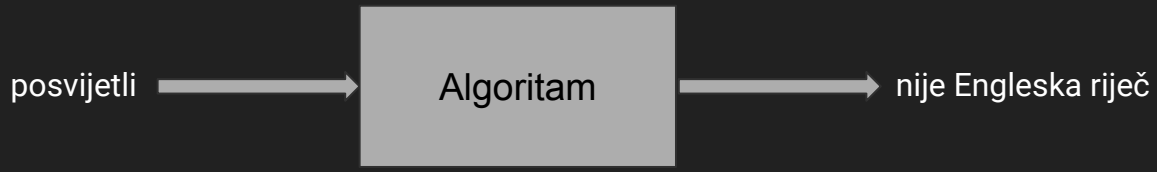
PRIMJER KORIŠTENJA STROJNOG UČENJA U NLP-u

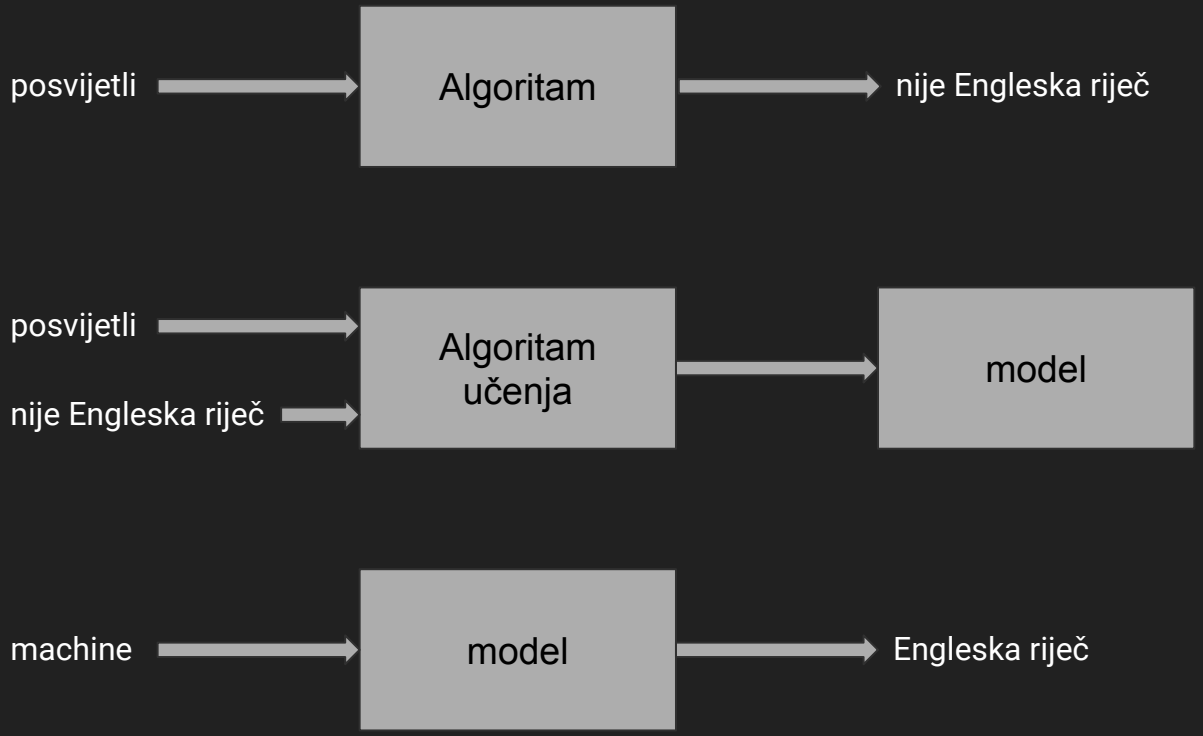
Mario Kučić


posvijetli
Algoritam
nije Engleska riječ


posvijetli nije eng
spektakularniju nije eng
monogamist engl
prijavu nije eng
fotografiravši nije eng


posvijetli
Algoritam
nije Engleska riječ
posvijetli
nije Engleska riječ
Algoritam učenja
model

posvijetli	nije eng
spektakularniju	nije eng
monogamist	engl
prijavu	nije eng
fotografiravši	nije eng



```
podaci = pd.read_csv('podaci.csv')
```

učitavanje podataka

učitavanje podataka

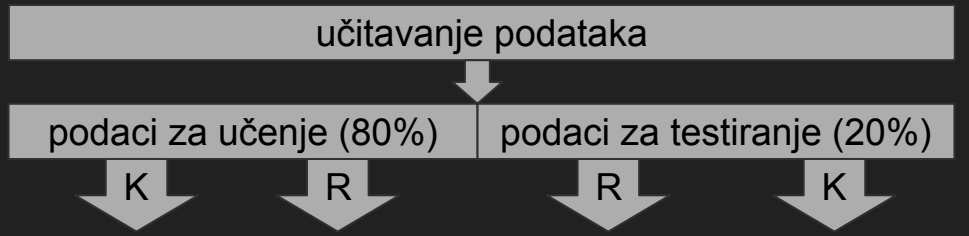
```
podaci = pd.read_csv('podaci.csv')

podaci.head()
```

	rijec	klasa
0	posvijetli	0
1	spektakularniju	0
2	monogamist	1
3	prijavu	0
4	fotografiravši	0

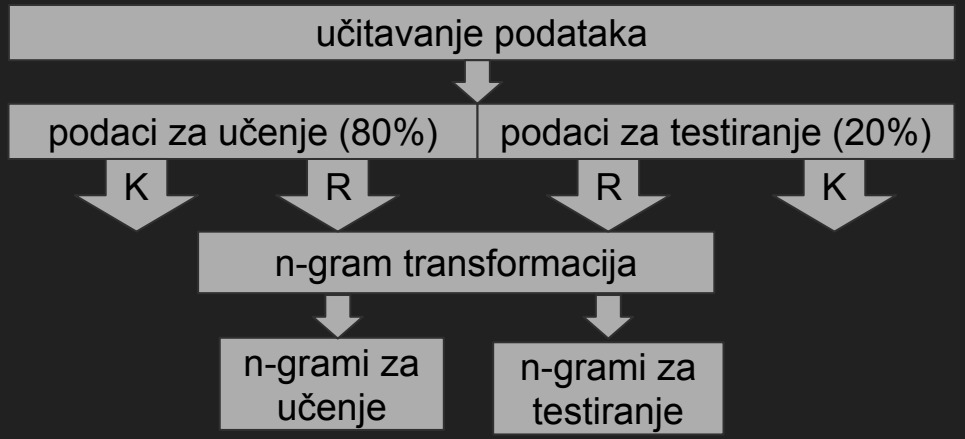
```
podaci = pd.read_csv('podaci.csv')
```

učitavanje podataka

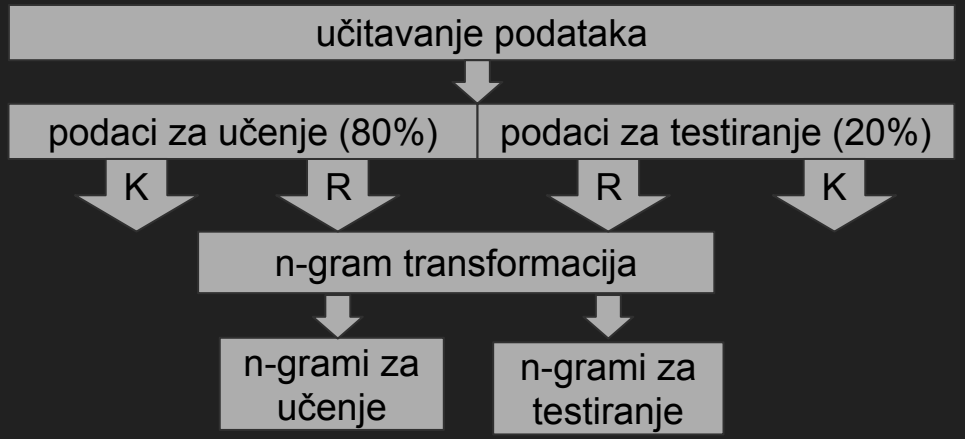
```
podaci = pd.read_csv('podaci.csv')

rijeci_ucenje, rijeci_test, klase_ucenje, klase_test =\
    train_test_split(podaci['rijec'], podaci['klasa'],
    test_size=0.2)
```

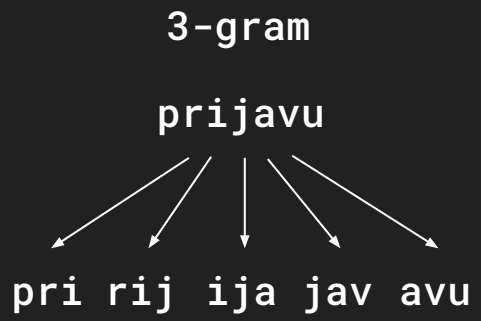


```
podaci = pd.read_csv('podaci.csv')

rijeci_ucenje, rijeci_test, klase_ucenje, klase_test =\
    train_test_split(podaci['rijec'], podaci['klasa'],
    test_size=0.2)

transformator = CountVectorizer(
      ngram_range=(1, 3), analyzer='char')

ngram_ucenje = transformator.fit_transform(
      rijeci_ucenje)

ngram_test = transformator.transform(rijeci_test)
```

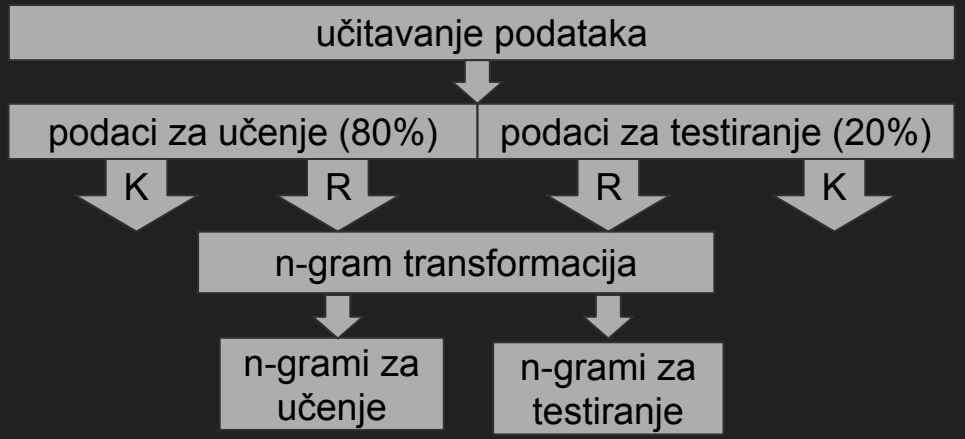


```
podaci = pd.read_csv('podaci.csv')

rijeci_ucenje, rijeci_test, klase_ucenje, klase_test =\
    train_test_split(podaci['rijec'], podaci['klasa'],
    test_size=0.2)

transformator = CountVectorizer(
      ngram_range=(1, 3), analyzer='char')

ngram_ucenje = transformator.fit_transform(
      rijeci_ucenje)

ngram_test = transformator.transform(rijeci_test)
```

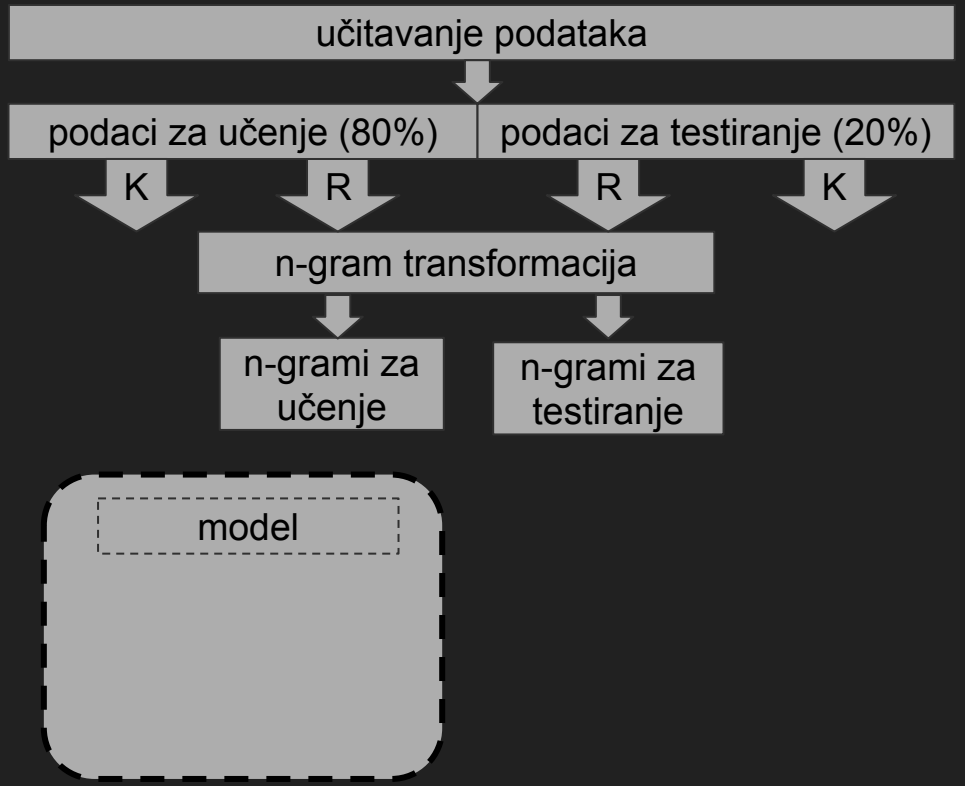




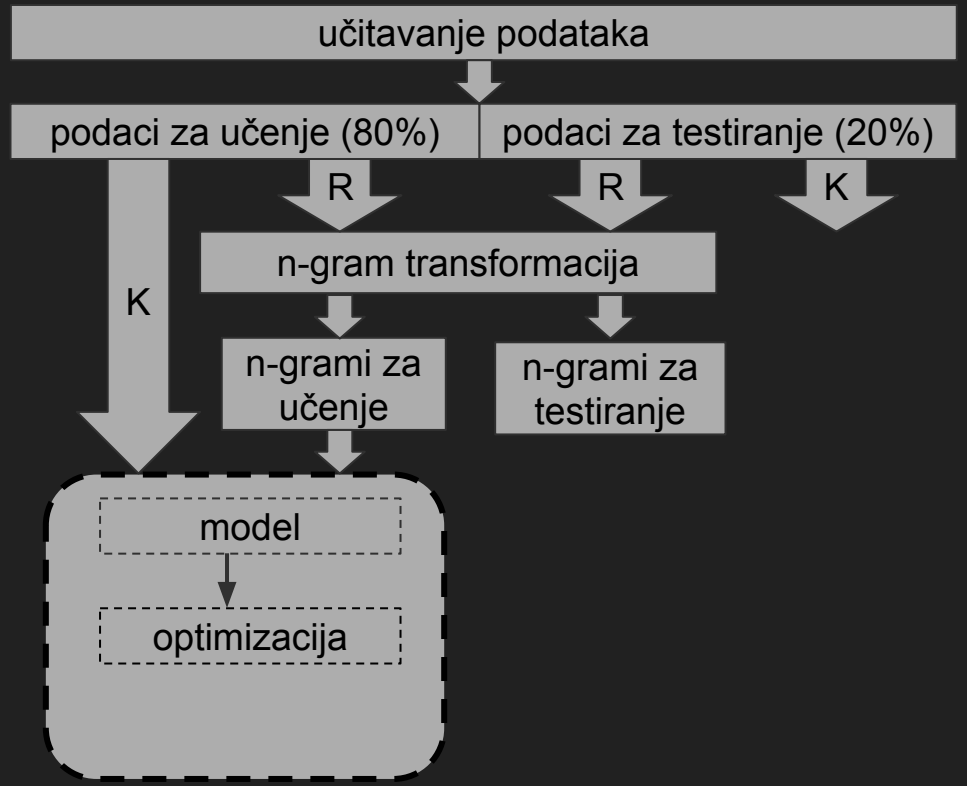
```
podaci = pd.read_csv('podaci.csv')

rijeci_ucenje, rijeci_test, klase_ucenje, klase_test =\
    train_test_split(podaci['rijec'], podaci['klasa'],
    test_size=0.2)

transformator = CountVectorizer(
      ngram_range=(1, 3), analyzer='char')

ngram_ucenje = transformator.fit_transform(
      rijeci_ucenje)

ngram_test = transformator.transform(rijeci_test)
```



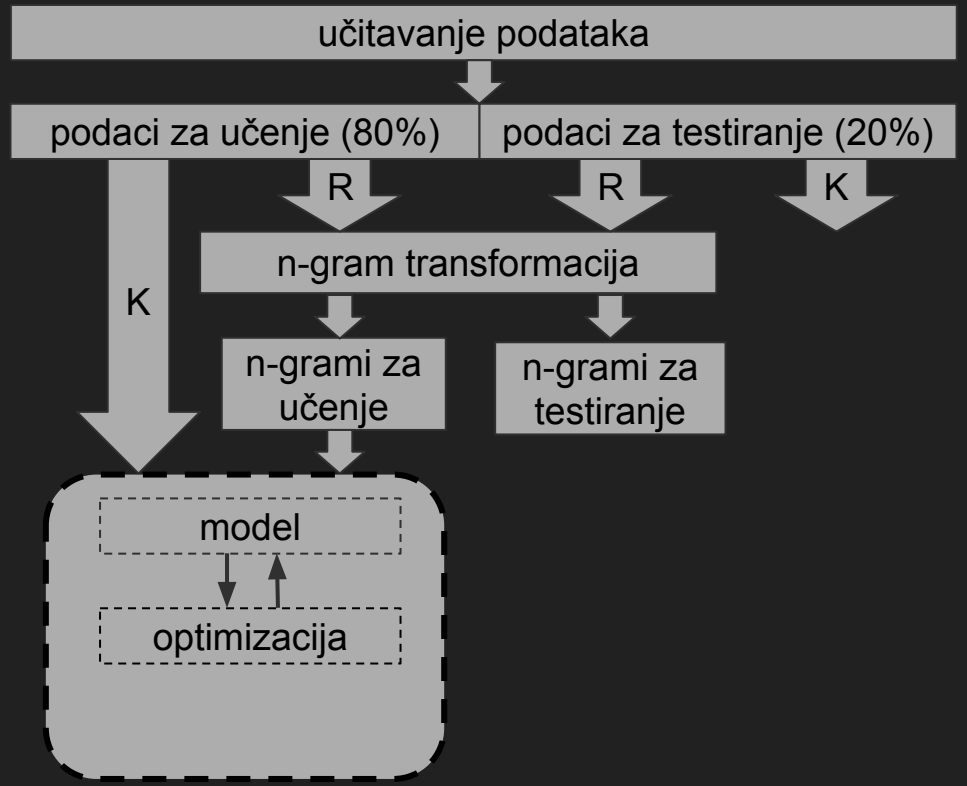
```python
podaci = pd.read_csv('podaci.csv')

rijeci_ucenje, rijeci_test, klase_ucenje, klase_test =\
    train_test_split(podaci['rijec'], podaci['klasa'],
    test_size=0.2)

transformator = CountVectorizer(
      ngram_range=(1, 3), analyzer='char')

ngram_ucenje = transformator.fit_transform(
      rijeci_ucenje)

ngram_test = transformator.transform(rijeci_test)

model = RandomForestClassifier(
      n_estimators=100, max_depth=3)
```



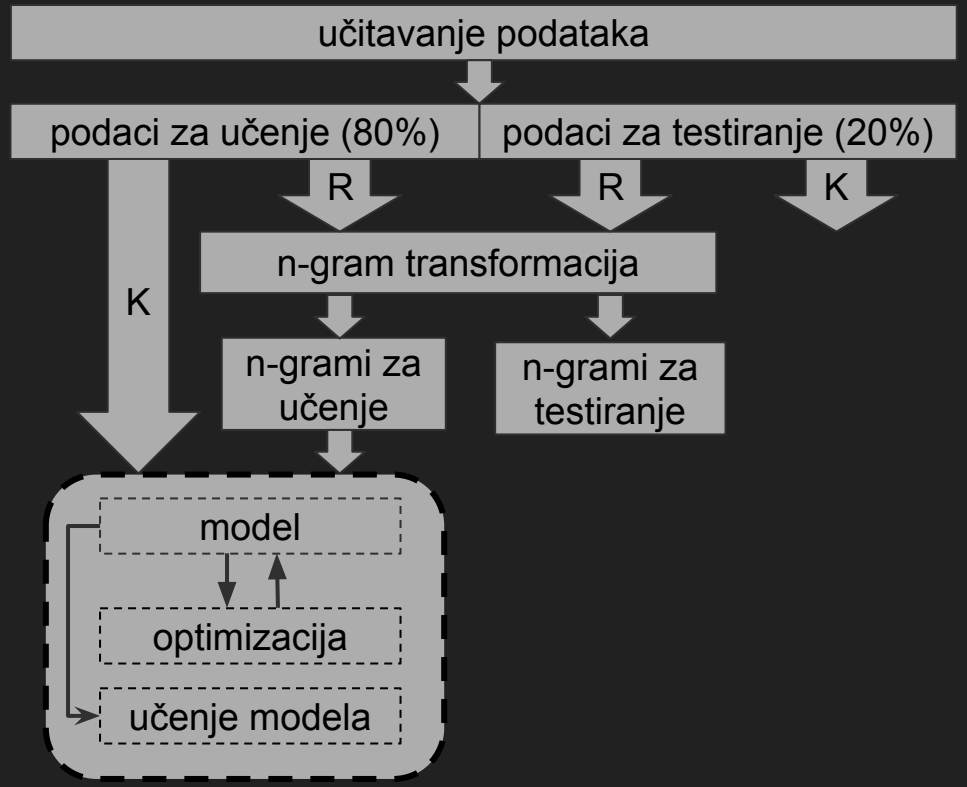
```python
podaci = pd.read_csv('podaci.csv')

rijeci_ucenje, rijeci_test, klase_ucenje, klase_test =\
    train_test_split(podaci['rijec'], podaci['klasa'],
    test_size=0.2)

transformator = CountVectorizer(
      ngram_range=(1, 3), analyzer='char')

ngram_ucenje = transformator.fit_transform(
      rijeci_ucenje)

ngram_test = transformator.transform(rijeci_test)

model = RandomForestClassifier(
      n_estimators=100, max_depth=3)

rezultati = cross_val_score(
      model, ngram_ucenje, klase_ucenje)

print(f'Avg F1 {rezultati.mean():.3f}') # Avg F1 0.870
```



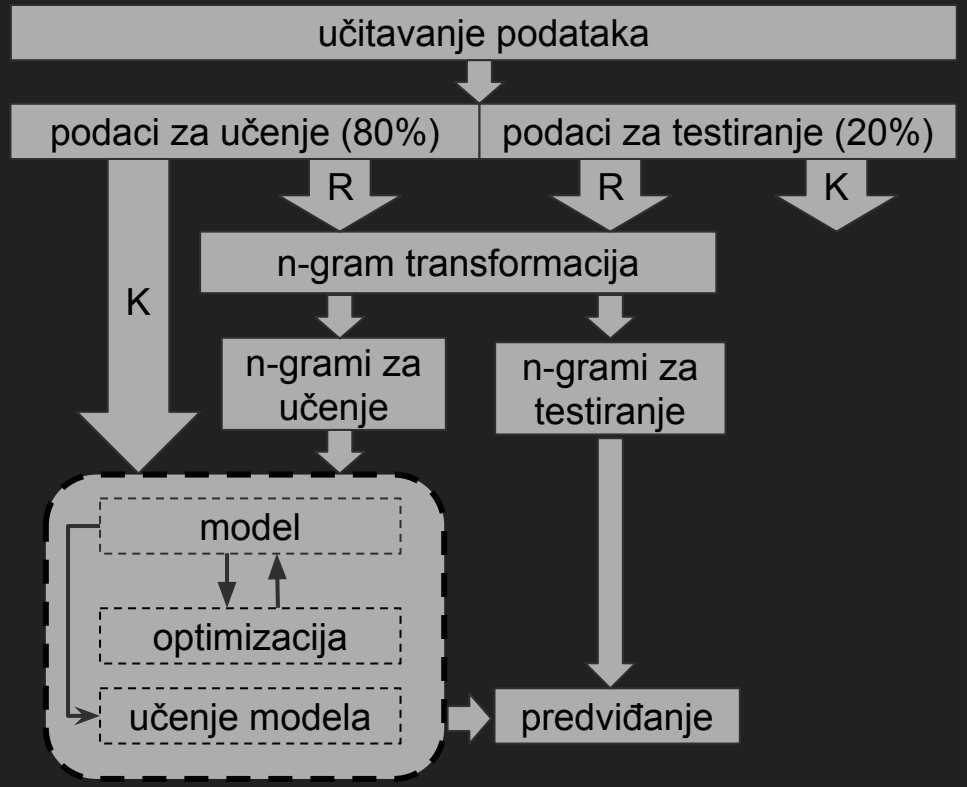
```python
podaci = pd.read_csv('podaci.csv')

rijeci_ucenje, rijeci_test, klase_ucenje, klase_test =\
    train_test_split(podaci['rijec'], podaci['klasa'],
    test_size=0.2)

transformator = CountVectorizer(
      ngram_range=(1, 3), analyzer='char')

ngram_ucenje = transformator.fit_transform(
      rijeci_ucenje)

ngram_test = transformator.transform(rijeci_test)

model = RandomForestClassifier(
      n_estimators=100, max_depth=3)

rezultati = cross_val_score(
      model, ngram_ucenje, klase_ucenje)

print(f'Avg F1 {rezultati.mean():.3f}') # Avg F1 0.870
```



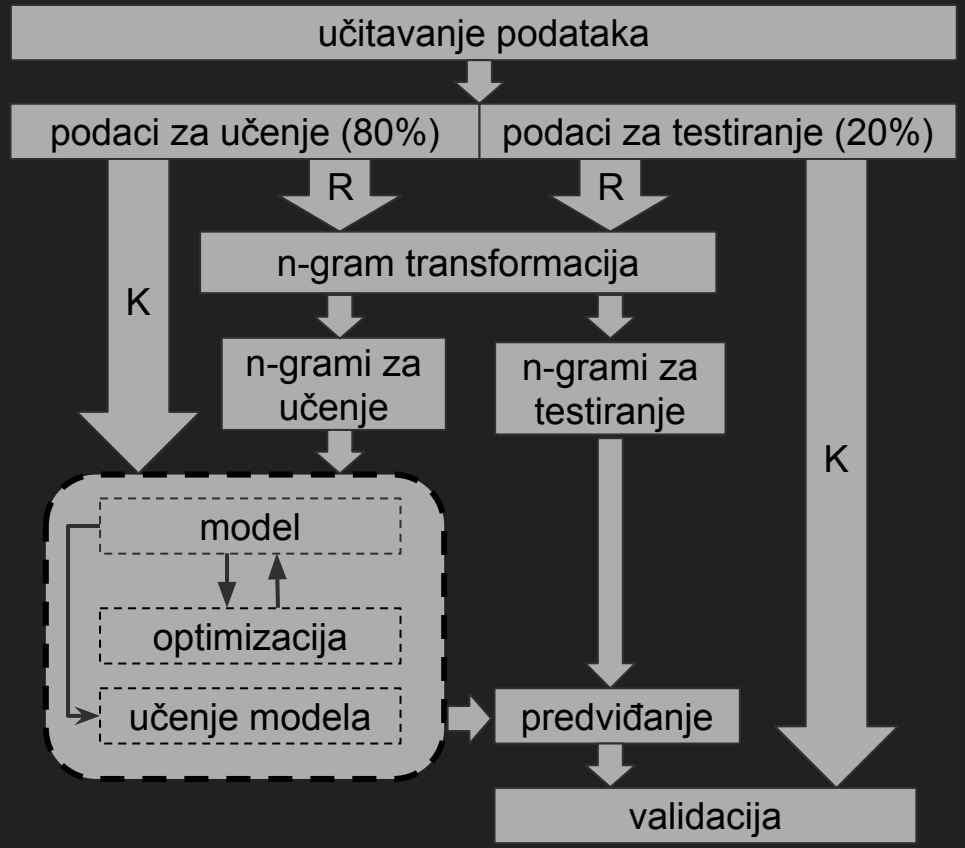
```python
podaci = pd.read_csv('podaci.csv')

rijeci_ucenje, rijeci_test, klase_ucenje, klase_test =\
    train_test_split(podaci['rijec'], podaci['klasa'],
    test_size=0.2)

transformator = CountVectorizer(
      ngram_range=(1, 3), analyzer='char')

ngram_ucenje = transformator.fit_transform(
      rijeci_ucenje)

ngram_test = transformator.transform(rijeci_test)

model = RandomForestClassifier(
      n_estimators=100, max_depth=3)

rezultati = cross_val_score(
      model, ngram_ucenje, klase_ucenje)

print(f'Avg F1 {rezultati.mean():.3f}') # Avg F1 0.870

model.fit(ngram_ucenje, klase_ucenje)
```



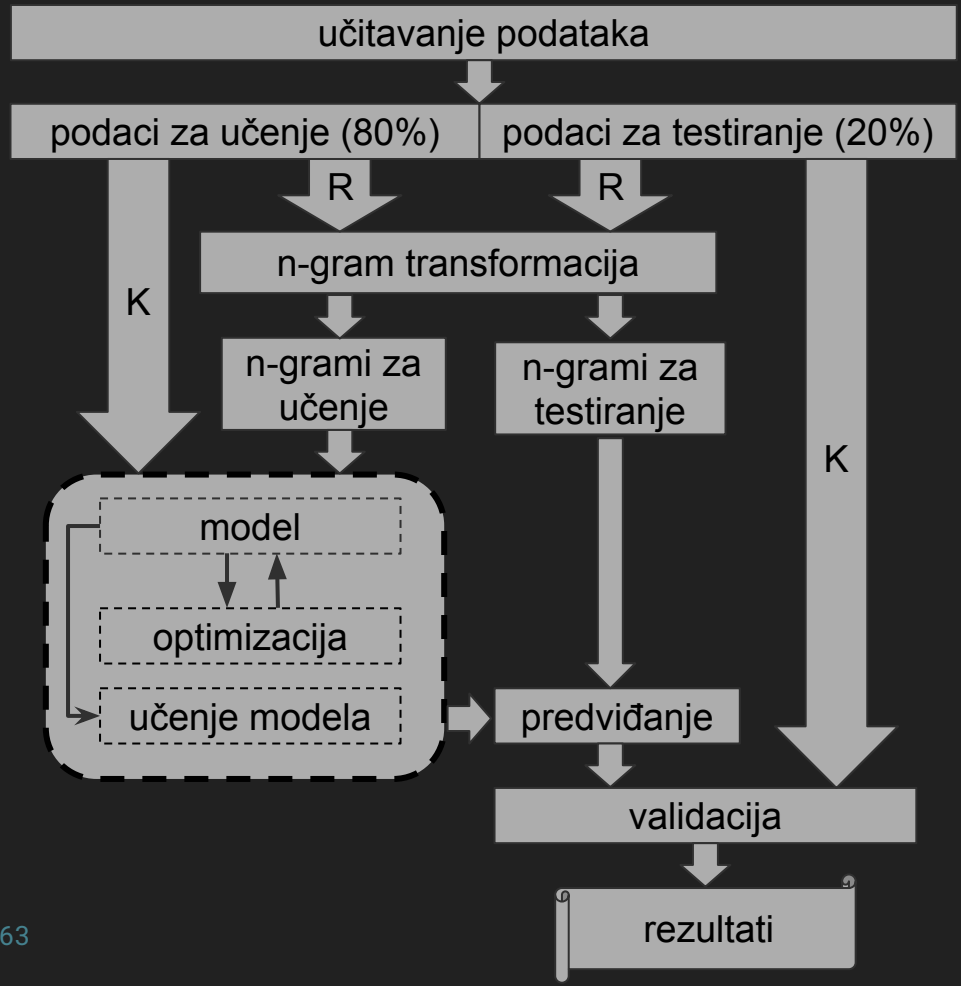
```python
podaci = pd.read_csv('podaci.csv')

rijeci_ucenje, rijeci_test, klase_ucenje, klase_test =\
    train_test_split(podaci['rijec'], podaci['klasa'],
    test_size=0.2)

transformator = CountVectorizer(
      ngram_range=(1, 3), analyzer='char')

ngram_ucenje = transformator.fit_transform(
      rijeci_ucenje)

ngram_test = transformator.transform(rijeci_test)

model = RandomForestClassifier(
      n_estimators=100, max_depth=3)

rezultati = cross_val_score(
      model, ngram_ucenje, klase_ucenje)

print(f'Avg F1 {rezultati.mean():.3f}') # Avg F1 0.870

model.fit(ngram_ucenje, klase_ucenje)

predvidjene_klase = model.predict(ngram_test)
```



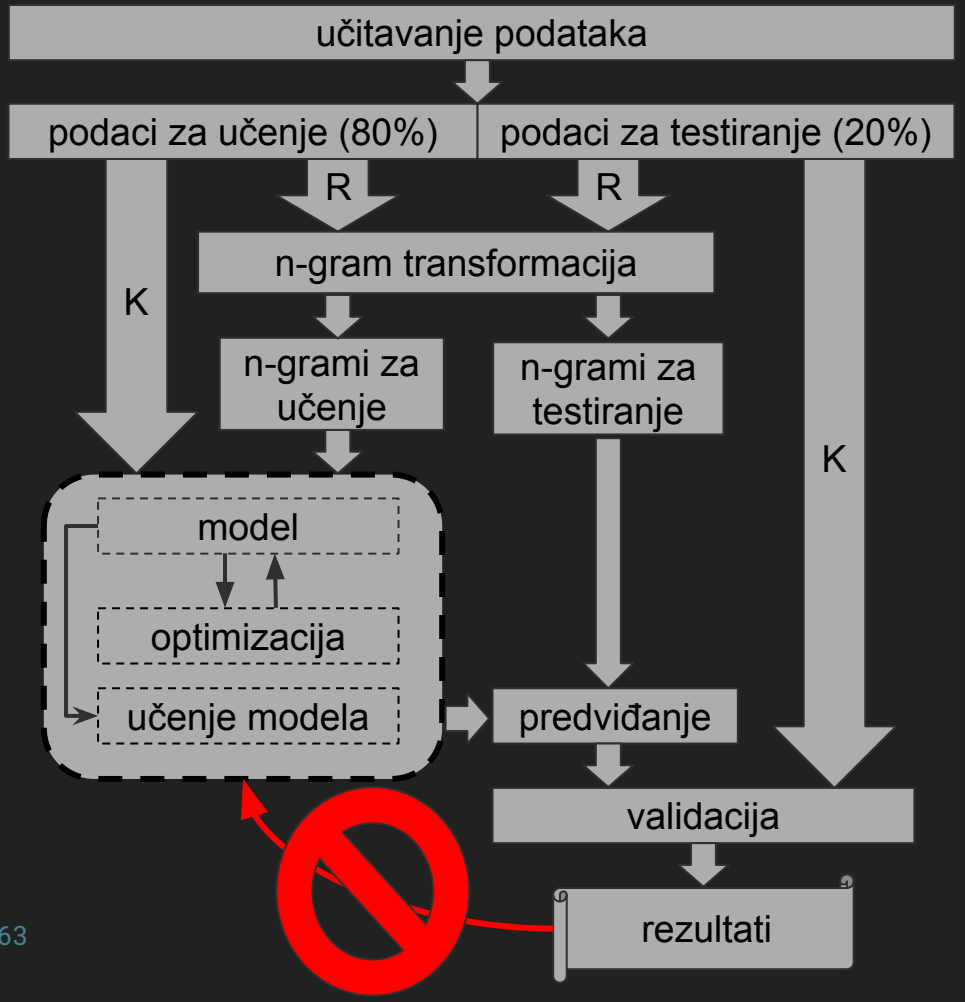
```python
podaci = pd.read_csv('podaci.csv')

rijeci_ucenje, rijeci_test, klase_ucenje, klase_test =\
    train_test_split(podaci['rijec'], podaci['klasa'],
    test_size=0.2)

transformator = CountVectorizer(
      ngram_range=(1, 3), analyzer='char')

ngram_ucenje = transformator.fit_transform(
      rijeci_ucenje)

ngram_test = transformator.transform(rijeci_test)

model = RandomForestClassifier(
      n_estimators=100, max_depth=3)

rezultati = cross_val_score(
      model, ngram_ucenje, klase_ucenje)

print(f'Avg F1 {rezultati.mean():.3f}') # Avg F1 0.870

model.fit(ngram_ucenje, klase_ucenje)

predvidjene_klase = model.predict(ngram_test)

rezultat = f1_score(klase_test, predvidjene_klase)
```



```python
podaci = pd.read_csv('podaci.csv')

rijeci_ucenje, rijeci_test, klase_ucenje, klase_test =\
    train_test_split(podaci['rijec'], podaci['klasa'],
    test_size=0.2)

transformator = CountVectorizer(
      ngram_range=(1, 3), analyzer='char')

ngram_ucenje = transformator.fit_transform(
      rijeci_ucenje)

ngram_test = transformator.transform(rijeci_test)

model = RandomForestClassifier(
      n_estimators=100, max_depth=3)

rezultati = cross_val_score(
      model, ngram_ucenje, klase_ucenje)

print(f'Avg F1 {rezultati.mean():.3f}') # Avg F1 0.870

model.fit(ngram_ucenje, klase_ucenje)

predvidjene_klase = model.predict(ngram_test)

rezultat = f1_score(klase_test, predvidjene_klase)

print(f'F1 rezultat: {rezultat:.3f}') # F1 rezultat: 0.863
```



```python
podaci = pd.read_csv('podaci.csv')

rijeci_ucenje, rijeci_test, klase_ucenje, klase_test =\
    train_test_split(podaci['rijec'], podaci['klasa'],
    test_size=0.2)

transformator = CountVectorizer(
      ngram_range=(1, 3), analyzer='char')

ngram_ucenje = transformator.fit_transform(
      rijeci_ucenje)

ngram_test = transformator.transform(rijeci_test)

model = RandomForestClassifier(
      n_estimators=100, max_depth=3)

rezultati = cross_val_score(
      model, ngram_ucenje, klase_ucenje)

print(f'Avg F1 {rezultati.mean():.3f}') # Avg F1 0.870

model.fit(ngram_ucenje, klase_ucenje)

predvidjene_klase = model.predict(ngram_test)

rezultat = f1_score(klase_test, predvidjene_klase)

print(f'F1 rezultat: {rezultat:.3f}') # F1 rezultat: 0.863
```



github.com/laconlab/lacon-workshop-2021