Politechnika Krakowska
Wydział Informatyki i Telekomunikacji

Projekt z przedmiotu
Metody i narzędzia analizy
dużych zbiorów danych

Projekt nr 2

Kierunek: Informatyka

Stopień studiów: II stopnia

Specjalizacja: Data Science

Wykonali:

Marek Dalida
Maciej Gicala
Rafał Gęgotek

1. Cel projektu

Celem projektu było wykorzystanie metod uczenia maszynowego na platformie Apache Spark do rozwiązywania problemów klasyfikacji oraz regresji. Dla każdego problemu powinny zostać zaimplementowane dwie metody.

2. Opis problemu wraz ze szczegółowym opisem danych

Wybranym przez nas zbiorem danych został TLC Trip Record Data, którego opis znajduje się na stronie https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page. Dane dotyczą kursów taksówek z Nowego Jorku za okres styczeń 2021. Zbiór danych posiada rozmiar 120 mb i jest utworzony w formacie csv.
Zbiór można pobrać pod adresem:
https://s3.amazonaws.com/nyc-tlc/trip+data/yellow_tripdata_2021-01.csv

Opis poszczególnych kolumn w zbiorze:

Nazwa pola	Opis
VendorId	Nazwa firmy świadczącej usługi 1= Creative Mobile Technologies, LLC; 2= VeriFone Inc
tpep_pickup_datetime	Czas rozpoczęcia
tpep_dropoff_datetime	Czas zakończenia
Passenger_count	Ilość pasażerów w pojeździe
Trip_distance	Odległość zmierzona przek taksometr
PULocationID	Strefa TLC Taxi gdzie taksometr rozpoczął pracę
DOLocationID	Strefa TLC Taxi gdzie taksometr zakończył pracę
RateCodeID	Kod stawki 1= Standard rate 2=JFK 3=Newark 4=Nassau or Westchester 5=Negotiated fare 6=Group ride
Store_and_fwd_flag	Flaga informująca czy rekord był przechowywany w taksówce ponieważ pojazd nie miał połączenia z serwerem Y= store and forward trip

	N= not a store and forward trip
Payment_type	Zmienna informująca w jaki sposób pasażer zapłacił za podróż 1= Karta kredytowa 2= Gotówka 3= Brak opłaty 4= Spór o płatność 5= Nie wiadomo 6= Pusta podróż
Fare_amount	Taryfa czasowo-dystansowa obliczona przez licznik.
Extra	Różne dodatkowe opłaty.
MTA_tax	0,50 USD podatku MTA, który jest automatycznie naliczany na podstawie taksometru w użyciu.
Improvement_surcharge	0,30 USD dopłaty improvement
Tip amount	Kwota napiwku automatycznie naliczana w przypadku karty kredytowej
Tolls_amount	Łączna kwota opłat zapłaconych w podróży.
Total_amount	Łączna kwota zapłacona przez pasażerów, nie zawiera napiwków.

W naszym projekcie zajęliśmy się analizą dwóch zagadnień na zbiorze danych.
Pierwszym problemem była klasyfikacja w której celem było zbudowanie modelu, który będzie przewidywał typ płatności.
Drugim problemem przez nas analizowanym było zbudowanie modelu, którym będzie przewidywał wartość kwoty napiwku wniesionej przez pasażera przy użyciu metod regresji.

3. Opis wykorzystanych metod

Wykorzystane metody regresji i klasyfikacji:

- **Regresja liniowa** – w modelowaniu statystycznym, metody oparte o liniowe kombinacje zmiennych i parametrów dopasowujących model do danych. Dopasowana linia lub krzywa regresji reprezentuje oszacowaną wartość oczekiwaną zmiennej **y** przy konkretnych wartościach innej zmiennej lub zmiennych **x**.Model regresji liniowej zakłada, że istnieje liniowa (afiniczna) relacja pomiędzy zmienną zależną y wektorem p × 1 regresorów x . Zależność ta jest modelowana przez uwzględnienie składnika losowego (błędu) ε i , który jest zmienną losową. Dokładniej, model ten jest postaci:

$$y_i = \beta_0 1 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \varepsilon_i, \qquad i = 1, \ldots, n,$$

- **Regresja logistyczna** – jedna z metod regresji używanych w statystyce w przypadku, gdy zmienna zależna jest na skali dychotomicznej (przyjmuje tylko dwie

wartości). Zmienne niezależne w analizie regresji logistycznej mogą przyjmować charakter nominalny, porządkowy, przedziałowy lub ilorazowy. W przypadku zmiennych nominalnych oraz porządkowych następuje ich przekodowanie w liczbę zmiennych zero-jedynkowych taką samą lub o 1 mniejszą niż liczba kategorii w jej definicji.

- **Drzewa decyzyjne** w uczeniu maszynowym służą do wyodrębniania wiedzy z zestawu przykładów. Zakładamy, że posiadamy zestaw przykładów: obiektów opisanych przy pomocy atrybutów, którym przyporządkowujemy jakąś decyzję.
- **Las losowy** – metoda zespołowa uczenia maszynowego dla klasyfikacji, regresji i innych zadań, która polega na konstruowaniu wielu drzew decyzyjnych w czasie uczenia i generowaniu klasy, która jest dominantą klas (klasyfikacja) lub przewidywaną średnią (regresja) poszczególnych drzew]. Losowe lasy decyzyjne poprawiają tendencję drzew decyzyjnych do nadmiernego dopasowywania się do zestawu treningowego

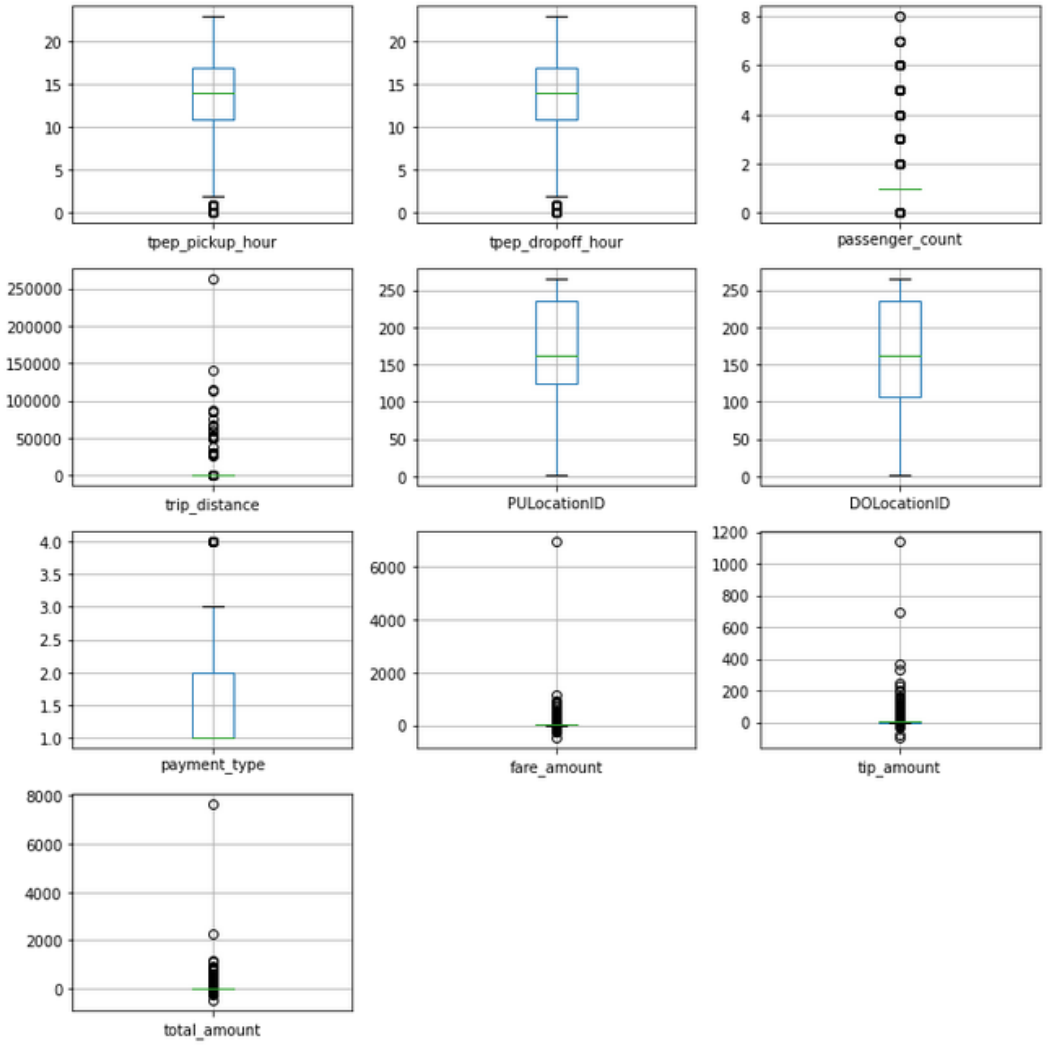
4. Analiza statystyczna danych oraz korelacja między atrybutami

4.1. Statystyki danych wejściowych

Po wczytaniu danych zmienne tpep_pickup_datetime i tpep_dropoff_datetime zostały przekonwertowane na wartości liczbowe odpowiadające godzinie.

	tpep_pickup_hour	tpep_dropoff_hour	passenger_count	trip_distance	PULocationID	DOLocationID	payment_type	fare_amount	tip_amount	tot
count	1.369765e+06	1.369765e+06	1.271413e+06	1.369765e+06	1.369765e+06	1.369765e+06	1.271413e+06	1.369765e+06	1.369765e+06	1.
mean	1.376327e+01	1.389418e+01	1.411508e+00	4.631982e+00	1.652472e+02	1.614956e+02	1.280521e+00	1.209662e+01	1.918099e+00	1.
std	4.556054e+00	4.592244e+00	1.059833e+00	3.939042e+02	6.783849e+01	7.210800e+01	4.916921e-01	1.291338e+01	2.597153e+00	1.
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	-4.900000e+02	-1.000000e+02	-4.
25%	1.100000e+01	1.100000e+01	1.000000e+00	1.000000e+00	1.240000e+02	1.070000e+02	1.000000e+00	6.000000e+00	0.000000e+00	1.
50%	1.400000e+01	1.400000e+01	1.000000e+00	1.700000e+00	1.620000e+02	1.620000e+02	1.000000e+00	8.500000e+00	1.860000e+00	1.
75%	1.700000e+01	1.700000e+01	1.000000e+00	3.020000e+00	2.360000e+02	2.360000e+02	2.000000e+00	1.350000e+01	2.750000e+00	1.
max	2.300000e+01	2.300000e+01	8.000000e+00	2.631633e+05	2.650000e+02	2.650000e+02	4.000000e+00	6.960500e+03	1.140440e+03	7.

4.2. Wykresy box plot dla każdej zmiennej w zbiorze



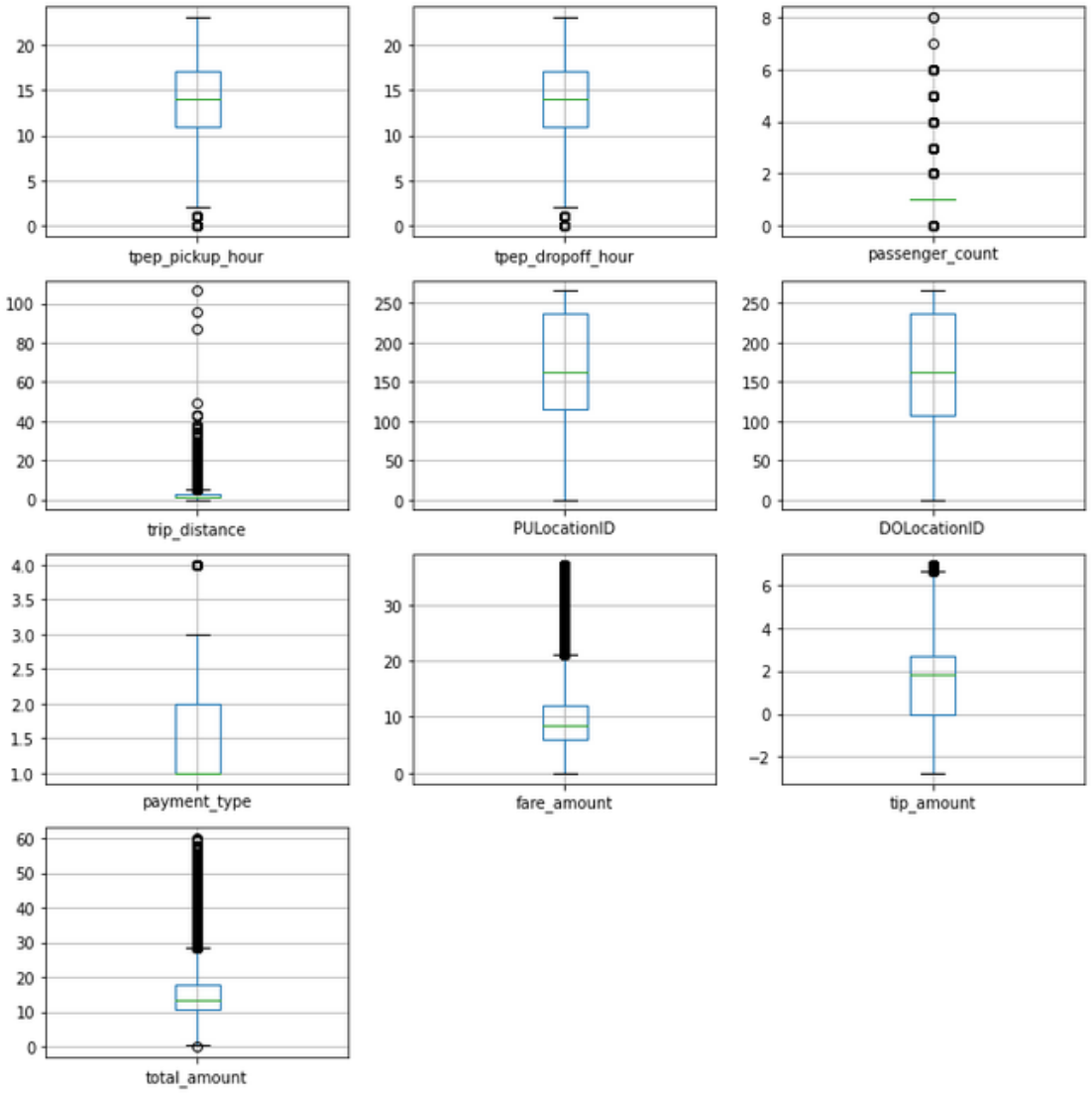
Z powyższych wykresów można zauważyć, że dla zmiennych trip distance, fare amount, tip amount, total amount jest sporo wartości odstających.

W przypadku zmiennej passager count, można zauważyć że najczęściej transport dotyczył jednego pasażera, ale zdarzały się też przypadki gdy pasażerów było aż 8.

4.3. Eliminacja wartości odstających i nieprawidłowych

W kolejnym kroku zostały wyeliminowane wartości odstające, czyli te których różnica od średniej było większa niż trzykrotność odchylenia standardowego.
Zostały także usunięte wartości opłat które były mniejsze od 0.

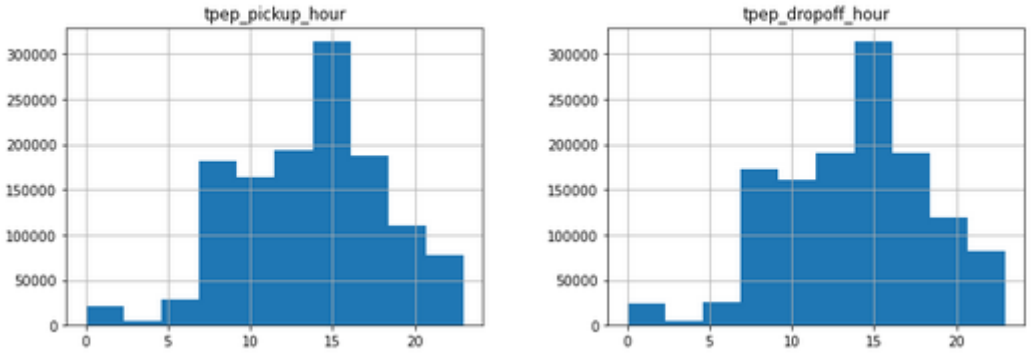
Zostało usuniętych w wyniku powyższych działań 83410 wierszy.
Poniżej zostały przedstawione wykresy box plot dla zmiennych po powyższych zmianach.



Z powyższych wykresów wynika, że udało się znacznie zmniejszyć ilość wartości odstających dla zmiennych, których ten problem dotyczył. Reszta danych zostanie uwzględniona, gdyż niekoniecznie musi mieć negatywny wpływ na skuteczność modeli regresji i klasyfikacji.

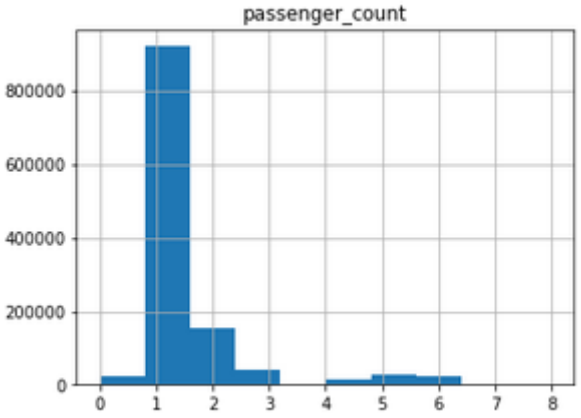
4.4. Histogramy dla każdej zmiennej

4.4.1. Rozkład czasu rozpoczęcia i zakończenia podróży



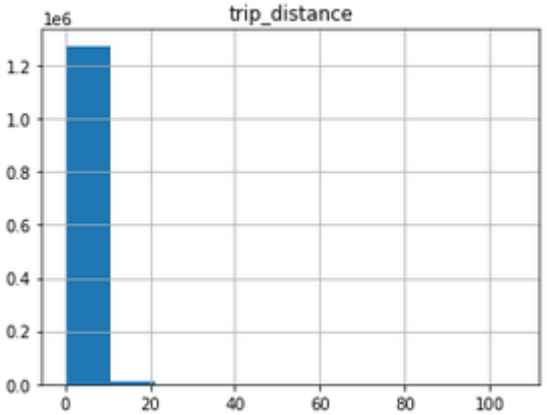
Rozkład czasu rozpoczęcia i zakończenia podróży wskazuje na to, że najmniejszy ruch kursów jest w nocy, w ciągu dnia największe natężenie pasażerów występuje w okolicy godziny 15.

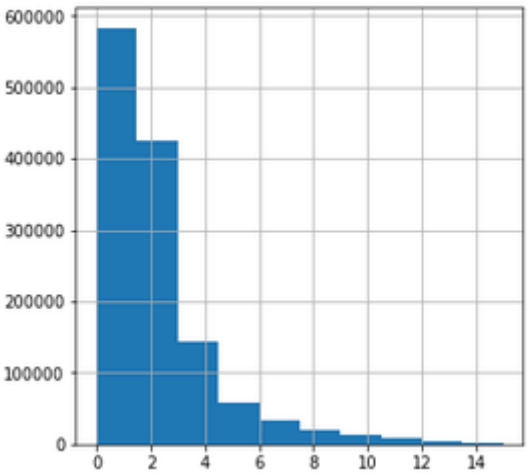
4.4.2. Rozkład liczby pasażerów



Widać, najczęściej kurs dotyczył jednego pasażera.

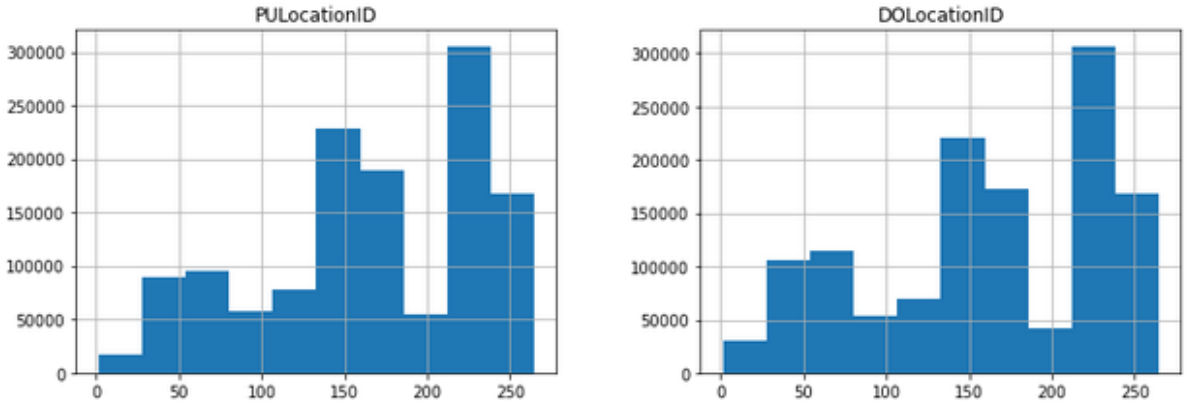
4.4.3. Rozkład odległości podróży





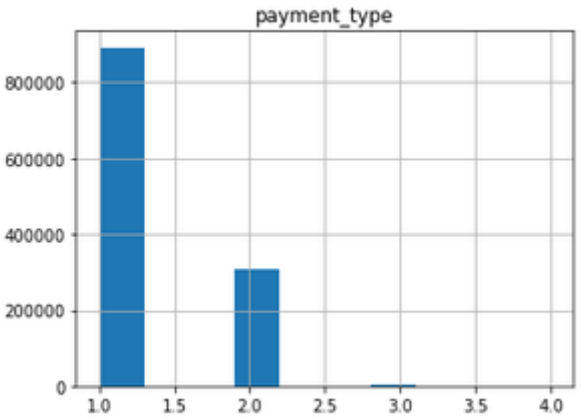
Zdecydowana większość kursów odbywała się na niewielkie odległości do 5 mil.

4.4.4. Rozkład zmiennych PULocationID i DOLocationID



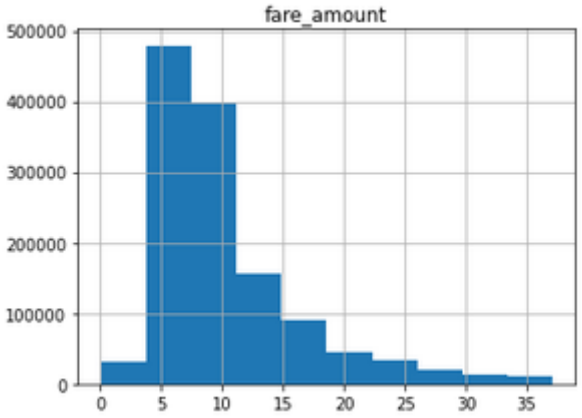
Widać, że strefy odbioru oraz dojazdu dotyczyły zazwyczaj podobnych stref, co potwierdza uwiarygadnia fakt, że przejazdy dotyczyły zazwyczaj niewielkich odległości.

4.4.5. Rozkład dla zmiennej typu płatności



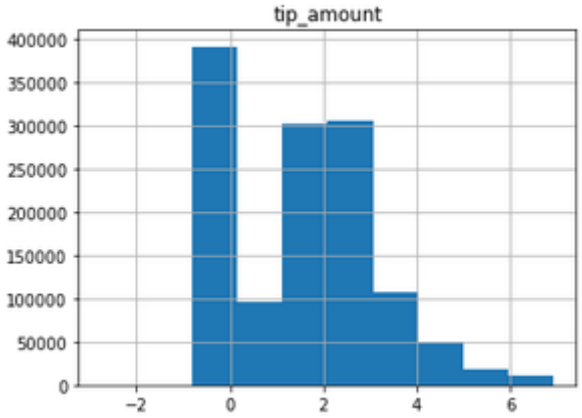
Z wykresu można zauważyć, że klienci płacili za przejazdy głównie kartą kredytową oraz gotówką, zdecydowanie największa ilość transakcji został sfinalizowana kartą.

4.4.6. Rozkład dla zmiennej fare_amount



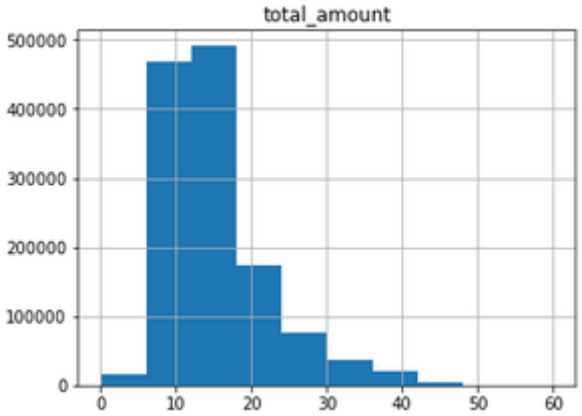
Wartość opłaty za taryfę mieściła się najczęściej w granicach od 5 do 10 dolarów.

4.4.7. Rozkład dla wartości napiwku



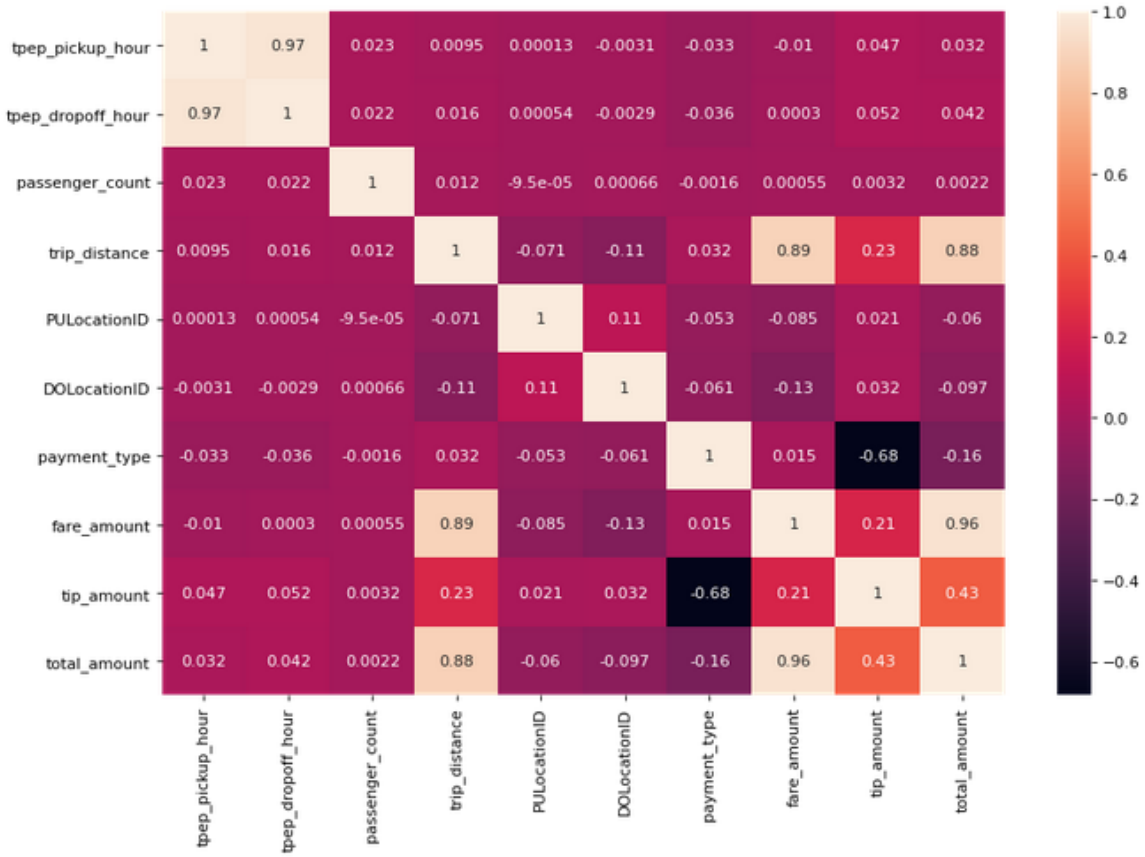
Bardzo często klienci nie płacili napiwku, lub jeśli płacili wynosił on zazwyczaj od 1 do 3 dolarów.

4.4.8. Rozkład dla zmiennej total amount



Z histogramu możemy zauważyć, że łączna opłata za przejazd mieściła się często w granicach od 5 do 20 dolarów.

4.5. Sprawdzenie korelacji danych



Z powyższej macierzy widać, że zmienne dotyczące czasu rozpoczęcia są ze sobą bardzo mocno skorelowane, wynik 0,97 wydaje się to logiczne biorąc pod uwagę fakt, że zdecydowana większość przejazdów odbywała się na niewielkie odległości.

Widać też, że zmienne fare_amount oraz total amount są mocno skorelowane z trip distance, wydaje się to również zrozumiałe ponieważ im dłuższa trasa tym proporcjonalnie powinny rosnąć opłaty.

Można również zauważyć, że payment type jest ujemnie skorelowane z tip amount - wynik -0,68.

5. Ocena i porównanie modeli

5.1. Klasyfikacja dla typu płatności

5.1.1. Podział danych na treningowe i testowe

Przy użyciu VectorAssembler został zbudowany wektor zmiennych, następnie zbiór został podzielony w proporcjach 0,7 i 0,3 na zbiór treningowy i testowy.

```
+--------------------+------------+
|            features|payment_type|
+--------------------+------------+
|[1.0,2.1,142.0,43...|           2|
|[1.0,0.2,238.0,15...|           2|
|[1.0,14.7,132.0,1...|           1|
|[0.0,10.6,138.0,1...|           1|
|[1.0,4.94,68.0,33...|           1|
+--------------------+------------+
only showing top 5 rows
```

5.1.2. Regresja logistyczna

```
+-----+----------+--------------------+--------------------+
|label|prediction|         probability|            features|
+-----+----------+--------------------+--------------------+
|    3|       1.0|[1.16958811165589...|(8,[0,1,2,3],[1.0...|
|    2|       1.0|[1.16958811165589...|(8,[0,2,3,4],[1.0...|
|    1|       1.0|[1.16958811165589...|(8,[0,2,3,4],[1.0...|
|    1|       1.0|[1.16958811165589...|(8,[0,2,3,4],[1.0...|
|    1|       1.0|[1.16958811165589...|(8,[0,2,3,4],[1.0...|
+-----+----------+--------------------+--------------------+
only showing top 5 rows

Test Error = 0.266523
Test accuracy = 0.7334769285309513
F1 score =  0.6207044303071524
```

Dla regresji logistycznej wynik accuracy wyniósł 0,73, natomiast biorąc pod uwagę fakt, że pomiędzy klasami typu płatności istnieje duża dysproporcja w zbiorze wydaje się, że bardziej miarodajnym jest wynik zwracany przez metrykę F1 czyli w tym wypadku 0,62.

5.1.3. Klasyfikator Random Forest

```
+----------+------------+--------------------+--------------------+
|prediction|indexedLabel|         probability|            features|
+----------+------------+--------------------+--------------------+
|       1.0|         2.0|[0.05670026580926...|(8,[0,1,2,3],[1.0...|
|       1.0|         1.0|[0.26556485278456...|(8,[0,2,3,4],[1.0...|
|       0.0|         0.0|[0.71955524729794...|(8,[0,2,3,4],[1.0...|
|       0.0|         0.0|[0.68479566764509...|(8,[0,2,3,4],[1.0...|
|       0.0|         0.0|[0.66743780026188...|(8,[0,2,3,4],[1.0...|
+----------+------------+--------------------+--------------------+
only showing top 5 rows

Test Error = 0.0381342
Test accuracy = 0.961865806527852
F1 score =  0.9574823287763095
```

```
RandomForestClassificationModel: uid=RandomForestClassifier_be0a6dc31302, numTrees=10, numClasses=4, numFeatures=8
```

W przypadku zastosowania klasyfikatora lasu losowego wyniki znacząco się poprawiły w porównaniu do regresji logistycznej. Accuracy wyniosło 0,9618, a F1 wyniosło 0,9574. Wytrenowany model składa się z 10 drzew.

5.1.4. Klasyfikator Drzewa decyzyjnego

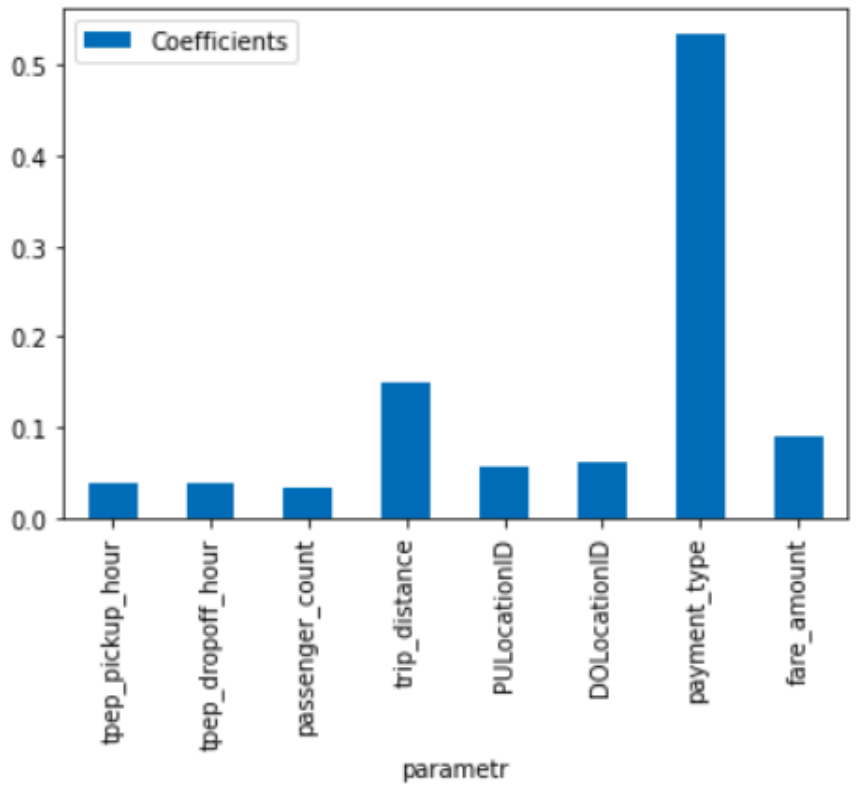
```
+----------+------------+--------------------+--------------------+
|prediction|indexedLabel|         probability|            features|
+----------+------------+--------------------+--------------------+
|       3.0|         2.0|[0.00519287833827...|(8,[0,1,2,3],[1.0...|
|       1.0|         1.0|[0.08355855855855...|(8,[0,2,3,4],[1.0...|
|       0.0|         0.0|[0.95391414141414...|(8,[0,2,3,4],[1.0...|
|       0.0|         0.0|[0.95391414141414...|(8,[0,2,3,4],[1.0...|
|       0.0|         0.0|[0.95391414141414...|(8,[0,2,3,4],[1.0...|
+----------+------------+--------------------+--------------------+
only showing top 5 rows

Test Error = 0.0366007
Test accuracy = 0.9633993387817147
F1 score =  0.9607494569401934
DecisionTreeClassificationModel: uid=DecisionTreeClassifier_00ebcf59252c, depth=5, numNodes=33, numClasses=4, numFeatures=8
```

Klasyfikator oparty o drzewo decyzyjne uzyskał nieznacznie lepsze wyniki od klasyfikatora lasu losowego. F1 score wyniósł 0,9607, natomiast accuracy wyniosło 0,9633. Model składa z 33 wezłów, głębokość drzewa wyniosła 5.

5.2. Regresja dla wartości napiwku

Przed wykorzystaniem metod regresji zaimplementowanych w bibliotece Spark, został wykorzystany ExtraTreeRegressor do sprawdzenia wpływu istontności zmiennych niezależnych na zmienna objaśnianą - cena napiwku.
Z wyniku modelu możemy odczytać, że najbardziej wpływowym parametrem jest rodzaj płatności, a także długość trasy i cena za ten przejazd.



5.2.1. Podział na zbiór testowy i treningowy

Przy użyciu VectorAssembler został zbudowany wektor zmiennych, następnie zbiór został podzielony w proporcjach 0,7 i 0,3 na zbiór treningowy i testowy.

```
+--------------------+-----------+
|            features|tip_amount|
+--------------------+-----------+
|[1.0,2.1,142.0,43...|        0.0|
|[1.0,0.2,238.0,15...|        0.0|
|[1.0,14.7,132.0,1...|       8.65|
|[0.0,10.6,138.0,1...|       6.05|
|[1.0,4.94,68.0,33...|       4.06|
+--------------------+-----------+
only showing top 5 rows
```

5.2.2. Regresja liniowa

Regresja liniowa została zrealizowana za pomocą implementacji regresji liniowej w sparku LinearRegeression. Model został zbudowany z dodatkowymi paramterami maxIter=20, regParam=0.3 oraz elasticNetParam=0.8. Zbudowany model posiada następujące wartości współczynników:

```
Coefficients: [0.0,0.1405669518954252,0.0,0.0,0.0,-1.543530493090399,0.0,0.0]
Intercept: 3.4356773134765928
```

```
+--------------------+-----+------------------+
|            features|label|        prediction|
+--------------------+-----+------------------+
|[0.0,0.0,61.0,61....|  0.0|-1.194914165794604|
|[0.0,0.0,69.0,264...|  0.0|0.3486163272957947|
|[0.0,0.0,74.0,74....|  0.0|0.3486163272957947|
|[0.0,0.0,79.0,79....|  2.0|1.8921468203861938|
|[0.0,0.0,113.0,11...| 7.87|1.8921468203861938|
+--------------------+-----+------------------+
only showing top 5 rows

None
Root Mean Squared Error (RMSE) on test data = 1.17761
R2 on test data = 0.470687
```

Wytrenowany model uzyskał następujące wyniki na zbiorze testowym, błąd pierwiastka średniokwadratowego wyniósł 1,17761. Natomiast wyniki R-kwadrat mówiący o jakości dopasowania modelu wyniósł zaledwie 0,470687.

5.2.3. Regresja za pomocą drzewa decyzyjnego

```
+--------------------+-----+--------------------+
|            features|label|          prediction|
+--------------------+-----+--------------------+
|[0.0,0.0,61.0,61....|  0.0|-6.67284522706209...|
|[0.0,0.0,69.0,264...|  0.0|1.054818940328892...|
|[0.0,0.0,74.0,74....|  0.0|7.247373072768338E-4|
|[0.0,0.0,79.0,79....|  2.0|  0.4123916464025153|
|[0.0,0.0,113.0,11...| 7.87|  0.4123916464025153|
+--------------------+-----+--------------------+
only showing top 5 rows

Root Mean Squared Error (RMSE) on test data = 0.926884
R2 on test data = 0.672087
```

W przypadku drzewa decyzyjnego pierwiastek błędu średniokwadratowego zmniejszył się i wyniósł 0,926884, znacznej poprawie uległ także R-kwadrat i wyniósł 0,67.

5.2.4. Regresja za pomocą Random Forest

```
+--------------------+-----+--------------------+
|            features|label|          prediction|
+--------------------+-----+--------------------+
|[0.0,0.0,61.0,61....|  0.0|0.06565098596434743|
|[0.0,0.0,69.0,264...|  0.0|0.03629254639504005|
|[0.0,0.0,74.0,74....|  0.0|0.06769661990398554|
|[0.0,0.0,79.0,79....|  2.0| 1.2260034097289931|
|[0.0,0.0,113.0,11...| 7.87|  1.5384054580607 63|
+--------------------+-----+--------------------+
only showing top 5 rows

Root Mean Squared Error (RMSE) on test data = 0.943829
R2 on test data = 0.659988
```

W przypadku zastosowania regresji za pomocą lasu losowego wyniki nieznacznie się pogorszyły w stosunku do modelu na podstawie drzewa decyzyjnego. Pierwiastek błędu średniokwadratowego wyniósł 0,94, natomiast wynik R-kwadrat wyniósł 0,6599.

6. Wnioski

- Z modeli regresji dla problemu kwoty napiwku najlepsze wyniki dawał model oparty o drzewo decyzyjne, natomiast najgorsze model regresji liniowej
- W przypadku klasyfikacji typu płatności najlepsze wyniki dawał klasyfikator drzewa decyzyjnego, natomiast najgorsze wyniki osiągał model regresji logistycznej