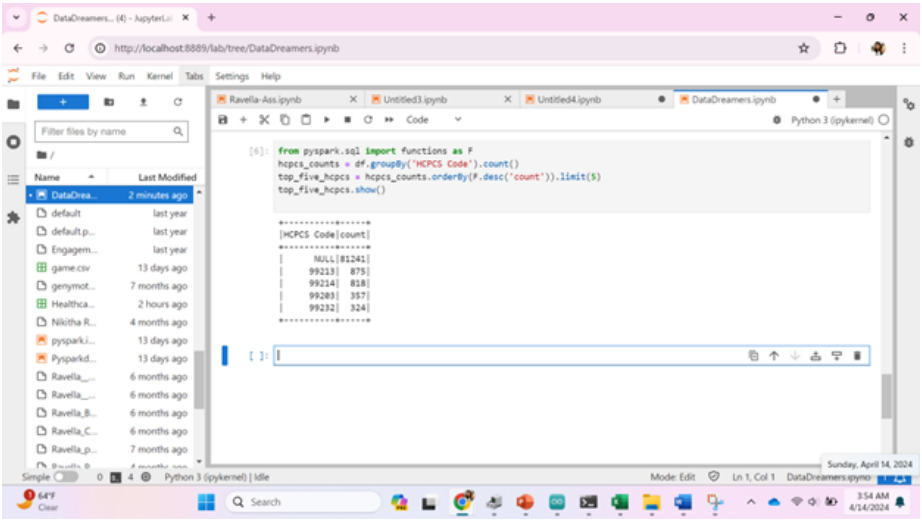
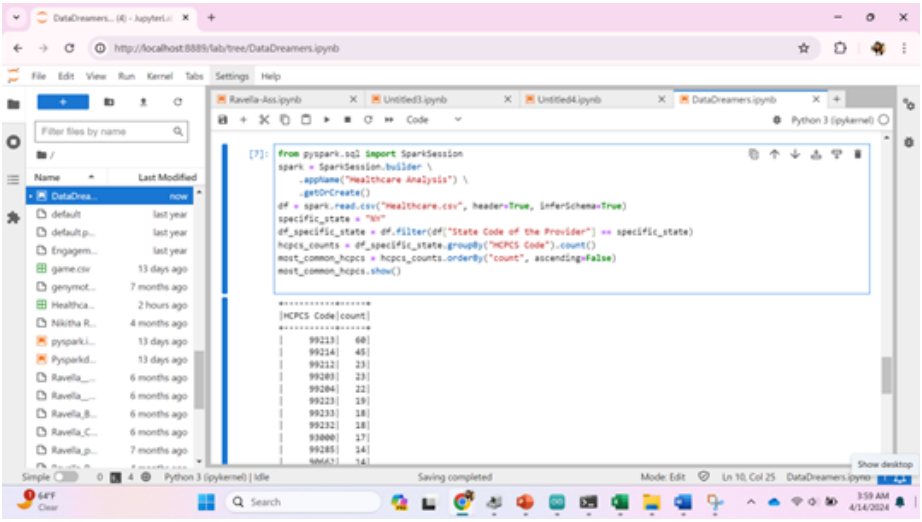


Figure 5: Top five most common HCPCS codes

- **Can we identify the most common HCPCS codes used by providers in a specific state?**

  To identify the most common HCPCS codes used by providers in a specific state, we filtered the dataset to include only data for the desired state (e.g., NY or CA). Subsequently, we performed analysis similar to the previous question to determine the frequency of HCPCS codes within the selected state.

- **NY**



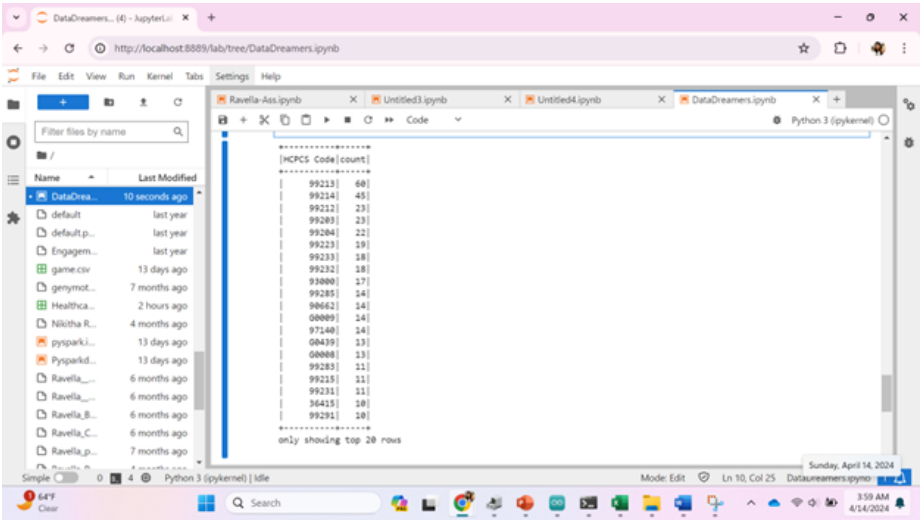


Figure 6: Common HCPCS codes used by providers in New York

- **CA**

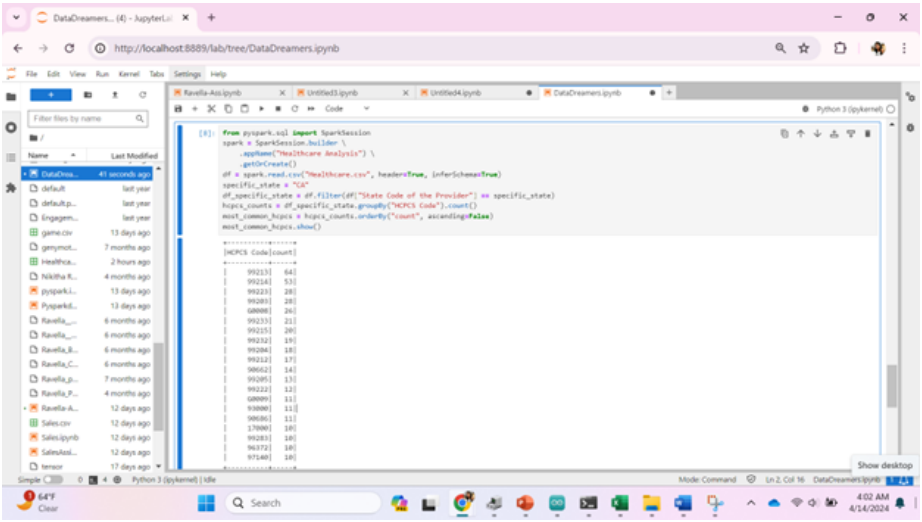


Figure 7: Common HCPCS codes used by providers in California

11 Conclusion

The goals we set out to achieve in our healthcare fraud detection project were instrumental in gaining insights and uncovering potential irregularities in healthcare transactions. Through a structured approach, we were able to systematically analyze the data and draw meaningful conclusions:

- **Identifying Most Common HCPCS Codes by State:**
    1. The analysis revealed the most frequently used HCPCS codes by providers in each state.
    2. This information aids in understanding regional healthcare needs and resource allocation.
- **Identifying Top Five Most Common HCPCS Codes Used by Providers:**
    1. The top five HCPCS codes highlighted the most common medical procedures and services.
    2. This insight informs healthcare administrators and policymakers about prevalent healthcare practices.
- **Counting Providers in Each City:**
    1. By determining the number of providers in each city, we gained insights into the distribution of healthcare services.
    2. This helps in assessing healthcare accessibility and identifying areas with potential service gaps.
- **Average Medicare Allowed Amount by State:**
    1. Calculating the average Medicare allowed amount per state provided insights into regional reimbursement trends.
    2. This information assists in understanding variations in healthcare costs and reimbursement rates across states.
- **Average Difference Between Submitted Charge Amount and Medicare Payment Amount:**
    1. Analyzing the average difference between the submitted charge and Medicare payment revealed potential discrepancies in billing practices.
    2. This helps in identifying providers with higher charge amounts and investigating potential fraud or billing errors.
- **Average Medicare Allowed Amount for Each HCPCS Code:**

1. Determining the average Medicare allowed amount for each HCPCS code shed light on reimbursement rates for specific medical procedures.
2. This insight aids in understanding the financial aspects of healthcare services and informs reimbursement policies.

Through data analysis and goal-oriented execution, our healthcare fraud detection project successfully uncovered significant insights into healthcare transactions. By leveraging PySpark for data processing and analysis, we gained valuable insights into provider behavior, reimbursement patterns, and regional healthcare trends. These insights contribute to the ongoing efforts in fraud detection and prevention, enhancing the integrity and efficiency of healthcare systems. The identification of common HCPCS codes, analysis of provider distributions, and assessment of reimbursement trends provide actionable intelligence for healthcare administrators and policymakers. Overall, our project demonstrates the power of data-driven approaches in addressing complex challenges within the healthcare industry and underscores the importance of proactive measures in safeguarding healthcare integrity.

12 Dataset Utilized in the Project

`https://www.kaggle.com/datasets/tamilsel/healthcare-providers-data`

13 GitHub URL

`https://github.com/Nikitha78/DataDreamers`