

Extracción automatizada de costos de servicios profesionales en facturas

Pedro Memoli Buffa

Introducción

En este informe se presenta un sistema que soluciona el problema de procesar costos de servicios profesionales en miles de facturas para la empresa Parlisur. Se adjunta al final sugerencias para los ejecutivos de como podría utilizarse el sistema y mejorar su rendimiento.

El trabajo consistió en extraer de un conjunto de 8411 facturas (pdfs en formato imagen y texto) los gastos totales de diversos servicios profesionales, donde la métrica de error fue el error absoluto medio. Para esto se utilizaron técnicas de OCR (optical character recognition), NLP (natural language processing), modelo lineal y test de hipótesis; creando así un sistema robusto, escalable y automático que es capaz de obtener costos profesionales con mucha precisión.

Implementación de la solución

Antes de describir el sistema de procesamiento de facturas, se presenta una motivación de como se lo fue construyendo.

Motivaciones y Desafíos

Para las facturas en formato imagen, *a priori* el problema parecía solucionarse aplicando un algoritmo de OCR sobre cada factura para pasarlas a texto, y a partir de ahí construir expresiones regulares que extraigan el costo profesional. El problema es que estas facturas en formato imagen muchas veces presentaban irregularidades que hacían que algoritmos de OCR invariablemente arrastraran errores insalvables. Un porcentaje no despreciable de facturas presento los siguientes problemas:

- Líneas blancas bloqueando el número de factura, costo total o costos profesionales.
- Mala calidad de imagen.
- Escrituras por arriba.

Resultando en que a veces se pierda una combinación de datos de numero de factura, costo total o costo profesional. En estos casos la única solución era *predecir* el resultado.

Por suerte, las facturas en formato texto eran sumamente consistentes y no presentaron los problemas de las imágenes. De estas pudo extraerse información con mucha precisión utilizando expresiones regulares. A partir de esos resultados confiables se pudo estimar una *predicción* del costo en servicios profesionales.

Como el objetivo es reducir el error absoluto medio, la mejor predicción que puede hacerse sin más información es la **mediana** del costo profesional. Pero si se tienen otras *variables* -como el costo total de la factura-, el mejor predictor pasa a ser la **mediana condicionada**¹; que es la mediana pero asumiendo que otra variable toma un valor.

¹Introduction to the Theory of Statistics"by Alexander M. Mood, Franklin A. Graybill, Duane C. Boes.

De los resultados de las facturas de texto se estimó la mediana de la distribución del costo profesional. Además, -bajo el razonable supuesto de que a mayor gasto total, más esperable es que el gasto profesional aumente- se estimó la mediana condicionada por el costo **total** ajustando un modelo lineal de regresión por cuantil. Este simplemente ajusta una recta sobre los datos con el cuidado de que minimice el *eam* de la muestra ².

El problema final fue el de detectar cuando el parseo del texto proveniente del algoritmo de OCR fallaba y ameritaba predecir. Para esto se separo en casos dependiendo de cuanta informacion se pudo extraer.

Cuando ambos datos de **costo total** y **costo profesional** estaban presentes, se implementó un test de hipótesis sobre el cociente

$$\frac{\text{CostoProfesional}}{\text{CostoTotal}}$$

cuya distribución se estimó a partir de los resultados para facturas en formato texto. Si el test retorno un p-valor menor a un determinado *threshold* -que indicaba un resultado sumamente improbable- se decidió rechazar el dato y estimar con el modelo lineal. En caso contrario se confió en el dato proveniente del parseo.

Cuando no estaba presente el costo profesional -o dio sospechosamente bajo-, se estimó con la mediana. Resultando así en un sistema robusto y escalable, incluso si muchas facturas son fotos totalmente ilegibles.

Diseño

A continuación se presenta la estructura general del proyecto. El sistema diseñado toma dos carpetas de facturas -en formato imagen y texto respectivamente- y retorna un csv con dos columnas *Invoice Number* y *Total Charged*; asociando cada factura a un costo total de servicios profesionales. El sistema se dividió en 5 sub-sistemas basándose en la técnica usada y el formato de la factura:

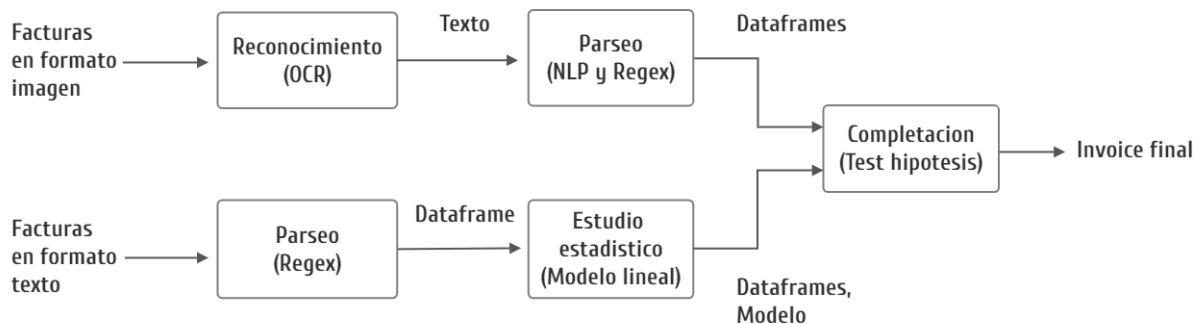


Figura 0.1: Pipeline de la conversión de facturas en formato imagen y texto al *invoice final*. Cada bloque representa un sistema que no tiene dependencias del resto.

- Las facturas en formato texto son el input del subsistema de *parseo* que se caracteriza por solo usar expresiones regulares. Este retorna un dataframe con columnas de número de factura, costo total y costo profesional. El resultado ingresa al *estudio estadístico*. De ahí se ajusta un modelo lineal usando regresión por cuantil y se extrae la mediana del costo profesional.
- Las facturas en formato imagen son el input de un modulo de *reconocimiento* de caracteres. Este retorna un array de strings -cada uno asociado a una factura distinta- que ingresa en otro subsistema de *parseo* cuyo output es un dataframe con columnas de número de factura, costo total y costo profesional. El parseo del texto es considerablemente más complejo que el de las facturas de texto. Además de expresiones regulares, usa ciertas técnicas de NLP.

²<https://d-nb.info/1209741377/34>


- Los outputs de los sistemas anteriores ingresan en un ultimo modulo. Este ejecuta un test de hipótesis entre otros *filtros* sobre los dataframes de las facturas en formato imagen. A partir del resultado del test se decide si el dato es poco confiable y si amerita predecir el costo profesional utilizando el análisis estadístico. Finalmente se completa y retorna un csv provisto por la empresa con todos los números de factura asociados a un costo en servicios profesionales.

Se desarrolla a continuación la implementación y experimentación de cada modulo.

Parseo de facturas en formato texto

Las facturas en formato de texto consisten en 2118 pdfs que se caracterizan por tener un peso menor a 60kb. Estas se almacenan en una carpeta de la cual el sistema de parseo extrae cada factura. El parseo de los pdf se lleva a cabo primero extrayendo el texto *crudo* con la librería PyPDF2 ³:

Designer - Danielle Reese	2	\$85.04	\$170.08
Amount Overdue - 358374817	1	\$249.06	\$249.06
Travel Expense from 2022-05-17	1	\$466.99	\$466.99
Individual/Group Meal on 2022-06-01	1	\$148.48	\$148.48
Project Engineer - Nicole Mason	3	\$176.13	\$528.39
	0	\$0.00	\$0.00
Amount Overdue - C71149052	1	\$176.13	\$176.13
Project Call - Equipment part	1	\$822.22	\$822.22
Administrative I - Mark Watkins	0.50	\$85.00	\$42.50
	0	\$0.00	\$0.00
Amount Overdue - 358374817	1	\$1,646.92	\$1,646.92



```

Designer - Danielle Reese
2
$85.04
$170.08
Amount Overdue - 358374817
1
$249.06
$249.06
Travel Expense from 2022-05-17
1
$466.99
$466.99
Individual/Group Meal on 2022-06-01
1
$148.48
$148.48
Project Engineer - Nicole Mason
3
$176.13
$528.39
0
$0.00
$0.00
Amount Overdue - C71149052
1
$176.13
$176.13
Project Call - Equipment part
1
$822.22
$822.22
Administrative I - Mark Watkins
0.50
$85.00
$42.50
0
$0.00
$0.00
Amount Overdue - 358374817
1
$1,646.92
$1,646.92

```

Figura 0.2: Conversión de un pdf en formato texto a un string con la librería Py2PDF

Cada fila de la factura se traduce a texto como *item, hora, costo por hora y costo total* separados por un newline character. El formato es siempre igual para **todas** las facturas, por lo que bastan expresiones regulares para detectar toda la información.

Del texto principalmente se extraen 3 datos: Número de factura, costo total y costo profesional. Cada extracción se separa en su propio *modulo*. El costo total y el número de factura son el resultado de algunas expresiones regulares, mientras que para obtener el costo profesional, el parseo es ligeramente más complejo.

Para ver si una fila se corresponde a un servicio profesional, el sistema evalúa si cumple alguna de estas tres condiciones:

- Contiene una profesión en una lista de *keywords* como *Engineer, Designer, Principal, etc.*
- Contiene alguno de los 850 apellidos mas comunes en los estados unidos
- La hora de trabajo es distinta a 1

Con el fin de reducir los falsos positivos, se incluyo otra serie de keywords -*Client, Expense, Amount Overdue, etc.*- que se asocian exclusivamente a costos no profesionales. Si la fila contiene alguna de esas palabras se ignora, incluso si cumplió alguna de las tres condiciones previas.

Esos tres datos -invoice, costo total y costo profesional- se obtienen para cada factura y se almacenan en un dataframe como csv. Algunos de los resultados testeados manualmente son los siguientes:

³<https://pypdf2.readthedocs.io/en/3.0.0/>

Factura	Invoice	Costo Profesional	Costo Total	Eam
102819958.pdf	102819958	5298.33	24585.89	0
105002622.pdf	105002622	24556.53	27216.52	0
109849880.pdf	139537191	8848.72	18641.91	0
187085275.pdf	187085275	21366.48	25385.16	0
201573028.pdf	201573028	12331.03	28612.2	0

Que junto con otros 20 resultados evaluados a mano indican que el sistema funciona correctamente. El dataframe se alimenta al sistema de *análisis estadístico*.

Análisis estadístico

El análisis estadístico tiene como objetivo construir buenas predicciones para cuando el parseo de facturas en formato imagen falle. Para esto recibe los datos del sistema anterior. Obtiene la mediana del costo profesional y ajusta una regresión por cuantil que modele una relación lineal entre el costo profesional y el costo total. Usa las librerías de python sklearn⁴ y numpy⁵.

La mediana de costo profesional es de 9591,8, que utilizada para predecir el costo de las facturas retorna un *eam* de 7739,4

Para ajustar el modelo se usa la clase de sklearn QuantileRegression⁶ que toma una lista de variables a predecir -en nuestro caso el costo profesional- y otra lista de variables que aportan información -costo total- para ajustar la regresión. El resultado fue el siguiente:

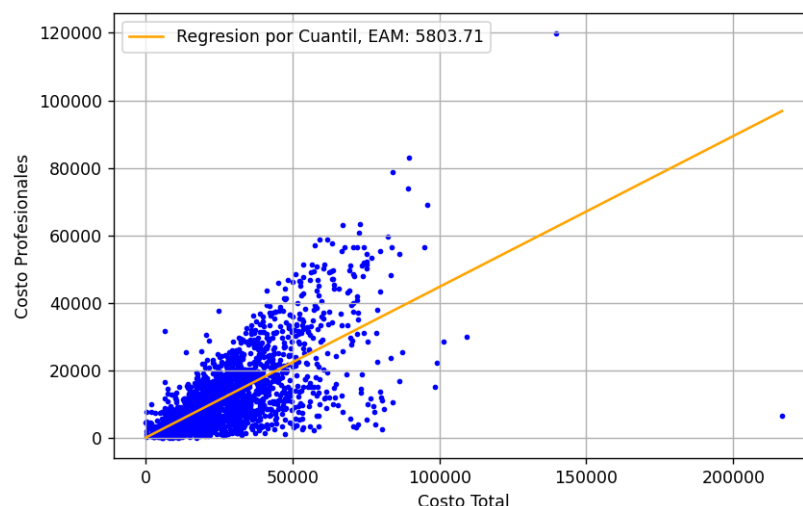


Figura 0.3: Regresión por cuantil del costo profesional basándose en el costo total.

Se observa una clara tendencia positiva. Además, el eam de predecir con la regresión es de 5803,7, que es menor al de usar la mediana; indicando así que realmente amerita usar el modelo lineal.

Este sistema acoplado al anterior completan el procesamiento de las facturas en formato texto. Los resultados se ingresan en el sistema de *completación* como muestra la figura 0.1. Antes de desarrollarlo se presenta el pipeline para el tratamiento de facturas en formato imagen.

Reconocimiento de caracteres en imágenes

La librería utilizada para aplicar algoritmos de OCR sobre las facturas que son imágenes es *PaddleOCR*⁷, ya que de todas que se probaron resultó ser la más precisa. Antes de hacer OCR, el

⁴<https://scikit-learn.org/stable/>

⁵<https://numpy.org/doc/stable/index.html>

⁶https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.QuantileRegressor.html

⁷<https://github.com/PaddlePaddle/PaddleOCR>

sistema aplica un pre-procesado a las imágenes. Este consiste en convertir el color a una escala de grises, y luego en usar una librería de *deskew*⁸ para enderezarlas.

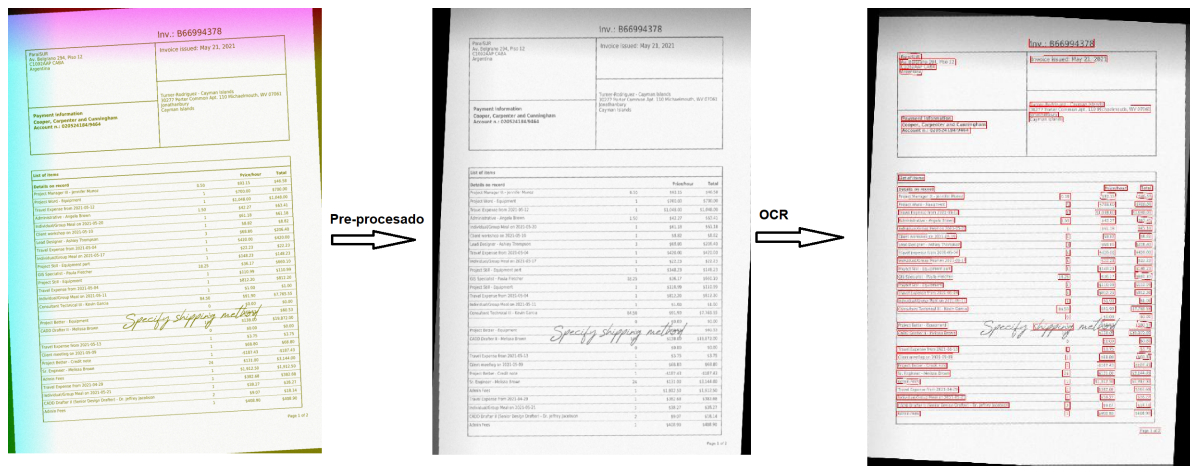


Figura 0.4: Secuencia de procesamiento sobre una factura.

Cada imagen se convierte con PaddleOCR en un array de tuplas que contiene las coordenadas del rectángulo en el que está cada secuencia de caracteres, junto con el texto dentro. Al igual que el texto resultante de Py2PDF (Figura 0.2), estas se ordenan naturalmente primero de izquierda a derecha y luego de arriba hacia abajo:

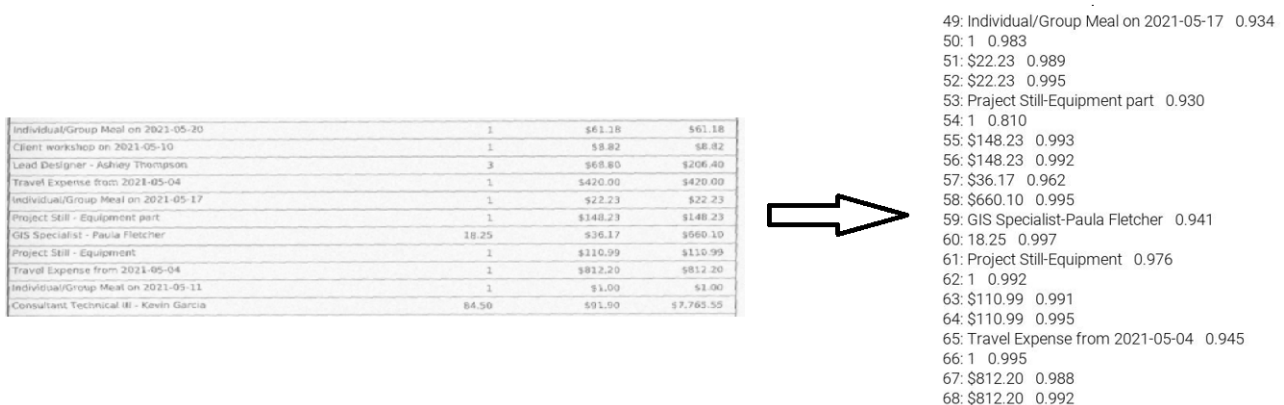


Figura 0.5: Esquema de la conversión de filas a texto.

Como se observa, algunas palabras se traducen con errores de ortografía (praject en vez de project) o algunas celdas se pierden (como el de GIS specialist). Todos esos errores hacen que el parseo del texto proveniente de OCR sea considerablemente más sofisticado que el del sistema anterior.

Una limitación del sistema es que con una GTX 1050 tarda aproximadamente 15 segundos en procesar cada imagen a texto. Como hay 6249 facturas que son fotos, el tiempo total de procesar las imágenes es de poco más de 26 horas. De cualquier forma, dada la naturaleza del problema y el esquema del proyecto, este reconocimiento solo es necesario correrlo una vez; por lo que el tiempo no es un problema.

Parseo de texto proveniente de OCR

Al igual que para las facturas en formato texto, del resultado proveniente de OCR se extraen tres cosas: Número de factura, costo total y costo profesional. El costo total y el número de factura se obtienen con el resultado de expresiones regulares, que por los errores de ortografía o cortes en la imagen son ligeramente más abarcativas -pero no tan distintas- del caso anterior.

⁸<https://pypi.org/project/deskew/>

El parseo del costo profesional también es similar, pero con el triple de expresiones regulares para evaluar todos los casos en que se pierde información de la fila (como por ejemplo cuando desaparece la hora).

Para ver si una fila se corresponde a un servicio profesional, el sistema evaluó si cumple alguna de estas tres condiciones:

- Contiene una profesión en una lista de *keywords* como *Engineer, Designer, Principal*
- Contiene alguno de los 850 apellidos mas comunes en los estados unidos
- La hora de trabajo es distinta a 1

Además se incluyo otra serie de keywords -*Client, Expense, Amount Overdue, etc.*- que se asocian exclusivamente a costos no profesionales, por lo que si contiene alguna de esas keywords ignora la fila incluso si cumplió alguna de las tres condiciones anteriores.

A diferencia del caso de parseo de facturas en formato texto, para evaluar si hay una profesión en una fila se implementó una función *contencionFuzzy* que usa la distancia de Levenshtein (usando la libreria *fuzzywuzzy*⁹) para evaluar si un string esta contenido en otro. Esto permite minimizar considerablemente el error por faltas de ortografía:

Profesión	Item	Contención normal	Contención Fuzzy
engineer	sr. sngenieer	False	True
administrative	edmenistrative	False	True
designer	Lead desgner	False	True

Los tres datos de *invoice*, *costo total* y *costo profesional* se obtienen para cada factura y se almacenan en un dataframe como csv. Algunos de los resultados testeados manualmente son los siguientes:

Factura	Invoice	Costo Profesional	Costo Total	Eam
10726698.pdf	10726698	5487.12	38885.33	0
116065291.pdf	116065291	4458.21	-1	0
2021-01-22.2952.pdf	915462705	0	10001.02	1276.82
2021-01-26.2143.pdf	937003582	35.48	54550.63	?
2021-01-27.1600.pdf	L51189888	9641.48	26558.37	0

Las facturas con eam distinto a 0 o ? en todos los casos testeados poseen errores insalvables, como por ejemplo una tira blanca tapando toda la columna de costos para cada item. Además, cuando el costo total o profesional no pudo ser encontrado, el resultado es -1 y 0 respectivamente. Esto se comprobó manualmente para otras 20 facturas.

Finalmente, se ingresa todo el procesamiento al sistema de completación.

Completación del dataframe final

Este sistema tiene el objetivo de llenar un csv provisto por la empresa, donde la primer columna tiene los números de factura y segunda los costos por servicios profesionales. A priori la segunda columna esta vacía. Le ingresan los dataframes de las facturas en formato texto e imagen junto con el modelo ajustado.

Lo primero que hace es llenar todos los costos para facturas del formato texto, ya que los resultados fueron de mucha confianza y no requieren pasar por un filtro.

Para procesar el dataframe de las facturas en formato imagen primero se construyo un filtro para decidir si cada dato era confiable o no. Como se explica en la motivación, para esto se

⁹<https://pypi.org/project/fuzzywuzzy/>

desarrolló un test de hipótesis basado en el cociente de costo profesional con el total. Se estimó la distribución con los datos de las facturas en formato texto según:

$$P\left(\frac{\text{costoprof}}{\text{costototal}} \leq t\right) = \frac{\#(\frac{\text{costoprof}}{\text{costototal}} \leq t)}{2118}$$

Y se definió el test de un nivel α arbitrario como

$$\delta_{\alpha} = 1 \left(\frac{\text{costoprofesional}}{\text{costototal}} < z_{\alpha} \right)$$

siendo z_{α} el cuantil α del cociente. El p-valor del test para un cociente c de costo total y profesional se calcula simplemente como $P(\frac{\text{costoprof}}{\text{costototal}} \leq c)$

El filtro separa en casos basándose en los datos que pudieron ser extraídos de cada factura:

- Si el costo total es -1 y el profesional no nulo: Se confía en el dato de costo profesional
- Si el costo total es -1 y el profesional nulo: Se estima con la mediana
- Si el costo profesional es menor a 30: Se estima con el modelo lineal
- Si falla el test de hipótesis (p-valor menor a 0.0001): Se estima con el modelo lineal

Luego de pasar el dato de una factura por el filtro, el sistema busca su número de invoice en el csv a llenar y completa la columna de costo. Finalmente, los invoices que no se lograron asociar a partir de los dataframes se completan con la *mediana* de costos de servicios profesionales (proveniente del análisis estadístico).

Ejecutando toda la solución se obtienen los siguientes resultados:

- Un 1.9 % de valores (156 de 8411) fueron sospechosos y fueron re-estimaron.
- Un 1.5 % números de factura (126 de 8411) no se lograron asociar.

El bajo porcentaje de errores indica los sistemas de parseo y de ocr son muy eficientes, incluso a pesar de que ciertos datos invariablemente se pierden. Cabe destacar que los valores que no pasaron el filtro no abarcan todos las facturas con eam no nulo, solo los que con total seguridad no se lograron parsear y realmente ameritaba predecir.

Conclusiones y Sugerencias

El sistema diseñado en este informe tiene la capacidad de procesar miles de facturas de la empresa Parlisur; obteniendo para cada una el *Invoice Number* y el *Costo en servicios profesionales*. Además, hace un análisis estadístico para las facturas en formato texto que le permiten ajustar un modelo para *predecir* costos en servicios basándose en el costo total. Combinado con un sub-sistema de filtrado para detectar cuando un escaneo tiene manchas o es de baja calidad, produce un mecanismo **escalable** y **robusto** para procesar todo tipo de facturas, incluso si muchas tienen problemas.

Sugerencias a ejecutivos

El carácter automático y escalable de la solución permite una variedad de aplicaciones. A continuación se presentan algunas de las cuales pueden ser de interés para figuras de autoridad de la empresa:

- Al **director ejecutivo** le podría interesar que se utilicen los datos extraídos para tener una visión más precisa de los gastos de la empresa y como se distribuyen por sector. El carácter automático de la solución permite que se pueda correr un monitoreo constante que proporcionaría una visión siempre actualizada de los gastos de la empresa.

- Al **director financiero** le podría ser útil que se realice un análisis de costo-beneficio de los servicios profesionales contratados. Se observan muchas facturas en las cuales el costo total y el profesional difieren por muy poco. Se podría indagar si esos servicios presentan algún patrón en su rentabilidad.
- Al **gerente de compras** se le presenta la posibilidad de utilizar el sistema para realizar auditorías constantes sobre cada proveedor. Podría evaluarse si, por ejemplo, una factura presenta un costo de servicio inusualmente alto que sea indicativo de sobreprecios.

Mejoras y alternativas

Como consultor creo que este sistema es la solución óptima al problema de procesar *este tipo* de facturas. No requiere mucho tiempo, es automático y puede escalarse a cualquier cantidad de datos. De cualquier forma, en mi opinión la mejor solución es conseguir las facturas en un formato digital como csv, excel, u otros. El tratado de esos archivos está muy estandarizado y haría del procesamiento algo simple, instantáneo y sin errores.

Se entiende de cualquier forma que lograr esto es difícil, ya que las facturas ingresan de distintos proveedores. Como mejoras provisionales a cambiar el sistema de facturas, hay dos recomendaciones que pueden optimizar mucho la precisión de los resultados:

- **Maximizar las facturas en formato texto:** Según el testeo, la solución es capaz de procesar todas estas facturas con un error prácticamente inexistente. Además permitiría entrenar modelos mas sofisticados para hacer predicciones.
- **Escanear con mejor tecnología:** Hoy en día existen aplicaciones como CamScanner que proveen escaneos gratis y que mejorarían mucho la precisión de los resultados. Evitando así por ejemplo las líneas verticales que tapan columnas enteras.

Si la empresa lograra convencer a los proveedores que entreguen las facturas en formato texto o que escaneen con mejor tecnología; la precisión del sistema sería aun mayor.