Análisis y clasificación de nombres por género

Parte 1

B) Clasificador de nombres por género

Método y herramientas

Para la construcción del clasificador utilicé R, con librerías de modelos de clasificación y manipulación de datos.
Para la manipulación / concatenación de los lotes de datos del dataset histórico invoqué un pequeño script en bash.

Resultados

Para la construcción del clasificador elegi Random Forest. El modelo consigue alta precisión sin necesidad de una preparación de datos previa exhaustiva. Dado que el problema no necesita alto grado de interpretación (donde Random Forest no se destaca), tampoco esto juega en contra al momento de seleccionar el método

La construcción del clasificador se dividió en 3 partes:

- manipulación inicial de los datasets
- selección de características de nombres con mayor significatividad, generación de nuevas variables y construcción del clasificador
- predicción sobre el dataset histórico

Manipulación inicial de los datasets

Datasets usados para la creación del clasificador:

- historico-nombres.csv: dataset histórico de nombres registrados desde 1922 - Ministerio de Modernización (http://datos.gob.ar/dataset/b8418d41-8e0c-4e85-8aa8-80d51a840132/resource/5e585bd4-bc7c-4ac6-b9a1-bcf44b85b28e/download/historico-nombres.zip)
- nombres-ref.csv: dataset de nombres permitidos por el RNP - Buenos Aires Ciudad (https://data.buenosaires.gob.ar/dataset/nombres-permitidos)

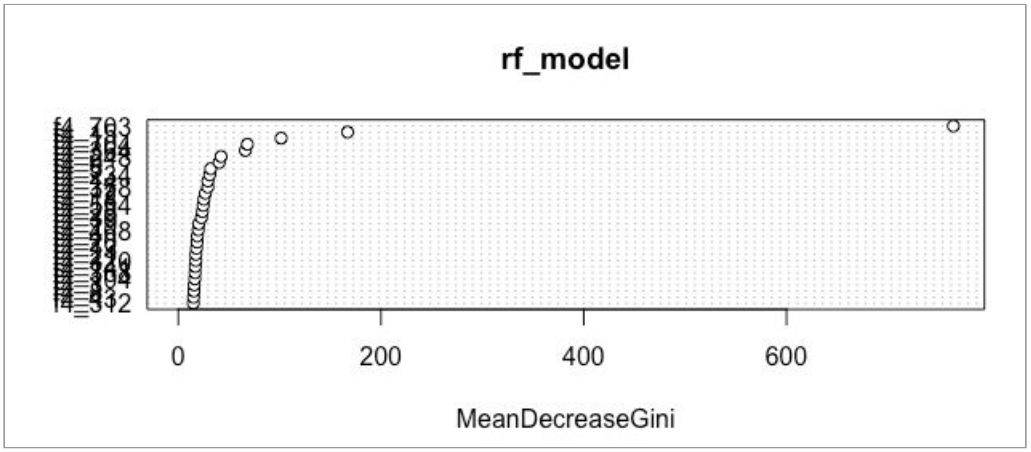
Ambos archivos fueron convertidos a minúsculas; para facilitar la manipulación y posterior matcheo de los datos. Se eliminaron duplicados del archivo histórico original, reduciendo la cantidad de registros en ~65%. Este es el archivo que utilicé posteriormente para la generación de nuevas variables y predicción.

Selección de características de nombres con mayor significatividad y generación de nuevas variables y construcción del clasificador

Para evaluar qué características de los nombres eran más significativas para la predicción del género, seleccioné 4 reglas:

- Regla 1: cantidad de cada caracter
- Regla 2: 2-grams
- Regla 3: caracter de final del nombre
- Regla 4: largo del nombre

Como primer paso dividí el dataset de referencia en train y test (75% / 25%), generando para el subset train todas las variables (total de variables: 704 = 26 + 26 * 26 + 2 - Ref: `char_count()`).
Para evaluar qué características eran más significativas usé la medida de importancia de las variables resultante del modelo Random Forest.



Ordenando los valores y seleccionando las 9 más importantes, el resultado fue el siguiente:

	a	f	m	MeanDecreaseAccuracy	MeanDecreaseGini
f4_703	10.67311876	56.1097250	59.4297973	62.099505	775.5391318
f4_15	16.76866401	24.7054696	22.2496196	29.343161	169.2492193
f4_1	4.74251409	20.1380168	18.8198900	25.563650	103.0045802
f4_364	1.93370667	15.2555702	18.1586270	23.190905	62.4079982
f4_9	0.21904224	14.2063249	17.1918928	20.988248	39.3558324
f4_378	5.12121931	14.1430328	14.2512900	19.289273	32.2956304
f4_248	6.63344947	15.0237601	13.1344969	19.267924	39.1014000
f4_234	7.18423617	9.2259836	15.4032782	18.947330	30.9399463
f4_704	7.32431196	8.5991722	12.9303478	18.286510	68.4565681

Con esta información construí entonces una segunda función (Ref: `char_count_tuned()`), ahora sí para la creación de las 9 variables con mayor contribución a la precisión del modelo.
Una vez generadas las variables y construído el modelo, lo corrí sobre el subset test, dando como resultado una precisión de 0.813%

```
Overall Statistics

               Accuracy : 0.813
                 95% CI : (0.797, 0.8283)
    No Information Rate : 0.5055
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6402
 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: a Class: f Class: m
Sensitivity          0.0000000   0.8004   0.9009
Specificity          0.9995723   0.8785   0.7586
Pos Pred Value       0.0000000   0.8418   0.7923
Neg Pred Value       0.9523227   0.8449   0.8822
Prevalence           0.0476578   0.4468   0.5055
Detection Rate       0.0000000   0.3576   0.4554
Detection Prevalence 0.0004073   0.4248   0.5747
Balanced Accuracy    0.4997861   0.8394   0.8298
```

Como último paso, generé las mismas variables en el dataset histórico (reducido luego de eliminar los duplicados) en lotes de 200.000 registros bajados a disco a archivos .csv.

Predicción sobre el dataset histórico

Con las nuevas variables generadas en el archivo histórico reducido, quedaba entonces concatenarlos y correr el clasificador.

Mediante un script ejecutado desde R, concatené los lotes del archivo histórico con las nuevas variables y ejecuté el modelo sobre el mismo para predecir el género de dichos nombres.
Una vez finalizado el proceso, el siguiente paso fue concatenar los valores y hacer un left join con el dataset histórico completo.
El resultado obtenido fue el siguiente:

```
summary(pers_hist_full_sexo$sexo)
     a       f       m
 15528 5789888 3956193
```

Conclusiones sobre el clasificador generado

Si bien segun la precisión obtenida en la predicción de test es alta, de antemano se detectaron errores mayormente en la clasificación de clases ‘a’; en este caso, si bien la cantidad de casos es significativamente baja, como refinamiento del modelo debería generarse una nueva regla que corrigiera la clasificación de estos casos.
Por otro lado, llama la atención la diferencia en cantidad de casos detectados como nombres de mujer vs nombres de hombre (‘f’ representan un 59%, ‘m’ apenas representan un 41%)