Slovenská technická univerzita v Bratislave
Fakulta elektrotechniky a informatiky

Neurónové siete

Zadanie č.2

2021 **Jakub Jančík**

Obsah

Zoznam obrázkov

1 Analýza, príprava a spracovanie

Naším cieľom pre toto zadanie bude predpovedať hlasnosť (loudness) piesne. Dostali sme Api Spotify dáta , ktoré obsahujú následujúce stĺpce.

```
dfTrain.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44776 entries, 0 to 44775
Data columns (total 27 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   id                    44776 non-null  object
 1   artist_id             44776 non-null  object
 2   artist                44776 non-null  object
 3   name                  44776 non-null  object
 4   popularity            44776 non-null  int64
 5   release_date          44776 non-null  object
 6   duration_ms           44776 non-null  int64
 7   explicit              44776 non-null  bool
 8   danceability          44776 non-null  float64
 9   energy                44776 non-null  float64
 10  key                   44776 non-null  int64
 11  loudness              44776 non-null  float64
 12  mode                  44776 non-null  int64
 13  speechiness           44776 non-null  float64
 14  acousticness          44776 non-null  float64
 15  instrumentalness      44776 non-null  float64
 16  liveness              44776 non-null  float64
 17  valence               44776 non-null  float64
 18  tempo                 44776 non-null  float64
 19  artist_genres         44776 non-null  object
 20  artist_followers      44775 non-null  float64
 21  url                   44776 non-null  object
 22  playlist_id           44776 non-null  object
 23  playlist_description  30974 non-null  object
 24  playlist_name         44755 non-null  object
 25  playlist_url          44776 non-null  object
 26  query                 44776 non-null  object
dtypes: bool(1), float64(10), int64(4), object(12)
```

Už teraz môžeme vidieť, ktoré stĺpce budeme potrebovať do ďalšieho spracovania. Napríklad úplne môžeme vylučiť stĺce ktoré majú id, názvy, duration_ms atd.

Po úprave máme následovné stĺpce, ktoré budeme používať pri predpovedi hlučnosti piesni.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44776 entries, 0 to 44775
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   danceability      44776 non-null  float64
 1   energy            44776 non-null  float64
 2   key               44776 non-null  int64
 3   loudness          44776 non-null  float64
 4   mode              44776 non-null  int64
 5   speechiness       44776 non-null  float64
 6   acousticness      44776 non-null  float64
 7   instrumentalness  44776 non-null  float64
 8   liveness          44776 non-null  float64
 9   valence           44776 non-null  float64
 10  tempo             44776 non-null  float64
dtypes: float64(9), int64(2)
```

```
corrMatrix = dfTrain[properties].corr()
sn.heatmap(corrMatrix, annot=True)
plt.show()
```

Listing 1: Kód pre vytvorenie korelačnej matice.

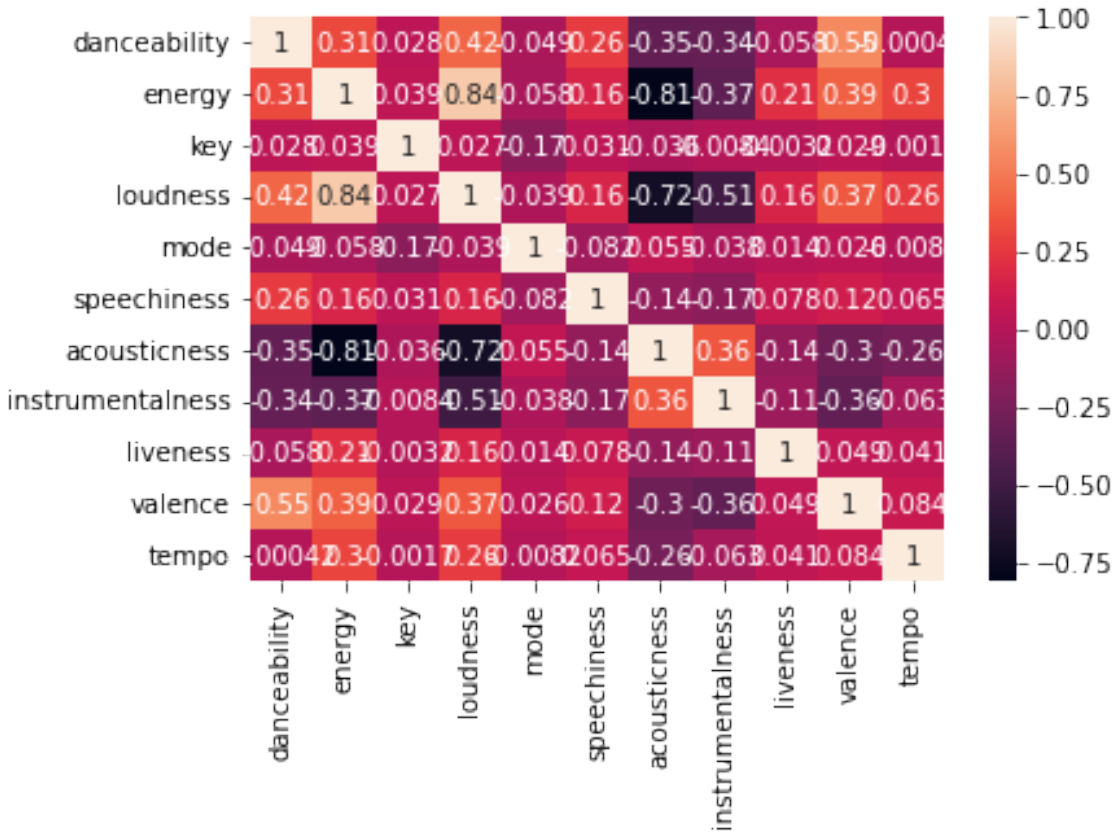
Výpis nulových hodnôt, ktoré sú v našom data frame.

```
dfTrain.isna().sum()

id                          0
artist_id                   0
artist                      0
name                        0
popularity                  0
release_date                0
duration_ms                 0
explicit                    0
danceability                0
energy                      0
key                         0
loudness                    0
mode                        0
speechiness                 0
```

```
acousticness                0
instrumentalness            0
liveness                    0
valence                     0
tempo                       0
artist_genres               0
artist_followers            1
url                         0
playlist_id                 0
playlist_description    13802
playlist_name              21
playlist_url                0
query                       0
```

Obr. 1: Korelačná matica.



Červené až svetlé odtiene predstavujú pozitívnu koreláciu, zatiaľ čo tmavšie odtiene predstavujú negatívnu koreláciu.

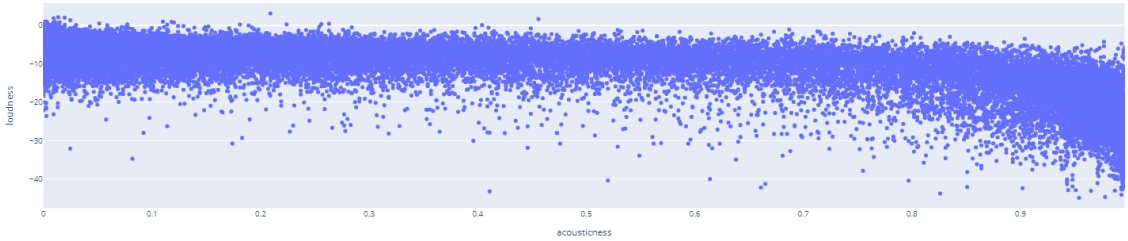
„Energy“ a „Loudness“ majú najvyššiu koreláciu, a to pozitívnu, čo neprekvapuje.

„Loudness“ a „Acousticness“ majú vysoko korelovaný inverzný vzťah, čo tiež

dáva úplný zmysel. Čím viac je skladba akustická, tým menej hlasná býva.

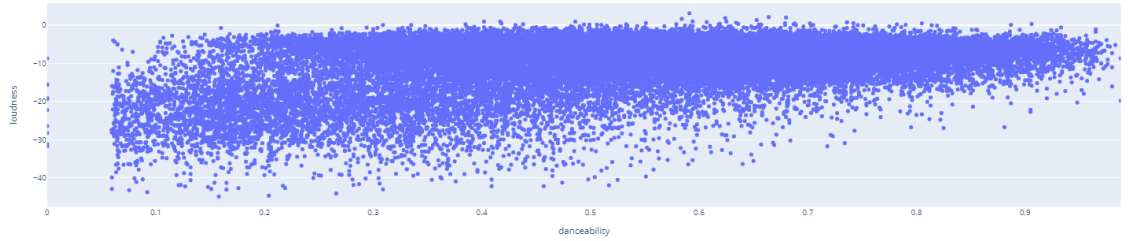
1.1 Analýza stĺpcov

Obr. 2: Závislosť loudness od acousticness.

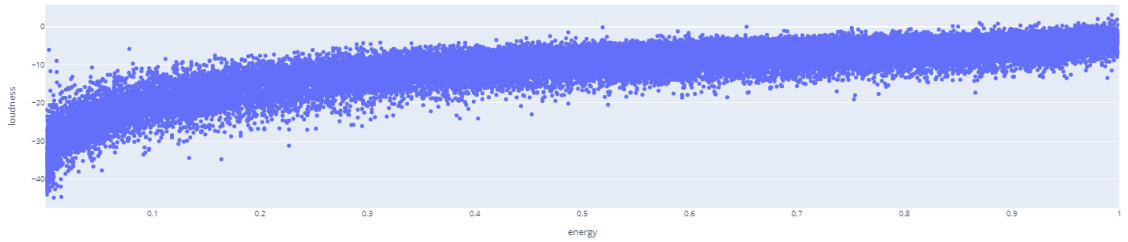


Na obr. 2 vidíme, že malá akustickosť = velká hlučnosť. Ak viac zanalyzujeme tento graf, tak pri vyššej äkustike"piesni sa nachadzajú piesne, ktoré nie sú až tak hlučné.

Obr. 3: Závislosť loudness od danceability.

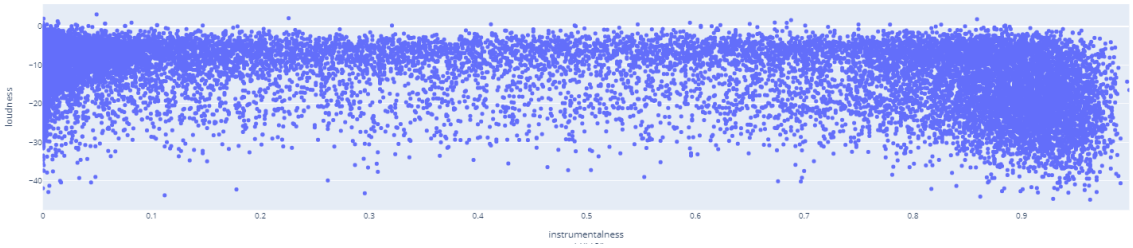


Obr. 4: Závislosť loudness od energy.



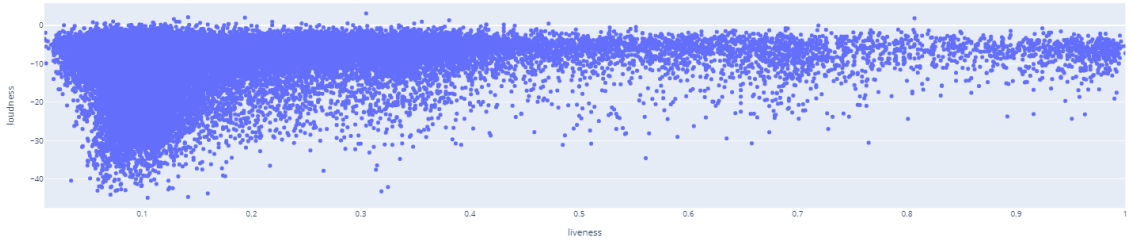
Na obr.4 vidíme jasnú koreláciu medzi hlučnosťou a energiou. Čím vyššia energia, tým vyššia hlučnosť.

Obr. 5: Závislosť loudness od instrumetalness.



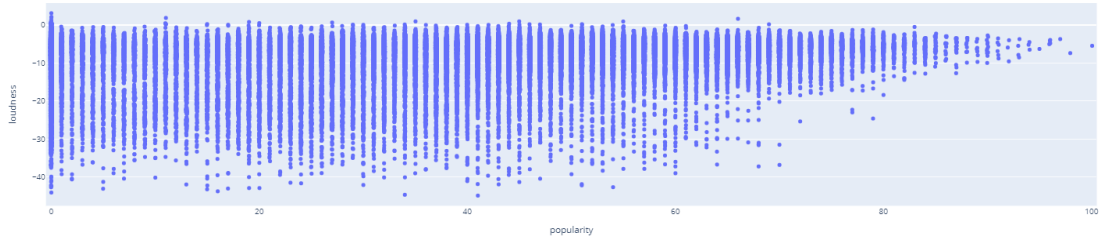
Na obr.5 sa zdá, že ak daná pieseň obsahuje veľa instumentalness, tak narastá aj počet menej hlučných piesni. Ale v tomto prípade neviem úplne presne určiť.

Obr. 6: Závislosť loudness od liveness.



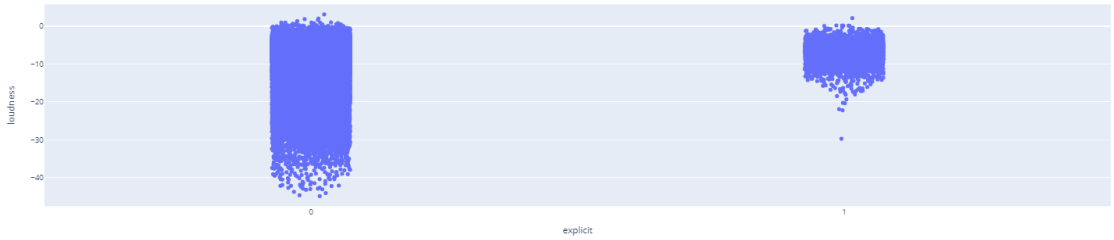
Na obr. 6 môžeme pozorovať, že ak piesne majú nízky liveness, tak je vačší počet piesni ktoré nie sú hlučné.

Obr. 7: Závislosť loudness od popularity.



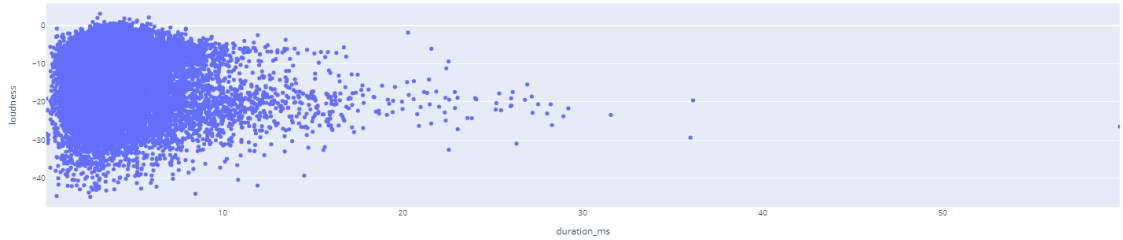
Na obr. 7 vidíme najviac populárne (od cca 70 po 100) piesne, ktoré sú prevažné hlučnejšie.

Obr. 8: Závislosť loudness od explicit.



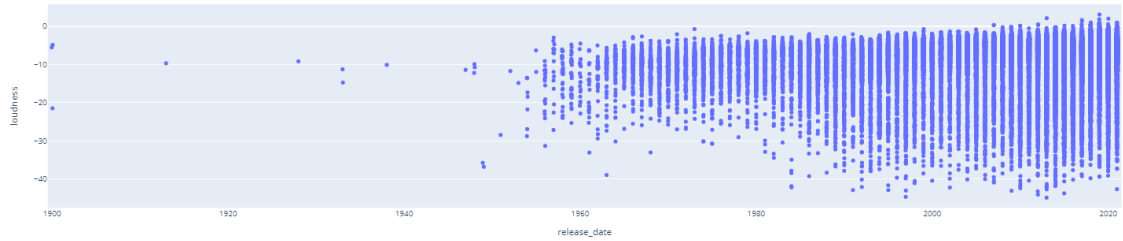
Na Obr. 8 x os sú čísla, kde 0 reprezentuje. že v piesni s nepoužívajú žiadné nadávky. Číslo 1 je pravý opak a to kde sa nadáva, alebo používajú nevhodné slová pre deti. Na tomto grafe je jasne vidieť, že piesne, ktoré sú **hlasnejšie** tak tam **sa aj nadáva**. Kde v prípade, že pisene niesu explicitné, sú piesne rôznej hlasitosti - kde sa nedá niečo presnejšie určiť.

Obr. 9: Závislosť loudness od duration(m).



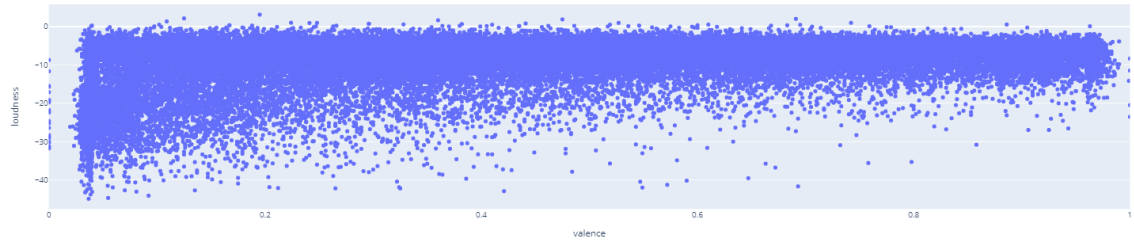
Na Obr.9 sme nevedeli nájsť nejakú zaujimavú vlastnosť pre stĺpec duration.

Obr. 10: Závislosť loudness od release date.



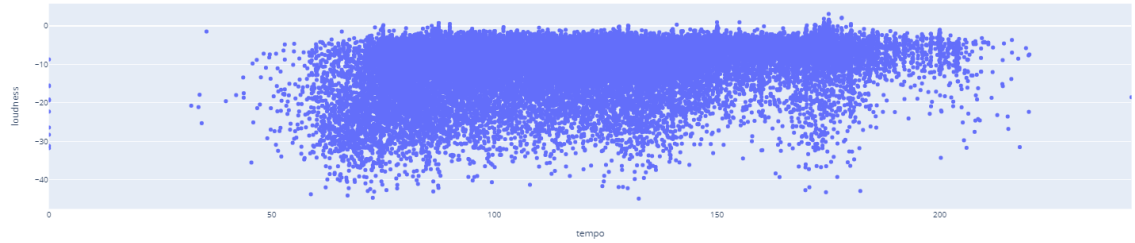
Na Obr. 10 vidíme závislosť hlučnosti od roku vydania danej piesne. Po analýze si vieme určit napríklad že **od roku 1955 po rok 1980 sa vydávali hlučnejšie piesne** . Po rok 1980 je hlučnosť rôznorodá.

Obr. 11: Závislosť loudness od valence.



Obr. 12: Závislosť loudness od speechiness.

Obr. 13: Závislosť loudness od tempo.



1.2 Záver analýzy

Vzhľadom na dané porovnavania medzi rôznymi stĺpcami sme vedeli zistiť alebo určiť hraničné dáta pri ďalšom spracovaní. V ďalšom postupe sa budeme snažiť zistíť pravdepodobnosť hlučnosti piesni. Podľa analýzy si vyberieme dané stĺpce s určitými hodnotami, ktoré nam môžu viac napovedať o predpovedi.

Stringové hodnoty sme upravili na číselné. Explicit (0=false,1=true). Release date sme vybrali z dátumu len rok. A žárner nevieme nájsť spôsob pre zmenu stĺpca na číslo. Ale vymysleli sme štýlom že nám vratí celý dataframe s výberom daného žánra.

```
mask = dfTrain.artist_genres.apply(lambda x: 'metal' in x)
df1 = dfTrain[mask]
```

Listing 2: vrati dataframe s uvedeným žánrom.

Obe množiniy sme vhodne normalizovali od 0 po 1. Bohužial, to sme aplikovali až po celej analýze, takže všetky grafy obsahujú pôvodne nedotknuté hodnoty.

```
#NORMALIZOVANIE
min_max_scaler = MinMaxScaler()
dfTrain[properties] = min_max_scaler.fit_transform(dfTrain[properties
    ])
```

```
min_max_scaler = MinMaxScaler()
dfTest[properties] = min_max_scaler.fit_transform(dfTest[properties])
```

Listing 3: normalizovanie oboch mnozin.

2 SVM regresor

2.1 Trénovanie SVM a vyhodnotenie

Pre trénovanie sme vybrali tieto stĺpce: ['popularity', 'explicit', 'danceability', 'energy', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence']

Hyperparameter sme nastavili pre tento model C=100 a gamma = scale.

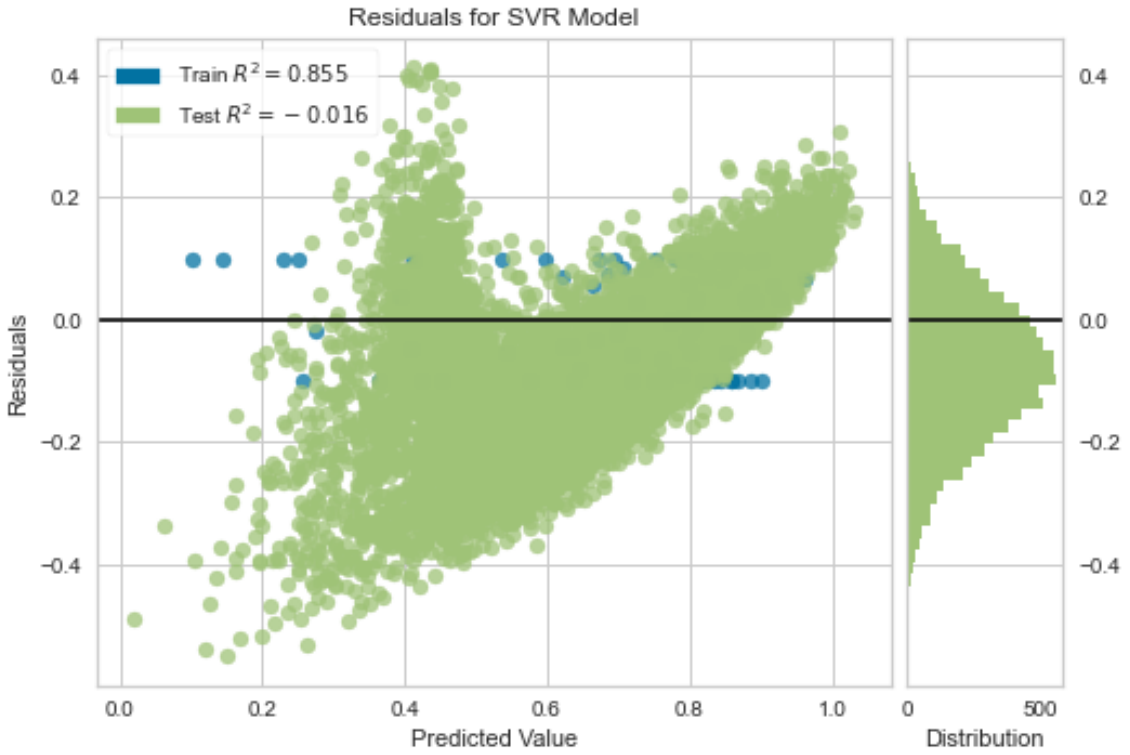
```
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size
    =0.2, random_state=12)
clf = SVR(kernel='rbf', C=100, gamma='scale',)
clf.fit(X_train, y_train)

crossValScore = cross_val_score(clf, X_train, y_train, cv=5)
print("%0.2f accuracy with a standard deviation of %0.2f" % (
    crossValScore.mean(), crossValScore.std()))
```

Listing 4: trenovanie SVM regresor.

MSE: 0.006249781792419973 r2 score: 0.8550404475671756 0.16 accuracy with a standard deviation of 0.45

Obr. 14: Residuals plot.



Test R2 skore máme ako zápornú hodnotu, to je keď vybraný model nesleduje trend údajov, čo vedie k horšiemu prispôsobeniu ako horizontálna čiara. Zvyčajne je to prípad, keď existujú obmedzenia buď na priesečník, alebo sklon lineárnej regresnej priamky.

2.2 Grid Search

Potom špecifikujeme hyperparametre, ktoré sa snažíme preskúmať. Pri použití jadra rbf SVR sa použijú tri hyperparametre C, epsilon a gama. Každému môžeme dať na výber niekoľko hodnôt. Nižšie sú moje náhodne vybrané hodnoty.

```
grid = GridSearchCV(

estimator=SVR(kernel='rbf'),
        param_grid={
            'C': [1.1, 5.4, 170, 100, 300, 500, 600, 1001],
            'epsilon': [0.0003, 0.007, 0.0109, 0.019, 0.14, 0.05,
   0.05, 0.5, 1, 8, 0.2, 3, 2, 7],
        },
        cv=5, scoring='neg_mean_squared_error', verbose=0, n_jobs=-1)
```

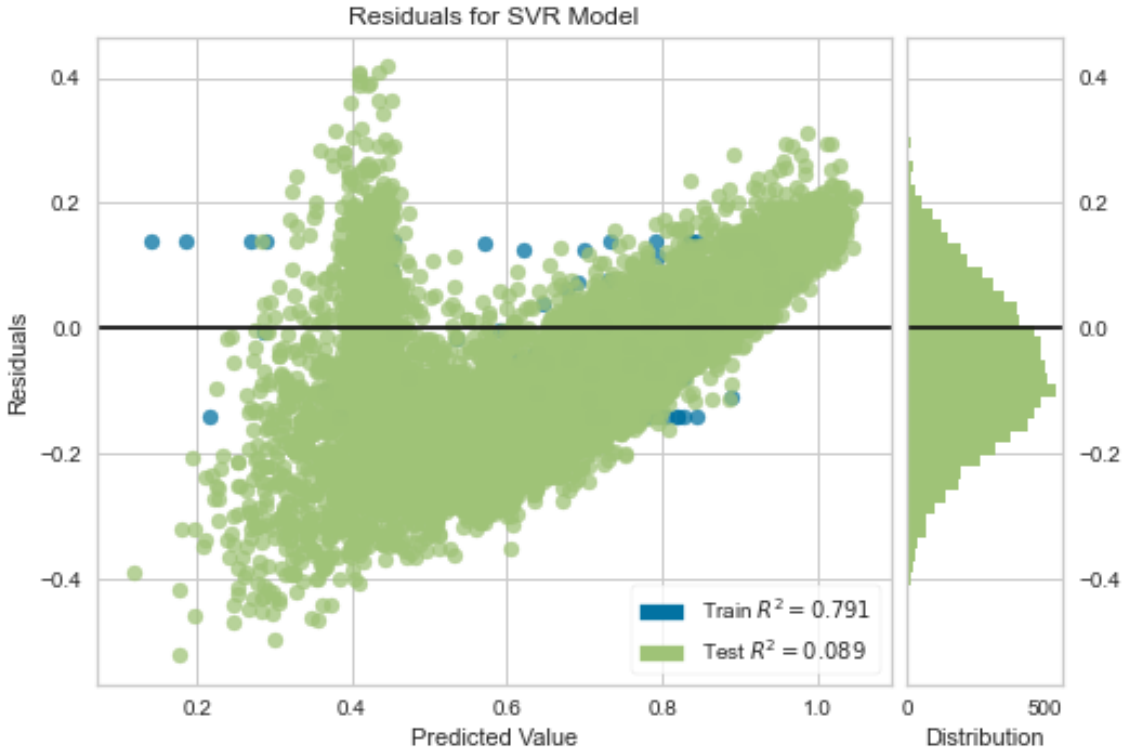
```
grid.fit(X_train,y_train)
```

Listing 5: Grid search kod.

GridSearch nám vyhodnotil ako najlepšie parametre: best parameters are: 'C': 600, 'epsilon': 0.14, 'gamma': 0.001

Keďže vyhľadávanie v mriežke skúša všetky mož0né kombinácie, stáva sa výpočtovo nákladnou metódou. Definovali sme vyhľadávanie v mriežke a na jednoduchom príklade sme preskúmali, ako to funguje.

Obr. 15: Residuals plot after grid search best parameters.



Vysledky po tom ako sme nastavili najlepšie parametre z naších náhodne vypísaných hodnôt. SVR(C=600, epsilon=0.14) MSE: 0.008994352328245003 r2 score: 0.7913819504055551 0.32 accuracy with a standard deviation of 0.34

Naše najlepšie parametre nezlepšili náš model. Tento model nam dal uspešnosť 79 percent. Pričom náš pôvodný a zatiaľ najlepší 85 percent. Existujú aj iné optimalizačné metódy, ktoré sa líšia zložitosťou a účinnosťou.

2.3 Súborové učenie

Bagging a „náhodné lesy“ sú „bagovacie“ algoritmy, ktorých cieľom je znížiť zložitosť modelov, ktoré presahujú trénovacie údaje. Na rozdiel od toho, boosting je prístup na zvýšenie zložitosti modelov, ktoré trpia vysokou zaujatosťou, teda modelov, ktoré nezodpovedajú tréningovým údajom.

2.3.1 Boosting

Pre naše súborové učenie - boosting sme si vybrali GradientBoostingRegressor. Podporuje množstvo rôznych stratových funkcií pre regresiu, ktoré možno špecifikovať pomocou argumentu strata; predvolená funkcia straty pre regresiu je druhá mocnina chyby ('squared_error') [1].

```
from sklearn.ensemble import GradientBoostingRegressor

est = GradientBoostingRegressor(n_estimators=100, learning_rate=0.1,
    max_depth=1, random_state=0,loss='squared_error').fit(X_train,
    y_train)
```

Listing 6: Suborove ucenie - boosting s parametrami.

Môžeme predpovedať testovacie údaje a skontrolovať chybovosť nasledovne.

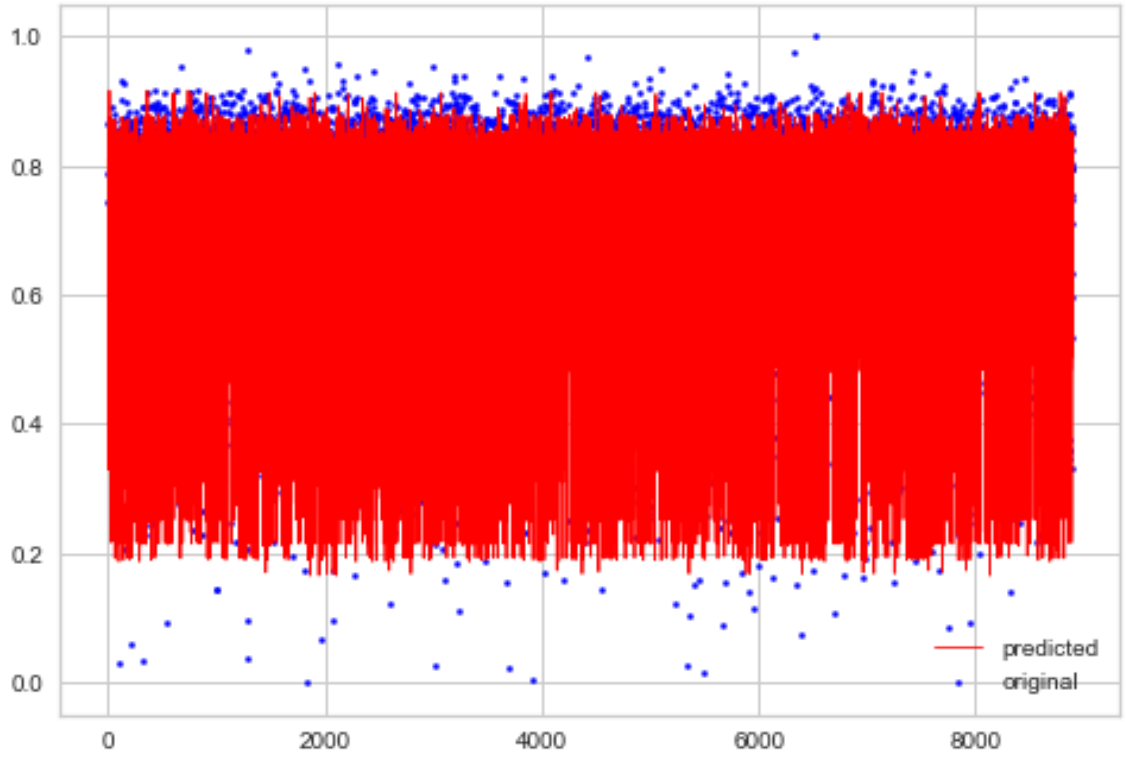
```
ypred = gbr.predict(xtest)
mse = mean_squared_error(ytest,ypred)
score = r2_score(y_test, ypred)

print("MSE: %.2f" % mse)
print("R2 score: %.2f" % score)
MSE: 0.02
R2 score: -0.03
```

Listing 7: Suborove ucenie - boosting.

Nakoniec vizualizujeme pôvodné a predpokladané hodnoty v grafe.

Obr. 16: Plot GradientBoostingRegressor.



2.3.2 Bagging

```
model = BaggingRegressor(base_estimator=DecisionTreeRegressor(
    max_depth=1),n_estimators=100, random_state=0)
model.fit(X_train, y_train)
```

Listing 8: Bagging Regressor model.

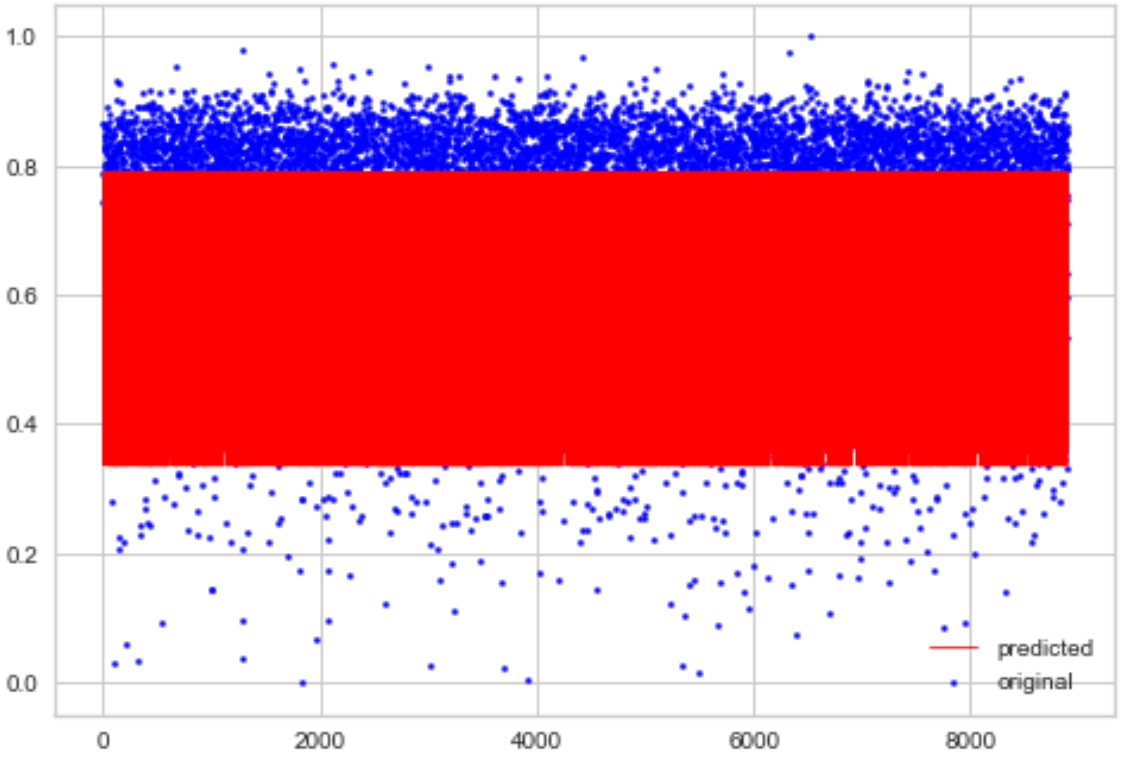
Môžeme predpovedať testovacie údaje a skontrolovať chybovosť nasledovne.

```
ypred = gbr.predict(xtest)
mse = mean_squared_error(ytest,ypred)
score = r2_score(y_test, ypred)

print("MSE: %.2f" % mse)
print("R2 score: %.2f" % score)
MSE: 0.03
R2 score: -0.19
```

Listing 9: Suborove ucenie - bagging.

Obr. 17: Plot BaggingRegressor.



2.4 Záver

Cieľom bolo predpovedať hlasitosť akejkoľvek skladby. Vykonala sa prieskumná analýza údajov na odvodenie poznatkov z údajov. V tomto zadaní sme vyhodnotili 3 modely. SVM, Bagging a Boosting. Náš najlepší model SVM, dosiahol najlepšie výsledky MSE = 0.0062, R2 = 85percent. Pričom Bagging a Boosting nám absolútne nevyšli a nezostrojili sme úspešný model, preto nám výsledky ako R2 vyšli negatívn (R2 -0.03 a -0.19) pričom MSE bola okolo 0.02.

Literatúra

[1] scikit-learn developers (BSD License). *Gradient Tree Boosting.* `https://scikit-learn.org/stable/modules/ensemble.html#gradient-tree-boosting`.