Statystyczna przewidywalność wyników konkursu Chopinowskiego: Czy możemy zastąpić godziny oczekiwania sekundami obliczeń?

Weronika Orzechowska

Spis treści

1 Wstęp

1.1 Motywacja

Konkurs Chopinowski za każdą swoją edycja przyciąga tysiące melomanów, którzy śledzą poczynania swoich ulubionych uczestników na żywo lub przez transmisje internetowe. Emocje kumulują się zawsze po ostatnim występie, kiedy nadchodzi czas przesądzenia dalszych losów pianistów w konkursie. Jury zbiera się na obrady i (zazwyczaj z dużym opóźnieniem), po wielu godzinach wyłania szczęśliwe grono pianistów przechodzących do następnego etapu.

Ale czy faktycznie musi to trwać tak długo? Czy informacja o tym, nie jest już zaszyta w wypełnionych tablekach z punktami, czy decyzja nie została już podjęta w ukryciu punktów i wskaźników?

1.2 Zbiór danych

Do przygotowania projektu wykorzystałam dane opublikowane przez Narodowy Instytut Chopina po 18. edycji konkursu chopinowskiego w 2021 roku: link. Są to punkty przyznawane przez jury w każdym etapie konkursu. Uczestniczy otrzymywali dwie wartości: 'y'/'n', czyli czy powinni pojawić się w kolejnym etapie oraz punkty przyznawane na podstawie wykonania.

Specyfika konkursów z dziedzin artystycznych sprawia, że nie zawsze punkty i czyste statystyki są wyznacznikiem tego, czy dana osoba znajduje powodzenie. Punkty traktowane są bardziej jako pomoc w podejmowaniu decyzji, a nie faktyczny wyznacznik sukcesu jednostki. Przynajmniej tak jest to przedstawiane przez jury konkursu po wielogodzinnych obradach i licznych opóźnieniach w ogłoszeniu wyników.

Dlatego też zdecydowałam się na wybór tych danych, aby zobaczyć na ile przyznawana punktacja przez jury przekłada się na faktyczną dynamikę kwalifikacji, które zawsze wzbudzają wiele emocji wśród melomanów.

Do dyspozycji miałam trzy zbiory danych, z każdego etapu konkursu. Znajdują się w nich następujące informacje: imię i nazwisko uczestnika, reprezentowana narodowość, ilość punktów oraz wskaźnik 'y' przyznany przez każdego uczestnika jury. Dodatkowo mamy dwie miary statystyczne - wyznaczony wskaźnik y oraz średnią liczbę punktów. Jednak w ramach analizy danych znalazłam sprzeczności w miarach wyznaczonych przez NIFC, dlatego nie uwzględniałam ich w analizie danych.

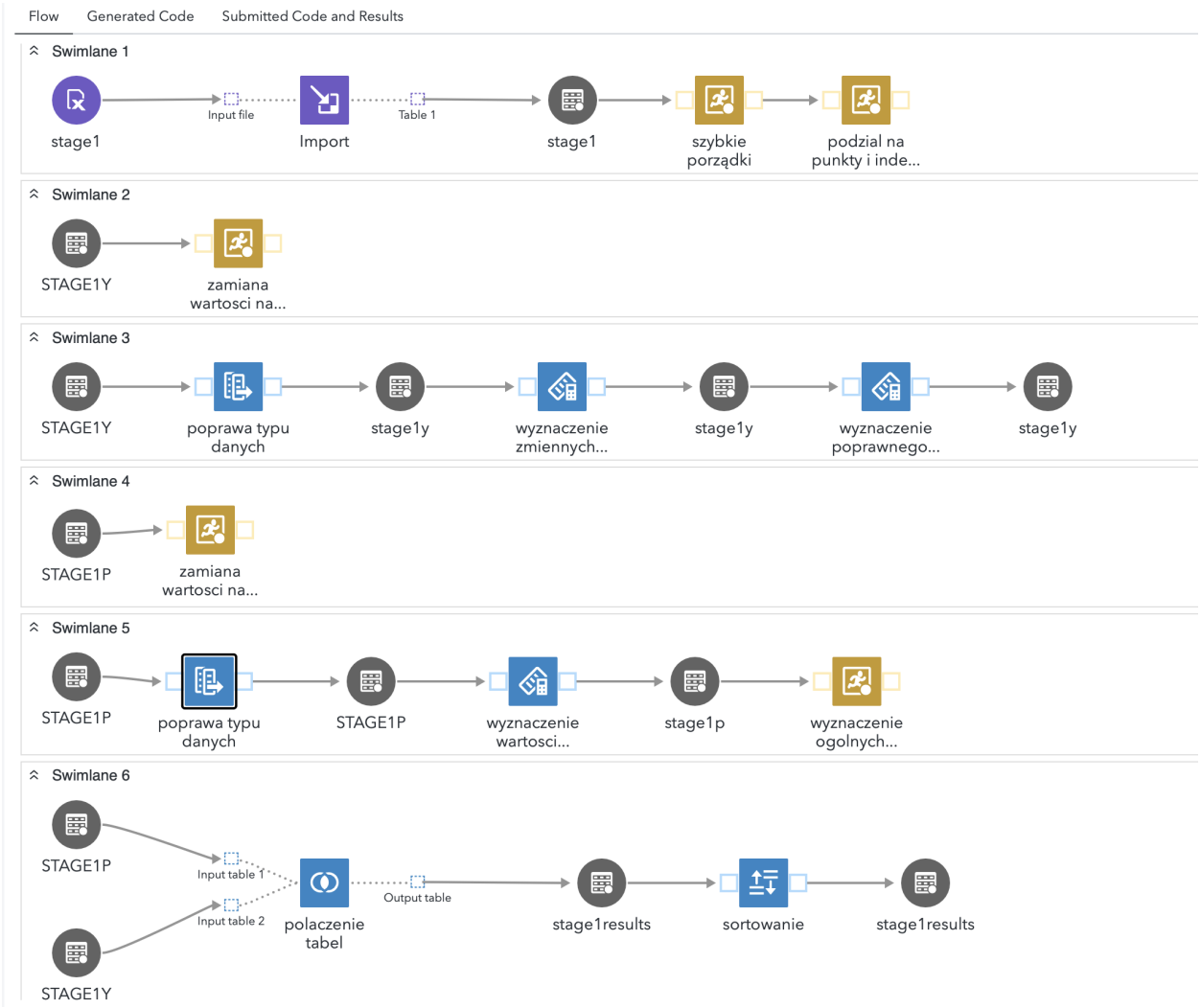
2 Analiza danych - Develop Code and Flows, Explore and Visualize

2.1 Wstępna obróbka danych

Analizę danych zaczęłam od początkowej obróbki danych. Dane w formacie .xlsx zostały zaimportowane do flow (rys. 1, swimlane 1). Dodatkowo poprawione zostały nazwy kolumn, ewentualne niepoprawne przeczytanie danych. Ze względu na specyfikę zbioru danych, do części eksploracyjnej zdecydowałam się na podział na punkty oraz indeks y. W ramach każdego etapu konkursu, każdy uczestnik otrzymuje od każdego członka jury dwie wartości - przydzielone punkty oraz wartość 'y'/'n' oznaczającą, czy powinien przejść do kolejnego etapu.

Analizę danych dla współczynnika y rozpoczęłam (rys. 1, swimlane 2,3) od zamiany typu tej zmiennej na zmienną binarną. 'y' miało przydzielone 1, 'n' 0. Wartości 's' oraz 'a' są uznawane za braki danych - członek jury był nieobecny lub nie brał udziału w ocenie danego uczestnika konkursu. W tym etapie wyznaczyłam też ponownie indeks y. W orginalnym zbiorze danych był on liczony bez uwzględnienia 'braków danych', co powoduje, że w wielu przypadkach jest on zaniżony.
Dla zbioru punktowego (rys. 1, swimlane 4,5) również wyróżniłam braki danych oraz wyznaczyłam przydatne miary statystyczne, które były wykorzystywane w dalszej analizie.

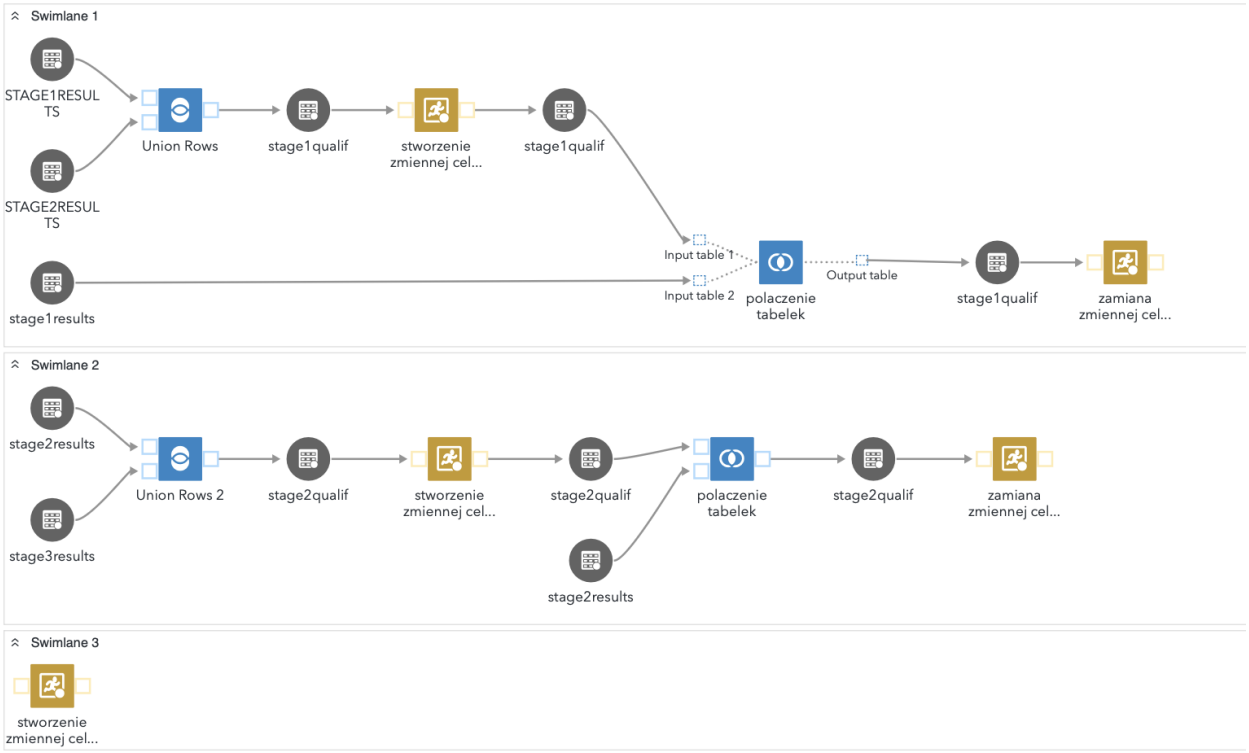
Tak przygotowane zbiory danych zostały następnie połączone (rys. 1, swimlane 6) zostawiając już w nich jedynie wyznaczone miary statystyczne. Na tej podstawie wyznaczyłam ranking najlepszych uczestników konkrsu dla każdego etapu. Ciekawym wnioskiem, wręcz rzucającym się w oczy jest, że zwycięzca konkursu zajmwał pierwsze miejsce w rankingu każdego etapu, istotnie górując nad innymi uczestnikami.



Rysunek 1:

W następnym kroku, dla każdego zbioru danych (dla każdego etapu) dodałam zmienną binarną qualif (rys. 2), decydującą czy dany uczestnik przeszedł dalej do konkursu. Zmienna ta była wykorzystana przy wizualizacji wyników oraz budowaniu modeli.

Ostatnim elementem przygotowania danych było przygotowanie zbiorów pod modele uczenia maszynowego. W tym celu połączone zostały zbiory ze wskaźnikiem y, punktami oraz zmienną celu qualif.



Rysunek 2:

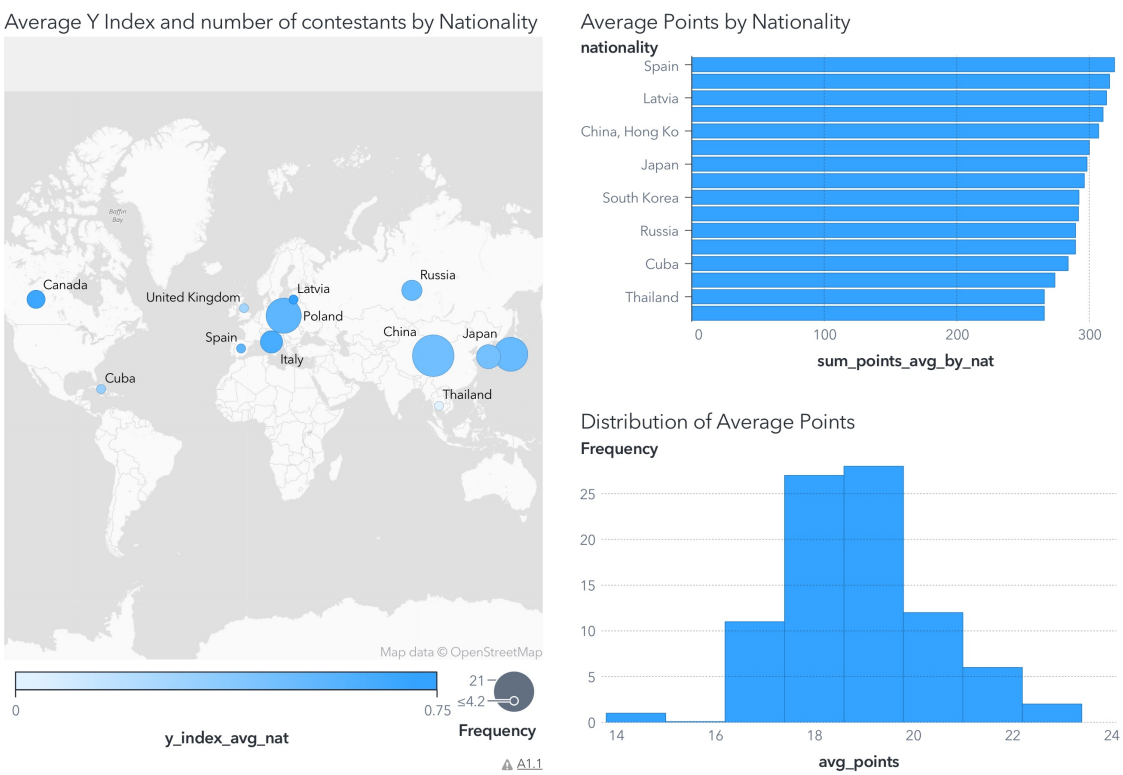
2.2 Analiza danych

W tej części wykorzystałam dodatkowy moduł w SAS Viya - Explore and Visualize, z którym zapoznałam się na podstawie darmowych kursów na SAS Skill Builder for students.

2.2.1 Etap pierwszy

Analiza wykresów (rys. 3, 4) przedstawia zróżnicowanie wyników i narodowości uczestników etapu pierwszego Konkursu Chopinowskiego 2021. Najliczniej reprezentowane są kraje azjatyckie, takie jak Japonia, Chiny i Korea Południowa, a także Rosja. Wartości wskaźnika mieszczą się w przedziale od 0 do 0,75, co wskazuje na zróżnicowanie poziomu wstępnego między narodowościami. To podkreśla dominację krajów z silnymi tradycjami edukacji muzycznej, szczególnie w Azji.
Histogram pokazuje rozkład średnich punktacji uczestników. Największa grupa uzyskała wyniki w przedziale 18–20 punktów, co świadczy o wyrównanym poziomie większości uczestników. Mniej liczne grupy to wykonawcy z punktacją poniżej 16 lub powyżej 22 punktów, co sugeruje obecność zarówno najsłabszych, jak i wyjątkowo wybitnych pianistów. Wykres słupkowy wskazuje, że najwyższą sumaryczną średnią punktację uzyskali uczestnicy z Japonii, Chin i Korei Południowej. Rosja oraz kraje europejskie, takie jak Łotwa i Hiszpania, również prezentują znaczące wyniki, choć na mniejszą skalę. Dane podkreślają dominację krajów azjatyckich i Rosji, co wynika z ich dużej liczby uczestników oraz wysokiego poziomu przygotowania.

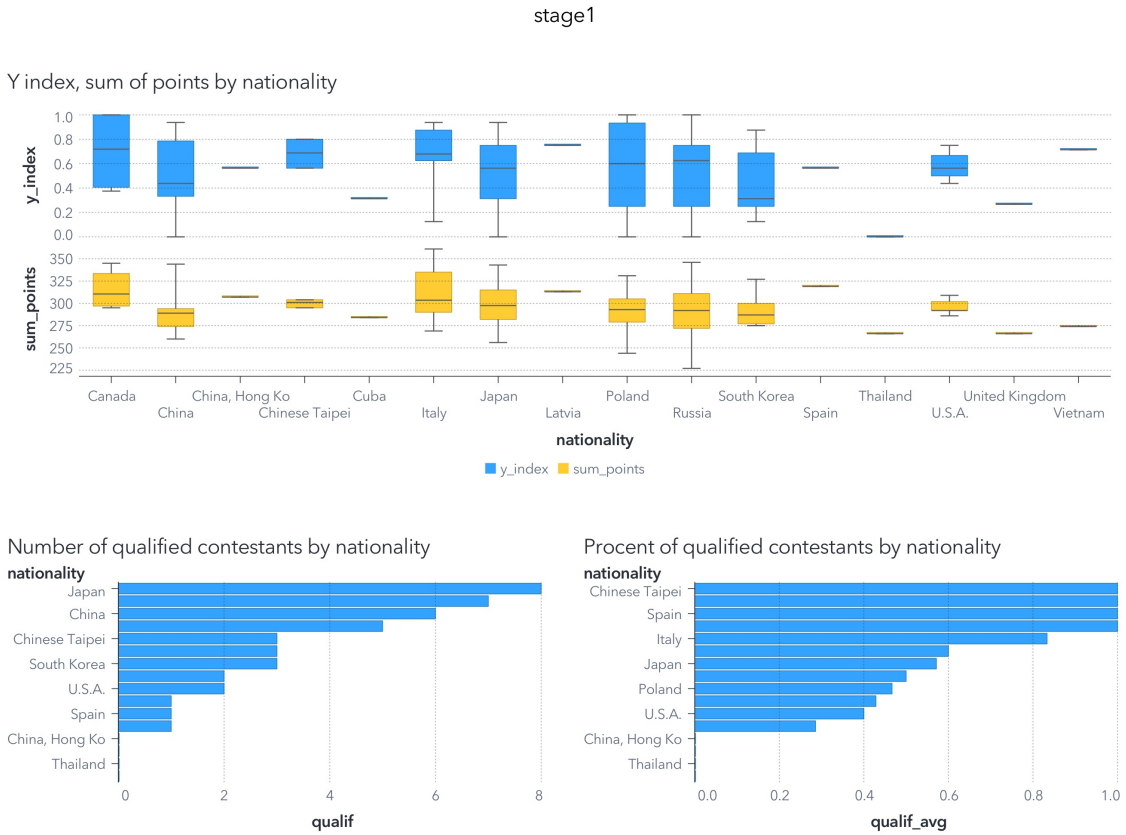
Wykres pudełkowy ilustruje średnie wartości wskaźnika y oraz sumy punktów dla poszczególnych krajów. Uczestnicy z Japonii, Polski, Chin i Korei Południowej osiągnęli zarówno wysoki wskaźnik y, jak i wysokie sumy punktów, co świadczy o ich wysokim poziomie artystycznym. Z kolei w przypadku Kanady i Włoch rozkład wskaźnika y jest bardziej zróżnicowany, co sugeruje różnorodny poziom wśród ich uczestników. Suma punktów uczestników w większości przypadków mieści się w przedziale od 250 do 350, z większą koncentracją wyższych wyników w krajach z rozwiniętą edukacją muzyczną.



Rysunek 3:

Wykres słupkowy ukazuje liczbę zakwalifikowanych uczestników z poszczególnych krajów. Japonia przoduje z największą liczbą pianistów przechodzących do kolejnego etapu, co potwierdza jej silną reprezentację. Chiny, Chińskie Tajpej oraz Korea Południowa również wyróżniają się wysoką liczbą zakwalifikowanych uczestników. Kraje takie jak Stany Zjednoczone i Hiszpania mają mniejszą liczbę pianistów w kolejnej rundzie, co może wynikać z większej selekcji.

Kolejny wykres przedstawia odsetek zakwalifikowanych uczestników w stosunku do całkowitej liczby reprezentantów danego kraju. Chińskie Tajpej oraz Hiszpania osiągają najwyższy procent zakwalifikowanych uczestników, co wskazuje na bardzo wysoki poziom artystyczny ich reprezentantów. Japonia, choć ma najwięcej zakwalifikowanych uczestników, odnotowuje nieco niższy procent w porównaniu z czołowymi krajami. Inne kraje, takie jak Polska i USA, osiągają umiarkowane wyniki, co wskazuje na stosunkowo zróżnicowany poziom artystyczny.



Rysunek 4:

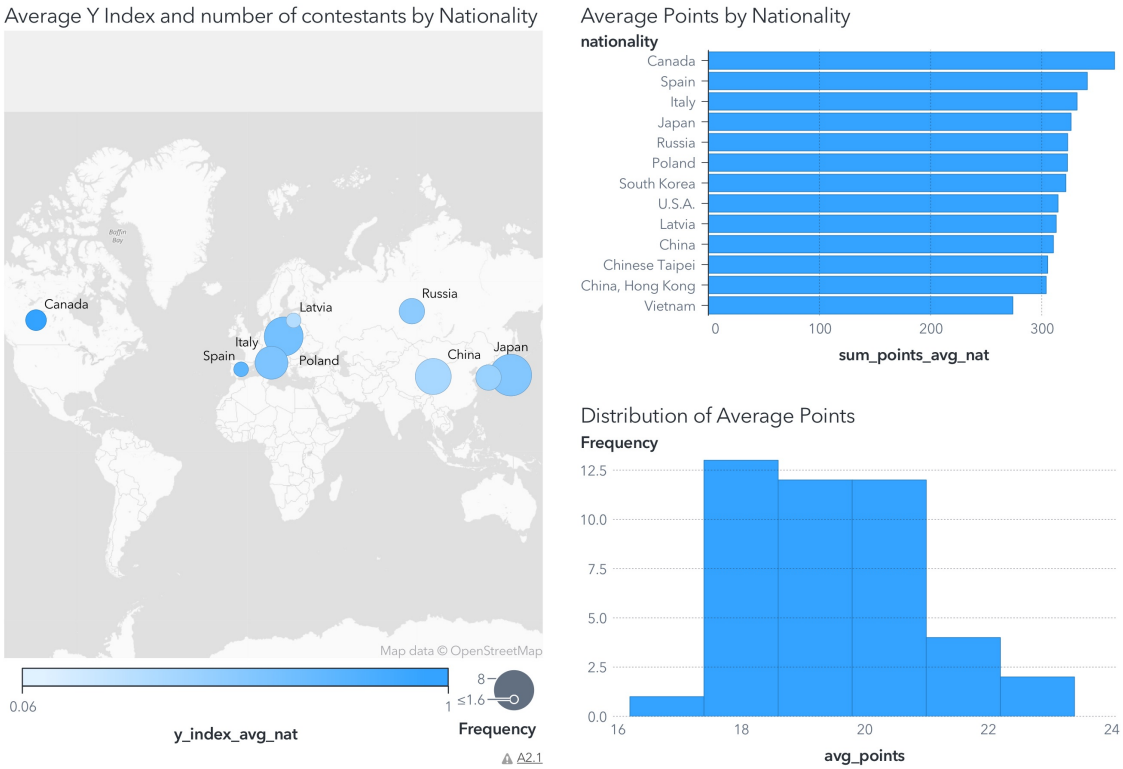
2.2.2 Etap drugi

W drugim etapie Konkursu Chopinowskiego (rys. 5,6) analiza wykresów pokazuje dalsze zróżnicowanie wyników i narodowości uczestników.

Mapa przedstawiająca średni wskaźnik y w podziale na kraje pokazuje, że Europa, w tym Polska, Włochy i Łotwa, ma znaczny udział w konkursie. Widoczne są również duże punkty reprezentujące Japonię, Koreę Południową i Kanadę, co świadczy o ich znaczącej liczbie uczestników w tym etapie. Wskaźnik y waha się od 0,06 do 1,6, co wskazuje na większe zróżnicowanie wyników w porównaniu z etapem pierwszym.
Histogram przedstawia rozkład średnich punktacji. Najwięcej uczestników osiąga wyniki w przedziale 18–20 punktów, co ponownie wskazuje na wysoki, ale wyrównany poziom wykonawczy. Uczestnicy z wynikami poniżej 18 punktów stanowią mniejszość, co sugeruje większą selekcję w tym etapie. Niewielka liczba pianistów uzyskuje wyniki powyżej 22 punktów, co wskazuje na ich wybitne umiejętności.
Na wykresie słupkowym najwyższą sumaryczną średnią punktację uzyskali uczestnicy z Kanady, Włoch i Hiszpanii, co może wynikać z jakości ich reprezentantów. Japonia, Rosja i Korea Południowa utrzymują wysoką pozycję z etapu pierwszego, podkreślając ich stałą obecność w gronie najlepszych. Wyniki te wskazują, że drugi etap jest bardziej selektywny, a dominacja krajów z rozwiniętą edukacją muzyczną nadal jest widoczna.

Wykres pudełkowy pokazuje średnie wartości wskaźnika y oraz sumy punktów dla różnych krajów. Uczestnicy z Polski i Rosji wyróżniają się wysokimi wartościami wskaźnika y oraz stosunkowo wysokimi sumami

stage2



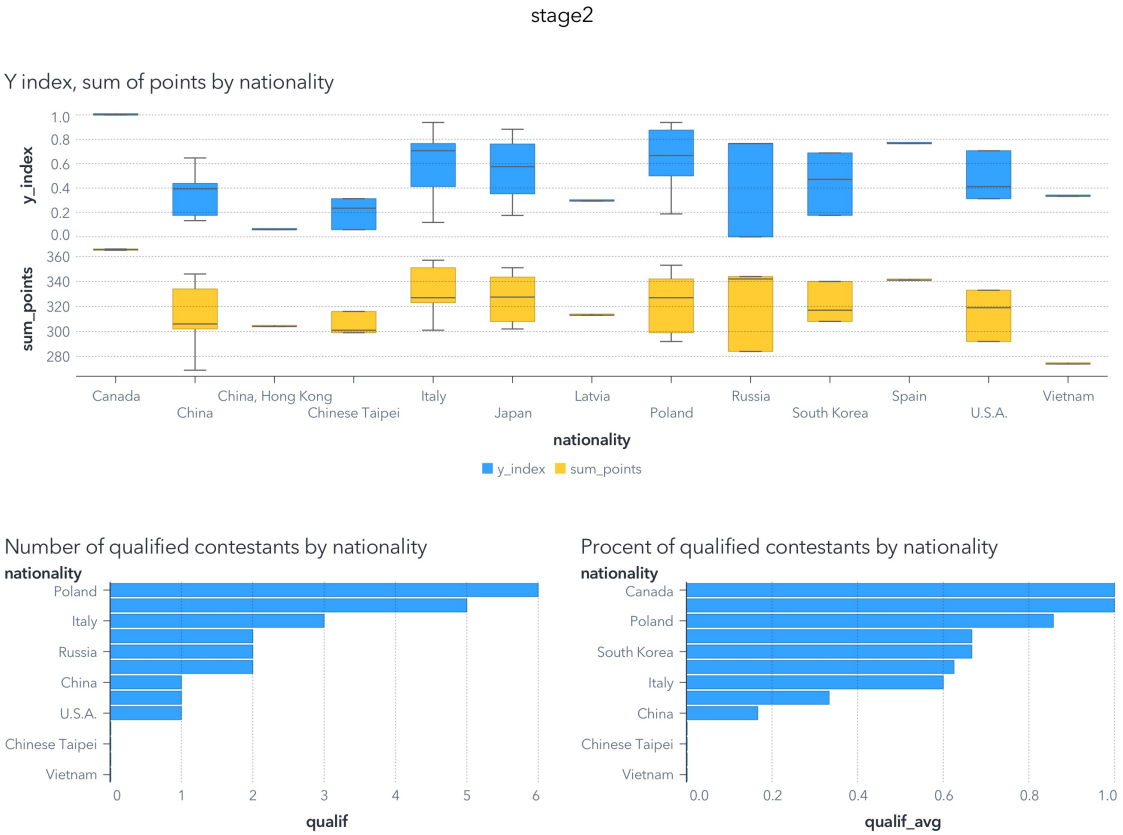
2

Rysunek 5:

punktów, co potwierdza ich wiodącą pozycję w konkursie. Japonia i Włochy również osiągają dobre wyniki, z mniejszym zróżnicowaniem w swoich danych, co wskazuje na wysoki i równomierny poziom reprezentantów. Z kolei kraje takie jak Łotwa, USA i Wietnam mają niższe wartości wskaźnika y i sum punktów, co może wskazywać na trudności w rywalizacji na tym etapie.

Wykres słupkowy dotyczący liczby zakwalifikowanych uczestników wskazuje, że Polska ma największą liczbę pianistów przechodzących do kolejnego etapu, co podkreśla siłę jej reprezentacji. Włochy i Rosja również mają znaczną liczbę zakwalifikowanych uczestników, co wskazuje na wysoki poziom ich artystów. Kraje takie jak USA i Chińskie Tajpej mają mniejszą liczbę zakwalifikowanych uczestników, jednak ich obecność w konkursie wciąż jest zauważalna.

Procentowy udział zakwalifikowanych uczestników w stosunku do całkowitej liczby reprezentantów pokazuje, że Kanada, Polska i Korea Południowa osiągają najwyższe wyniki, co wskazuje na bardzo wysoki poziom jakości ich pianistów. Włochy i Chiny również osiągają dobre wyniki w tym wskaźniku, co podkreśla ich silną pozycję w konkursie.

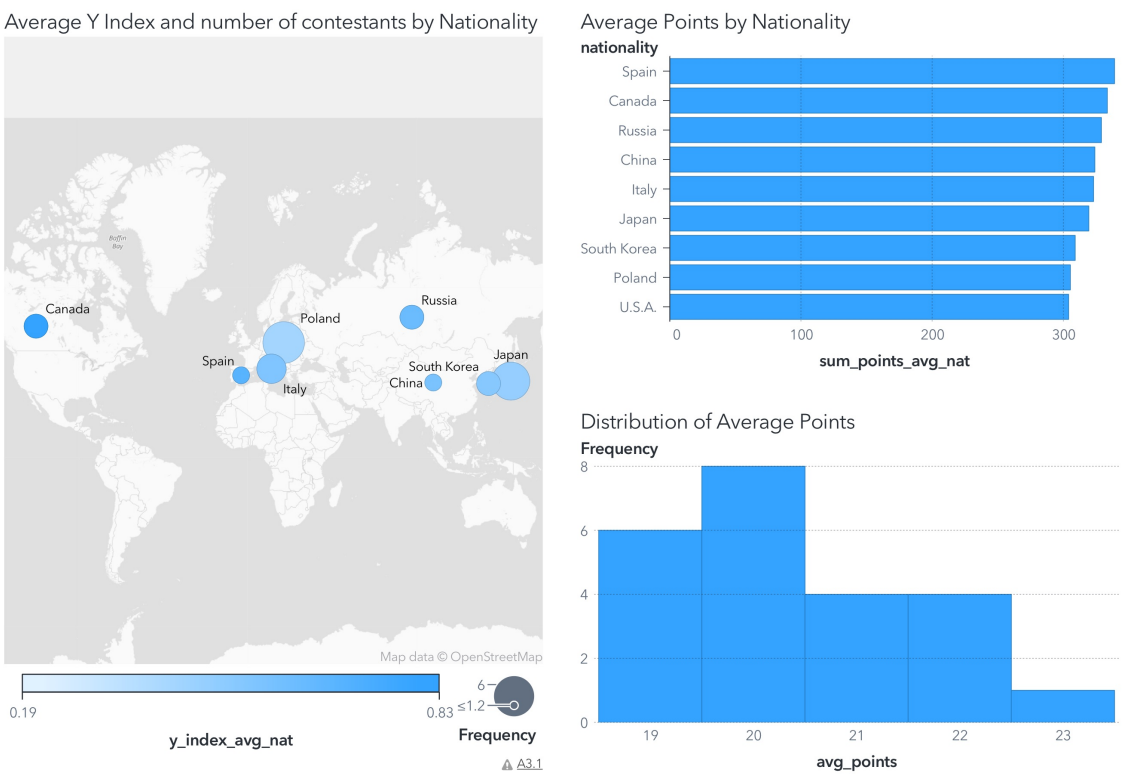


Rysunek 6:

2.2.3 Etap trzeci

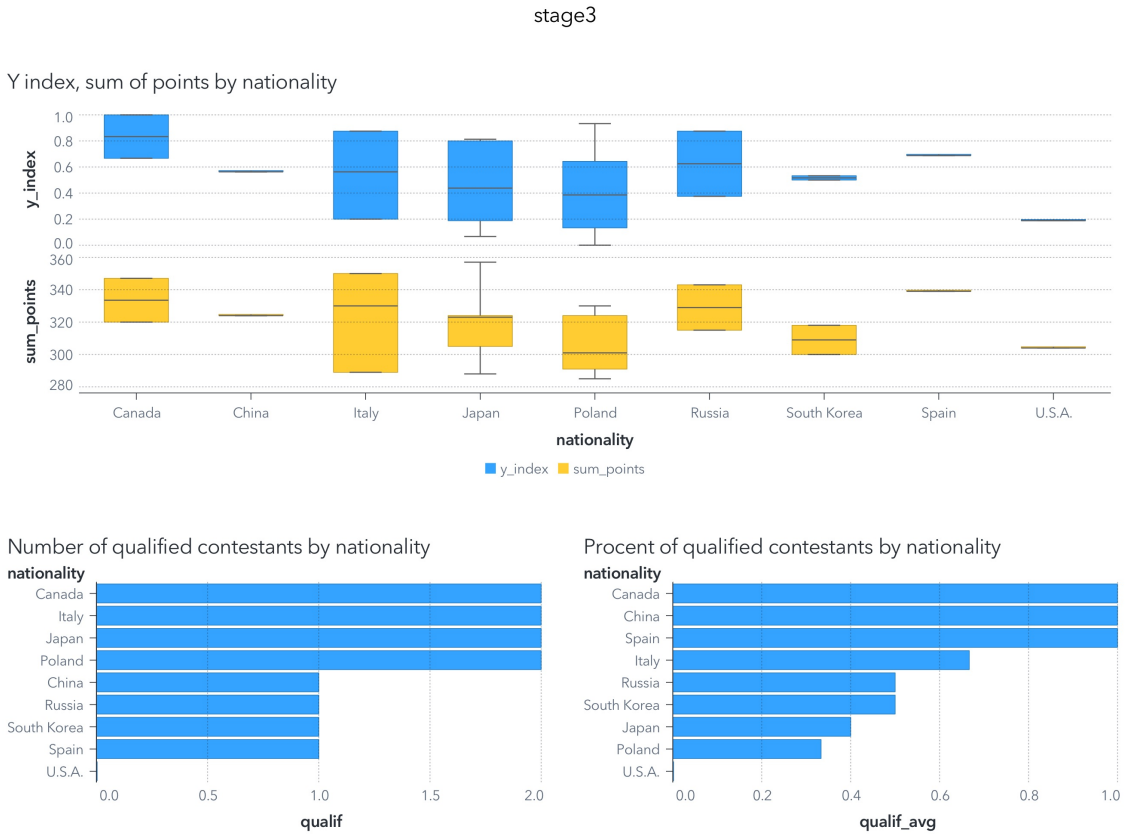
W trzecim etapie Konkursu Chopinowskiego analiza wykresów (rys. 7,8) podkreśla dalsze zaostrzenie selekcji wyników i zawężenie grupy uczestników pod względem narodowości.

Mapa przedstawia globalny rozkład uczestników na podstawie średniego wskaźnika y w podziale na narodowości, gdzie wielkość punktów symbolizuje liczbę uczestników. Kraje europejskie, takie jak Hiszpania, Włochy i Polska, utrzymują silną pozycję, co potwierdza ich znaczący udział w konkursie. Poza Europą większe punkty reprezentują Japonię, Koreę Południową i Kanadę, co wskazuje na ich wysoką liczbę uczestników. Wskaźnik y mieści się w przedziale od 0,19 do 0,83, co odzwierciedla bardziej selektywną grupę pianistów w porównaniu z wcześniejszymi etapami.
Histogram przedstawia rozkład średnich punktacji wśród uczestników trzeciego etapu. Najwięcej uczestników uzyskało wyniki w przedziale od 19 do 21 punktów, co świadczy o utrzymującym się wysokim poziomie wykonawczym. Spadek liczby uczestników z wynikami poniżej 19 punktów wskazuje na rosnącą selektywność. Z kolei niewielka grupa pianistów osiąga wyniki powyżej 22 punktów, co podkreśla ich wyjątkowe umiejętności i artystyczną dojrzałość.
Wykres słupkowy ukazuje sumę średnich punktacji uzyskanych przez uczestników z różnych krajów. Na czele znajdują się Hiszpania, Kanada i Rosja, co podkreśla wyjątkowe osiągnięcia ich reprezentantów. Włochy, Chiny i Japonia również utrzymują wysokie wyniki, a Polska i Korea Południowa pokazują swoją ciągłą konkurencyjność. Wyniki wskazują, że trzeci etap sprzyja krajom z rozwiniętą edukacją muzyczną, a różnice w wynikach pomiędzy reprezentantami z poszczególnych krajów stają się coraz mniejsze.



Rysunek 7:

Wykres pudełkowy ilustruje wartości wskaźnika y oraz sum punktów dla poszczególnych krajów. Kanada wyróżnia się bardzo wysokim wskaźnikiem y, co wskazuje na wybitne wykonania wśród jej reprezentantów, mimo że liczba uczestników z tego kraju jest stosunkowo niewielka. Polska, Japonia oraz Włochy kontynuują swoją silną pozycję w konkursie, z wysokimi sumami punktów i umiarkowaną zmiennością wyników. Z kolei Rosja, Korea Południowa i Hiszpania osiągają niższe wyniki, co sugeruje mniejszy sukces ich pianistów na tym etapie.
Wykres słupkowy liczby zakwalifikowanych uczestników wskazuje, że kraje takie jak Kanada, Włochy, Japonia i Polska mają równą liczbę reprezentantów, co świadczy o ich wyrównanym poziomie w konkursie. Chiny, Rosja oraz Korea Południowa są również obecne, choć z mniejszą liczbą pianistów.
Procentowy udział zakwalifikowanych uczestników w stosunku do liczby startujących wskazuje, że Kanada osiągnęła najwyższy wskaźnik, co podkreśla wyjątkową jakość jej reprezentacji. Hiszpania i Chiny również uzyskują wysokie wyniki w tym względzie, co wyróżnia ich pianistów na tle innych narodowości.



Rysunek 8:

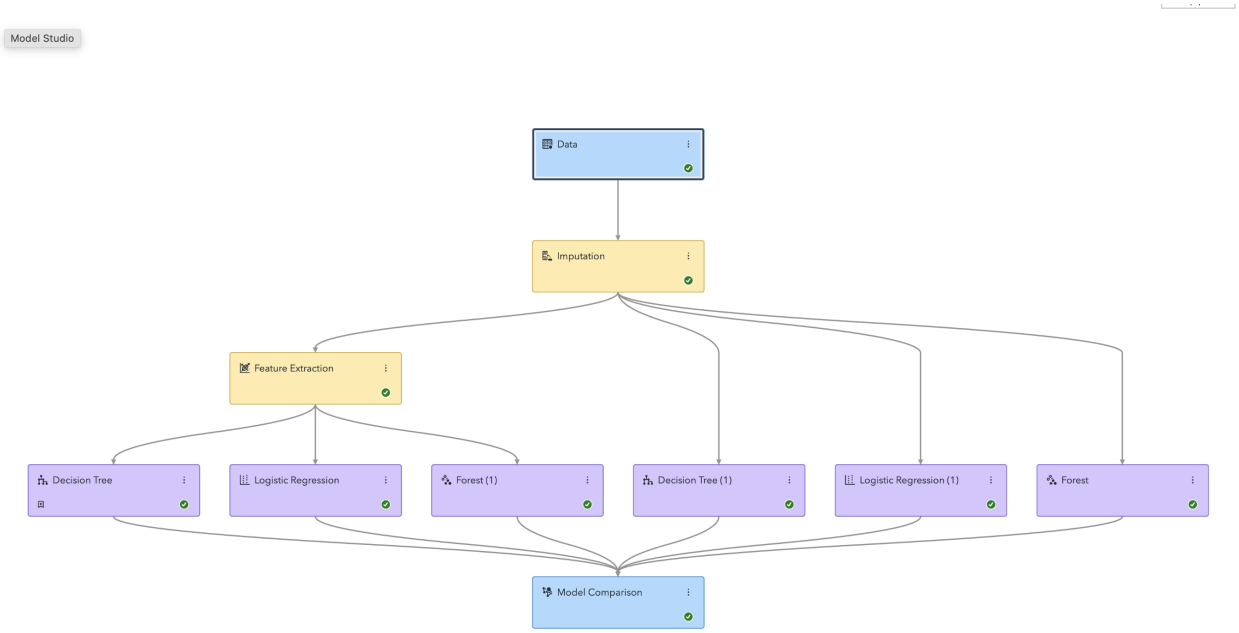
2.2.4 Podsumowanie

Analiza wyników Konkursu Chopinowskiego na przestrzeni trzech etapów ukazuje stopniowe zaostrzenie selekcji i rosnącą konkurencję wśród uczestników z różnych krajów. Pierwszy etap charakteryzował się dużą liczbą uczestników, z dominacją pianistów z krajów azjatyckich, takich jak Japonia, Chiny i Korea Południowa, oraz Rosja. Kraje te wyróżniały się zarówno pod względem liczby uczestników, jak i uzyskanych punktacji, co wskazuje na ich silne tradycje edukacji muzycznej. Rozkład wyników w tym etapie był stosunkowo równomierny, z przewagą punktacji w przedziale 18–20.
W drugim etapie widoczna była większa selekcja, co znalazło odzwierciedlenie w większej koncentracji wyników w węższym przedziale oraz zmniejszonej liczbie uczestników. Do rywalizacji dołączyły kraje europejskie, takie jak Włochy, Łotwa czy Polska, które zaczęły wyraźnie zaznaczać swoją obecność. Jednocześnie Japonia, Korea Południowa i Kanada utrzymały silne pozycje, a różnice w średnich wynikach między krajami zaczęły się zmniejszać.
Trzeci etap przyniósł dalsze zawężenie grupy uczestników, co było szczególnie widoczne w bardziej selektywnym rozkładzie punktacji. Na prowadzenie wysunęły się kraje europejskie, takie jak Hiszpania, Włochy i Polska, choć Japonia, Korea Południowa i Kanada nadal utrzymywały wysoką pozycję. Rozkład wyników wskazuje na bardzo wyrównany poziom artystyczny, z niewielką liczbą wybitnych pianistów uzyskujących wyniki powyżej 22 punktów. Podsumowując, konkurs ukazuje dominację krajów z rozwiniętą edukacją muzyczną, a selekcja na kolejnych etapach prowadzi do wyrównania wyników i wyłonienia najlepszych pianistów niezależnie od ich narodowości.

3 Zbudowanie modeli predykcyjnych - Build Models

W ramach budowania modeli skupiłam się na porównaniu trzech modeli predykcyjnych: Drzew Decyzyjnych, Regresji Logistycznej oraz Lasów Losowych. W tym celu przeanalizowałam miary jakości modeli, które otrzymałam blokiem Model Comparison. W tej części stawiałam pytanie, czy decyzje jury są na tyle przewidywalne, że wielogodzinne obrady mogłyby zostać zastąpione pracą modelu liczoną w sekundach?

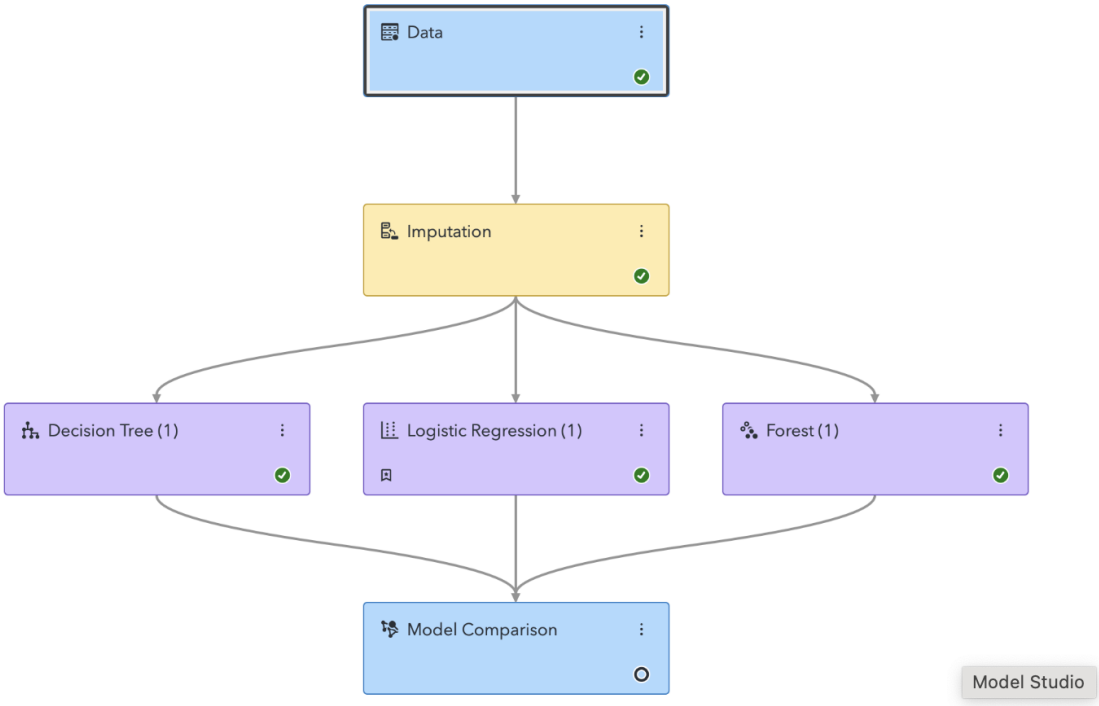
Spośród zastosowanych modeli najlepsze wyniki według SAS Viya uzyskał model drzewa decyzyjnego, który wykorzystał ekstrakcję cech w procesie klasyfikacji. Model ten osiągnął wysoką skuteczność, odzwierciedloną w kluczowych metrykach: dokładność (Accuracy) wyniosła 1.0, podobnie jak pole pod krzywą ROC (AUC), F1 Score oraz współczynnik Giniego. Wyniki te wskazują, że model ten idealnie odwzorował decyzje jurorów na etapie testowym, co czyni go wyjątkowo skutecznym narzędziem w analizie danych konkursowych.
Regresja logistyczna, mimo że w niektórych wariantach osiągnęła również bardzo dobre wyniki (np. Accuracy = 1.0 w jednym z modeli), została oceniona niżej przez system SAS Viya. Wynika to prawdopodobnie z różnic w sposobie klasyfikacji i separacji danych – regresja logistyczna zakłada liniowy charakter relacji między zmiennymi, co może być ograniczeniem w przypadku bardziej złożonych struktur danych. Drzewo decyzyjne, dzięki swojej zdolności do uchwycenia nieliniowości i prostocie interpretacji, okazało się bardziej odpowiednie w tym konkretnym zastosowaniu.



Rysunek 9: modele dla etapu 1, 2

Osiągnięcia modelu drzewa decyzyjnego wskazują, że proces oceniania uczestników pierwszego etapu Konkursu Chopinowskiego można z dużą dokładnością odzwierciedlić za pomocą algorytmów opartych na ekstrakcji cech i hierarchicznej klasyfikacji.

Wyniki analizy dostarczają cennych informacji na temat procesu oceniania i jego przewidywalności. Drzewo decyzyjne jako najlepszy model może wspierać organizatorów w ocenie spójności decyzji jurorów, a także służyć jako narzędzie umożliwiające symulacje wyników w przypadku hipotetycznych zmian w składzie jury lub kryteriach oceny. Jednocześnie, wysoka skuteczność regresji logistycznej w pewnych konfiguracjach sugeruje, że modele liniowe również mogą być wartościowym elementem przyszłych analiz, szczególnie w kontekście



Rysunek 10: modele dla etapu 3

transparentności i interpretowalności.

W analizie danych z drugiego etapu Konkursu Chopinowskiego wykorzystałam podobne podejście jak w przypadku pierwszego etapu.

Spośród zastosowanych modeli, najlepsze wyniki uzyskało drzewo decyzyjne, które według SAS Viya zostało wybrane jako model Champion. Drzewo decyzyjne osiągnęło doskonałą dokładność (Accuracy = 1.0), wartość KS = 1, współczynnik Giniego = 1 oraz F1 Score = 1.0. Wszystkie te miary wskazują, że model perfekcyjnie klasyfikował uczestników w zbiorze testowym, co podkreśla jego skuteczność w analizie danych z tego etapu konkursu.

Regresja logistyczna, mimo że nie została wybrana jako model mistrzowski, również osiągnęła zadowalające wyniki. Dokładność wyniosła 0.75, a F1 Score 0.8, co wskazuje na solidne, choć nie tak idealne jak w przypadku drzewa decyzyjnego, wyniki klasyfikacji. Warto jednak zauważyć, że ASE (średni błąd kwadratowy) w przypadku regresji logistycznej był wyższy (0.186) w porównaniu do drzewa decyzyjnego, co mogło wpłynąć na niższą ocenę tego modelu. Modele lasów losowych osiągnęły również dobre wyniki, ale nie przewyższyły wyników drzewa decyzyjnego w żadnej z kluczowych miar.
Drzewo decyzyjne ponownie okazało się najbardziej efektywnym modelem w analizie danych z drugiego etapu

Konkursu Chopinowskiego. Jego zdolność do idealnego odwzorowania decyzji jurorów czyni go narzędziem o wysokiej wartości dla dalszej analizy i potencjalnego zastosowania w procesie oceny. Dzięki przejrzystej strukturze drzewa decyzyjnego możliwe jest również łatwe interpretowanie wyników, co może być istotne w kontekście wyjaśniania decyzji podejmowanych przez modele predykcyjne.

Wyniki wskazały, że najlepszym modelem według SAS Viya okazała się Regresja Logistyczna. Model ten osiągnął najwyższą dokładność klasyfikacji (ACC) równą 1, co oznacza, że wszystkie obserwacje testowe zostały prawidłowo sklasyfikowane. Ponadto, Regresja Logistyczna charakteryzowała się najniższym średnim błędem kwadratowym (ASE) równym 0.00033, co podkreśla jej wyjątkową zdolność do przewidywania wyników.

Drzewo Decyzyjne, choć także uzyskało idealny wynik dokładności (ACC = 1), nie zostało uznane za Champion. Warto jednak zauważyć, że model ten również cechował się wysoką efektywnością, z zerowym wskaźnikiem błędu klasyfikacji (MCE = 0) oraz wysokim współczynnikiem Giniego (GINI = 1). Wydaje się, że Drzewo Decyzyjne może być bardziej interpretowalne i łatwiejsze do zastosowania w praktyce, szczególnie w kontekście oceny subiektywnych danych, takich jak oceny jurorów.

Z kolei model Lasu Losowego osiągnął niższą skuteczność w porównaniu z pozostałymi dwoma algorytmami. Jego dokładność (ACC) wyniosła 0.5, a średni błąd kwadratowy (ASE) wyniósł 0.25. Chociaż Las Losowy jest z natury bardziej złożonym algorytmem, w tym przypadku jego wyniki sugerują mniejszą przydatność w analizie danych z etapu trzeciego.

Podsumowując analizę modeli predykcyjnych, można stwierdzić, że decyzje jury Konkursu Chopinowskiego są w dużym stopniu przewidywalne przy zastosowaniu zaawansowanych algorytmów klasyfikacyjnych. Najlepszym modelem okazała się Regresja Logistyczna, która osiągnęła doskonałe wyniki w kluczowych miarach jakości, takich jak dokładność (ACC = 1) i minimalny błąd kwadratowy (ASE = 0.00033).

Z kolei model Lasów Losowych osiągnął wyraźnie niższą skuteczność, co sugeruje jego mniejszą przydatność w kontekście przewidywania decyzji jury. Wyniki te potwierdzają, że dane zebrane na etapie trzecim są na tyle spójne, iż algorytmy mogą z powodzeniem zastąpić czasochłonne i subiektywne procesy decyzyjne.

Choć technologia może wspierać lub symulować decyzje jury, należy pamiętać, że modele, mimo swojej skuteczności, bazują na historycznych danych i statystycznych wzorcach. Dlatego ostateczna decyzja o zastosowaniu takich narzędzi powinna uwzględniać również aspekty niemierzalne, takie jak artystyczna wrażliwość i indywidualne preferencje jurorów. Wyniki tej analizy pokazują jednak potencjał zastosowania modeli predykcyjnych jako wartościowego narzędzia wspierającego proces oceny.

4 Opis techniczny przygotowania analizy

4.1 Przygotowanie środowiska

1. W 'My Folder' tworzymy nowy floder: daneProjekt, do którego importujemy z komputera dane, na podstawie których będziemy przygotowywać model.
2. Otwieramy Hamburger Menu i wybieramy Develop Code and Flows
3. Wybieramy zakładkę Libraries po lewej stronie ekranu, po jej wybraniu otworzy się ona i pokaże dostępne biblioteki, te podpięte i nie.
4. Pod nazwą zakładki, w lewym rogu wybieramy ikonkę Create a new library connection
5. Ustawiamy:
    - Connection name: projektChopin
    - Library name (libref): projekt
    - Library type: SAS Base Engine

- Physical name support: klikamy ikonkę Select Folder, rozwijamy Files, rozwijamy Home, rozwijamy casuser, wybieramy folder wynikiChopin

6. Po wybraniu odpowiednich ustawień, klikamy Test connection, jeśli nie dostajemy żadnych błędów (tak powinno się stać), klikamy Save

4.2 Budownie flow: stage1, stage2, stage3

Trzy flow zbudowane są w sposób analogiczny, dla każdego z nich wybieramy inne dane: dla odpowiedniego etapu konkursu. W opisie będę bazować na danych etapu pierwszego.

1. Po prawej stronie ekranu klikamy ikonkę Submission Order i tam wybieramy Enable submission order
2. Wybieramy zakładkę Explorer, rozwijamy folder SAS Content, dalej Users, dalej MyFolder, dalej daneProjekt, gdzie znajdujemy dane do etapu pierwszego w formacie .xlxs. Przeciągamy je na pierwszy Swimlane
3. Klikamy na nowy kafelek, przechodzimy do zakładki Node i tam ustawiamy Node name: stage1
4. Wybieramy zakładkę Steps, z części Data(Input and Output) przeciągamy kafelek Import, który dołączamy do stage1
5. Z tej samej części przeciągamy kafelek Table
6. Otwieramy ustawienia kafelka i wpisujemy: Library: projekt, Table name: stage1
7. Dodajemy kafelek z części Develop: SAS Program, za pomocą którego uporządkujemy dane. W zakładce Code dla tego kafelka umieszczamy następujące:

```
data projekt.stage1;
  set projekt.stage1;
  length surname $ 30;
  title = B;
  name = C;
  surname = D;
  if strip(surname) = "" then surname = E;
  nationality = F;
  if strip(nationality) = "" then nationality = G;
  result = H;
  Dmitri_Alexeev = I;
  Sa_Chen = J;
  Dang_Thai_Son = K;
  Akiko_Ebi = L;
  Philippe_Giusiano = M;
  Nelson_Goerner = N;
  Adam_Harasiewicz = O;
  Krzysztof_Jablonski = P;
  Kevin_Kenner = Q;
  Arthur_Moreira_Lima = R;
  Janusz_Olejniczak = S;
  Piotr_Paleczny = T;
  Ewa_Poblocka = U;
  Katarzyna_Popowa_Zydron = V;
  John_Rink = W;
  Wojciech_Switala = X;
  Dina_Yoffe = Y;
  Index = Z;
  points_avg = AA;
  if _N_ > 1 then output;
  drop _18th_Chopin_Competition__Stage B C D E F G H I J K L M N O P R S T U W X Y Z V
    Q AA;
run;

data projekt.stage1;
  set projekt.stage1;
```

```
  if index(nationality, '/') > 0 then do;
    nationality = scan(nationality, 1, '/');
  end;
run;
```

8. Dodajmy następny kafelek z kodem, za pomocą którego utworzymy dwa nowe podzbiory - chcemy wyodrębnić wskaźnik y i punkty, co znacznie ułatwi dalsze przetwarzanie danych:

```
data projekt.stage1y;
  set projekt.stage1;
  if mod(_N_, 2) = 1 then output;
run;

data projekt.stage1p;
  set projekt.stage1;
  retain t n s nat;
  if mod(_N_, 2) = 1 then do;
    t = title;
    n = name;
    s = surname;
    nat = nationality;
  end;
  if mod(_N_, 2) = 0 then do;
    title = t;
    name = n;
    surname = s;
    nationality = nat;
    output;
  end;
  drop t n s nat;
run;
```

9. Rozpoczynamy nowy swimlane: z Data(Input and Output) przeciągamy kafelek Table (poniżej pierwszego swimlane - wtedy automatycznie dodamy nowy), wybieramy Library: Projekt, Table name: stage1y

10. Dodajamy kafelek z Develop: SAS Program, za pomocą którego zamienimy wartości w tabeli na binarne oraz braki danych:

```
data projekt.stage1y;
  set projekt.stage1y;
  array jury(17) $ Dmitri_Alexeev Sa_Chen Dang_Thai_Son Akiko_Ebi Philippe_Giusiano
                   Nelson_Goerner Adam_Harasiewicz Krzysztof_Jablonski Kevin_Kenner
                   Arthur_Moreira_Lima Janusz_Olejniczak Piotr_Paleczny Ewa_Poblocka
                   Katarzyna_Popowa_Zydron John_Rink Wojciech_Switala Dina_Yoffe;
  do i = 1 to 17;
    if jury[i] = 's' or jury[i] = 'a' then jury[i] = .;
    else if jury[i] = 'y' then jury[i] = 1;
    else jury[i] = 0;
  end;
  drop i;
run;
```

11. Rozpoczynamy nowy swimlane: z Data(Input and Output) przeciągamy kafelek Table (poniżej pierwszego swimlane - wtedy automatycznie dodamy nowy), wybieramy Library: Projekt, Table name: stage1y

12. Z Transform Data przeciągamy kafelek Manage Columns. Otwieramy ustawienia tego kafelka. Przyciągamy wszystkie kolumny tego zbioru oprócz Index, points_avg. Oprócz kolumn title, name, surname, nationality, result, Type ustawiamy na Numeric, Length na 8.

13. Dodajemy kafelek Table aby otrzymać tabelkę wyjściową, ustawiamy Library: Porojekt, Table Name: stage1y

14. Z Transform Data wybieramy kafelek Calculate Columns. Rozwijamy ten kafelek, w Options, pod Available Columns, klikamy new. Otwiera się okienko Expression Builder, na ekranie głównym definiujemy nowe obliczenie, zapisujemy je jako Column name: num_yes

```
sum(Dmitri_Alexeev, Sa_Chen, Dang_Thai_Son, Akiko_Ebi, Philippe_Giusiano,
    Nelson_Goerner, Adam_Harasiewicz, Krzysztof_Jablonski, Kevin_Kenner,
    Arthur_Moreira_Lima, Janusz_Olejniczak, Piotr_Paleczny, Ewa_Poblocka,
    Katarzyna_Popowa_Zydron, John_Rink, Wojciech_Switala, Dina_Yoffe)
```

15. W analogiczny sposób tworzymy drugą kolumnę: missing_jury:

```
sum(
MISSING(Dmitri_Alexeev),
MISSING(Sa_Chen),
MISSING(Dang_Thai_Son),
MISSING(Akiko_Ebi),
MISSING(Philippe_Giusiano),
MISSING(Nelson_Goerner),
MISSING(Adam_Harasiewicz),
MISSING(Krzysztof_Jablonski),
MISSING(Kevin_Kenner),
MISSING(Arthur_Moreira_Lima),
MISSING(Janusz_Olejniczak),
MISSING(Piotr_Paleczny),
MISSING(Ewa_Poblocka),
MISSING(Katarzyna_Popowa_Zydron),
MISSING(John_Rink),
MISSING(Wojciech_Switala),
MISSING(Dina_Yoffe)
)
```

16. Dodajemy kafelek Table aby otrzymać tabelkę wyjściową, ustawiamy Library: Porojekt, Table Name: stage1y

17. Ponownie przeciągamy kafelek Calculate Columns, tworzymy nową kolumnę y_index tworząc nowe obliczenie zdefiniowane następująco:

```
round(num_yes/ (17 - missing_jury), 0.001)
```

18. Dodajemy kafelek Table aby otrzymać tabelkę wyjściową, ustawiamy Library: Porojekt, Table Name: stage1y

19. Rozpoczynamy nowy swimlane: przeciągamy kafelek Table, wybieramy Library: Projekt, Table name: stage1p

20. Dodajemy kafelek SAS Program, za pomocą którego zamienimy odpowiednie wartości na braki danych:

```
data projekt.stage1p;
  set projekt.stage1p;
  array jury(17) $ Dmitri_Alexeev Sa_Chen Dang_Thai_Son Akiko_Ebi Philippe_Giusiano
                   Nelson_Goerner Adam_Harasiewicz Krzysztof_Jablonski Kevin_Kenner
                   Arthur_Moreira_Lima Janusz_Olejniczak Piotr_Paleczny Ewa_Poblocka
                   Katarzyna_Popowa_Zydron John_Rink Wojciech_Switala Dina_Yoffe;
  do i = 1 to 17;
    if jury[i] = 's' or jury[i] = 'a' then jury[i] = .;
  end;
  drop i;
run;
```

21. Rozpoczynamy nowy swimlane: przeciągamy kafelek Table, wybieramy Library: Projekt, Table name: stage1p

22. Z Transform Data przeciągamy kafelek Manage Columns. Otwieramy ustawienia tego kafelka. Przyciągamy wszystkie kolumny tego zbioru oprócz Index, points_avg. Oprócz kolumn title, name, surname, nationality, result, Type ustawiamy na Numeric, Length na 8.

23. Dodajemy kafelek Table aby otrzymać tabelkę wyjściową, ustawiamy Library: Porojekt, Table Name: stage1p

24. Dodajemy kafelek Calculate Columns, w którym tworzymy nowe kolumny z wykorzystaniem expression builder. Robimy to analogicznie do poprzednich, na podstawie następujących wyrażeń:

```
/*sum_points*/
sum(Dmitri_Alexeev, Sa_Chen, Dang_Thai_Son, Akiko_Ebi, Philippe_Giusiano,
    Nelson_Goerner, Adam_Harasiewicz, Krzysztof_Jablonski, Kevin_Kenner,
    Arthur_Moreira_Lima, Janusz_Olejniczak, Piotr_Paleczny, Ewa_Poblocka,
    Katarzyna_Popowa_Zydron, John_Rink, Wojciech_Switala, Dina_Yoffe)

/*max_points*/
max(Dmitri_Alexeev, Sa_Chen, Dang_Thai_Son, Akiko_Ebi, Philippe_Giusiano,
    Nelson_Goerner, Adam_Harasiewicz, Krzysztof_Jablonski, Kevin_Kenner,
    Arthur_Moreira_Lima, Janusz_Olejniczak, Piotr_Paleczny, Ewa_Poblocka,
    Katarzyna_Popowa_Zydron, John_Rink, Wojciech_Switala, Dina_Yoffe)

/*min_points*/
min(Dmitri_Alexeev, Sa_Chen, Dang_Thai_Son, Akiko_Ebi, Philippe_Giusiano,
    Nelson_Goerner, Adam_Harasiewicz, Krzysztof_Jablonski, Kevin_Kenner,
    Arthur_Moreira_Lima, Janusz_Olejniczak, Piotr_Paleczny, Ewa_Poblocka,
    Katarzyna_Popowa_Zydron, John_Rink, Wojciech_Switala, Dina_Yoffe)

/*median_points*/
median(Dmitri_Alexeev, Sa_Chen, Dang_Thai_Son, Akiko_Ebi, Philippe_Giusiano,
    Nelson_Goerner, Adam_Harasiewicz, Krzysztof_Jablonski, Kevin_Kenner,
    Arthur_Moreira_Lima, Janusz_Olejniczak, Piotr_Paleczny, Ewa_Poblocka,
    Katarzyna_Popowa_Zydron, John_Rink, Wojciech_Switala, Dina_Yoffe)

/*avg_points*/
mean(Dmitri_Alexeev, Sa_Chen, Dang_Thai_Son, Akiko_Ebi, Philippe_Giusiano,
    Nelson_Goerner, Adam_Harasiewicz, Krzysztof_Jablonski, Kevin_Kenner,
    Arthur_Moreira_Lima, Janusz_Olejniczak, Piotr_Paleczny, Ewa_Poblocka,
    Katarzyna_Popowa_Zydron, John_Rink, Wojciech_Switala, Dina_Yoffe)
```

25. Dodajemy kafelek Table aby otrzymać tabelkę wyjściową, ustawiamy Library: Porojekt, Table Name: stage1p

26. Dodajemy kafelek SAS Program, na podstawie którego, wyznaczamy globalne statystki:

```
proc sql;
  create table projekt.stage1p as
  select *, max(max_points) as total_max, min(min_points) as total_min
  from projekt.stage1p;
run;
```

27. Rozpoczynamy nowy swimlane: przeciągamy kafelek Query z Transform Data. Prawym przyciskiem myszy klikamy w ten kafelek i rozwijamy menu, klikamy Add input Port

28. Przeciągamy dwa kafelki table i dołączamy je do Input Ports Query. Tabelka 1 to stage1p, tabelka 2 to stage1y

29. Otwieamy kafelek Query. Do Columns przeciągamy następujące kolumny: (z tabelki t1) title, name, surname, nationality, sum_points, max_points, min_points, median_points, avg_points, total_max, total_min, (z tabelki t2) y_index. Następnie w zakładce Join wybieramy Inner Join t1 z t2 po kolumnach name, surname

30. Dodajemy kafelek Table aby otrzymać tabelkę wyjściową, ustawiamy Library: Porojekt, Table Name: stage1results

31. Dodajemy kafelek Sort z Transform Data, gdzie dodajmy kolumny y_index, avg_points, sum_points, ustawiając sortowanie na Descending. Ten krok nie ma istotnego wpływu na dalszą analizę danych, był przydatny, aby obejrzeć dane po wstępnej obróbce.

4.3 Budowanie flow: qualif

Dla pierwszego i drugiego etapu konkursu do stworzenia zmiennej binarnej qualif wykorzystamy agregację danych. Flow dla pierwszego i drugiego jest analogiczny.

1. Z Transform Data wybieramy kafalek Union Rows. Prawy przyciskiem myszy rozwiajmy jego menu i klikamy Add input ports
2. Przeciągamy dwa kafelki Table, które dołączamy do Input Ports. Pierwsza tabelka to stage1results, druga to stage2results
3. Przeciągamy kolejny kafelek Table jak tabelę wyjściową. Zapisujemy ją jako stage1qualif w bibliotece projekt.
4. W następnym kroku przeciągamy kafelek SAS Program, aby stworzyć zmienną celu qualif:

```
proc sql;
    create table projekt.stage1qualif as
    select
        title,
        name,
        surname,
        nationality,
        count(*) as qualif
    from projekt.stage1qualif
    group by title, name, surname, nationality
  having qualif = 2;
quit;
```

5. Przeciągamy kafelek Query
6. Przeciągamy kafelek Table, aby dołączyć w drugim Input Port tabelę stage1results
7. Otwieramy kafelek Query: wybieramy wszystkie kolumny z tabeli stage1results oraz kolumnę qualif z stage1qualif. Robimy Full join t2(stage1results) z t1(stage1qualif) po zmiennych name, surname
8. Wyniki zapisujemy w tabeli stage1qualif, w bibliotece projekt, przeciągając kafelek Table to portu wyjściowego Query
9. Dodajemy kafelek z SAS Program, którym zmienimy wartości qualif na binarne:

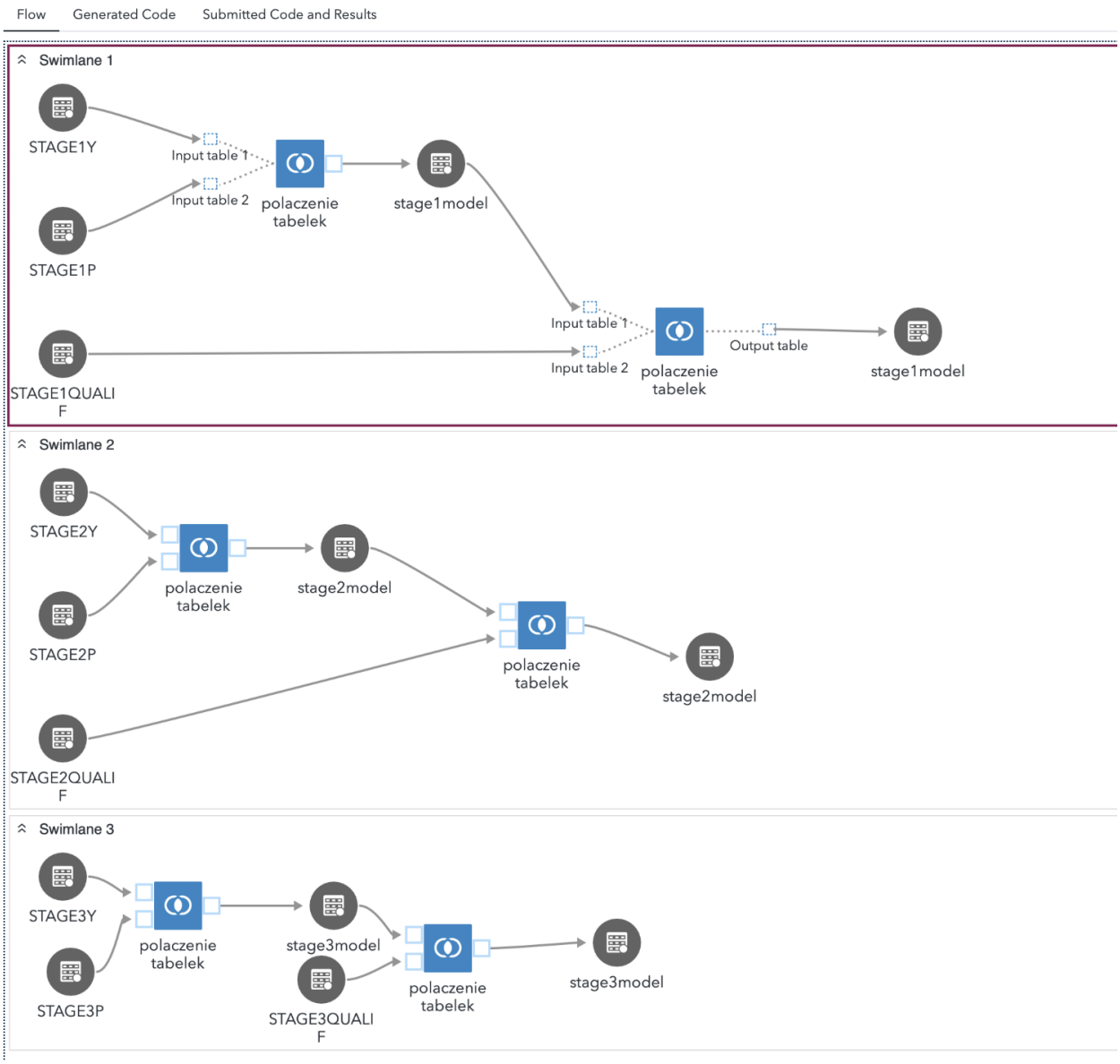
```
data projekt.stage1qualif;
    set projekt.stage1qualif;
    if qualif = 2 then qualif = 1;
    else qualif = 0;
run;
```

W ramach danych opublikowanych przez NIFC nie było tabeli finałowej. Dlatego też zmienna qualif została stworzona ręcznie za pomocą jednego kafelka z kodem:

```
data projekt.stage3qualif;
  set projekt.stage3results;
  if name = 'Leonora' and surname = 'Armellini' then qualif = 1;
  else if name = 'J J Jun Li' and surname = 'Bui' then qualif = 1;
  else if name = 'Alexander' and surname = 'Gadjiev' then qualif = 1;
  else if name = 'Martin' and surname = 'Garcia Garcia' then qualif = 1;
  else if name = 'Eva' and surname = 'Gevorgyan' then qualif = 1;
  else if name = 'Aimi' and surname = 'Kobayashi' then qualif = 1;
  else if name = 'Jakub' and surname = 'Kuszlik' then qualif = 1;
  else if name = 'Hyuk' and surname = 'Lee' then qualif = 1;
  else if name = 'Bruce (Xiaoyu)' and surname = 'Liu' then qualif = 1;
  else if name = 'Kamil' and surname = 'Pacholec' then qualif = 1;
  else if name = 'Hao' and surname = 'Rao' then qualif = 1;
  else if name = 'Kyohei' and surname = 'Sorita' then qualif = 1;
  else qualif = 0;
run;
```

4.4 Budowanie flow: flow_model

Dla danych z pierwszego drugiego i trzeciego etapu postępujemy analogicznie.



Rysunek 11: flow_model

1. Przeciągamy kafelek Query i dodajemy drugi Input Port
2. Przeciągamy dwa kafelki Table, do których przypisujemy tabelki odpowiedni stage1y, stage1p.
3. Rozwijamy kafelek Query. Wybieramy wszystkie kolumny z tabeli t1(stage1y) oprócz results oraz wszystkie z tabeli t2(stage1p) oprócz title, name, surname, nationality, result. Robimy Inner Join po zmiennych surname, name
4. Przeciągamy kafelek Table tworząc tablkę wyjściową stage1model
5. Przeciągamy drugi kafelek Query. W pierwszym input port podłączmy nową tablekę wyjściową stage1model, do drugiego przeciągamy kafelek Table, do którego podpinamy tabelkę stage1qualif.

6. Otwieramy kafelek Query, wybieramy wszystkie kolumny z tabelki t1(stage1model) oprócz name, surname oraz kolumnę qualif z tableki t2(stage1qualif). Robimy Inner Join t1 z t2 po zmiennych name, surname

7. Wyniki agregacji zapisujemy w tablce wyjściowej stage1model Ten flow posłużył przygotowaniu tabel pod model predykcyjny

4.5 Modele predykcyjne - Build Models

W częsci modele predykcyjne stworzyłam trzy projekty, każdy dla odrębnego etapu konkursy

1. Z Menu Hambuerger wybieramy Build Models

2. W prawy górnym roku wybieramy New project, wybieramy następujące:

   - Name: projekt_stage1
   - Type: Data Mining and Machine Learning
   - Data: importujemy plik stage1results

3. Wchodzimy w zakładkę Data, odnajdujemy zmienną qualif, wybieramy ją. Po lewej stronie ekranu otworzy się wówczas zakłdak dla tej zmiennej, ustawiamy Role: Target

4. Wchodzimy w zakładkę Pipelines, gdzie będziemy budować model

5. Do kafelka data dokładamy Imputation, a do Imputation Feature Extraction z zakładki Data Mining Preprocessing.

6. Do kafelka Imputation dokładamy Decision Tree oraz Forest z zakładki Supervised Learning, analogicznie robimy z kafelkiem Feature Extraction.

7. tworzymy nowy Pipeline. W nim chcemy znaleźć najlepszy model regresji logistyczny porównując różne Selection method

8. Do kafelka Data dodajmy Imputation z Data Mininng Preprocessing, a następnie do Imputation, cztery kafelki Logistic Regression z Supervised Learning

9. W kolejnych kafelkach Logistic Regression ustawiamy selection method jako Stepwise, Lasso, Forward, Backward

10. W prawym górnym rogu klikamy Run Pipeline

11. Po wykonaniu, w kafelku Model Comparison, który powinien pojawić się automatycznie, klikamy prawym przyciskiem myszy i wybieramy Results. Najlepszy model ma w kolumnie Champion symbol gwiazdki. W tym przypadku najlepiej wypada model regresji logistycznej z Selection Method Lasso.

12. Wracamy do Pipeline 1, dodajemy Logistic Regression z Selection Method Lasso do Imputation i Feautre Extraction

13. Klikamy Run Pipeline

W ten sposób tworzymy model dla pierwszego i drugiego etapu konkursu, dla trzeciego pomijamy jedynie Feature Extraction.