О задании

На сайтах для поиска работы можно найти сотни тысяч объявлений, каждое из которых состоит из пространного описания вакансии и предлагаемой зарплаты. Есть ли связь между описанием и зарплатой? Существуют ли определенные слова, которые наиболее сильно характеризуют зарплату? Можно ли найти другие информативные факторы? Вам предстоит ответить на эти вопросы, проанализировав выборку объявлений о работе в Великобритании.

Практическое задание 2 посвящено работе с текстовыми данными и категориальными признаками и задачам бинарной классификации. Вы научитесь:

- работать с категориальными признаками;
- строить вещественные представления текстовых данных;
- обучать и строить прогнозы линейных классификаторов при помощи scikit-learn и Vowpal Wabbit;
- тестировать модели и проводить оценку качества в задачах бинарной классификации.

Оценивание и штрафы

Каждая из задач имеет определенную «стоимость» (указана в скобках около задачи). Максимально допустимая оценка за работу — 10 баллов. Кроме того, некоторые из заданий являются опциональными (необязательными), однако за их выполнение можно получить дополнительные баллы, которые позднее будут учитываться при проставлении оценок автоматом по курсу.

Сдавать задание после указанного срока сдачи нельзя. При выставлении неполного балла за задание в связи с наличием ошибок на усмотрение проверяющего предусмотрена возможность исправить работу на указанных в ответном письме условиях.

Задание выполняется самостоятельно. «Похожие» решения считаются плагиатом и все задействованные студенты (в том числе те, у кого списали) не могут получить за него больше 0 баллов (подробнее о плагиате см. на странице курса). Если вы нашли решение какого-то из заданий (или его часть) в открытом источнике, необходимо указать ссылку на этот источник в отдельном блоке в конце Вашей работы (скорее всего вы будете не единственным, кто это нашел, поэтому чтобы исключить подозрение в плагиате, необходима ссылка на источник).

Неэффективная реализация кода может негативно отразиться на оценке.

Данные

Как было упомянуто ранее, в рамках данного задания мы будем решать задачу бинарной классификации для предсказания уровня заработной платы по тексту объявления о вакансии на примере набора данных с соревнования Adzuna - Job Salary Prediction (https://www.kaggle.com/c/job-salary-prediction). Для начала пройдите по ссылке (https://www.kaggle.com/c/job-salary-prediction/data) и скачайте файл Train_rev1 (при необходимости, зарегистрируйтесь на Kaggle).

Посмотрим на данные в файле и загрузим их в DataFrame:

In [1]:

```
%pylab inline

import pandas as pd
```

```
Populating the interactive namespace from numpy and matplotlib
```

In [2]:

```
# print first 2 rows from Train_rev1.csv
!head -n 2 "Train_rev1.csv"
```

In [3]:

```
df = pd.read_csv("Train_rev1.csv", sep=',', low_memory=True)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244768 entries, 0 to 244767
Data columns (total 12 columns):
Id                  244768 non-null int64
Title               244767 non-null object
FullDescription     244768 non-null object
LocationRaw         244768 non-null object
LocationNormalized  244768 non-null object
ContractType        65442 non-null object
ContractTime        180863 non-null object
Company             212338 non-null object
Category            244768 non-null object
SalaryRaw           244768 non-null object
SalaryNormalized    244768 non-null int64
SourceName          244767 non-null object
dtypes: int64(2), object(10)
memory usage: 22.4+ MB
```

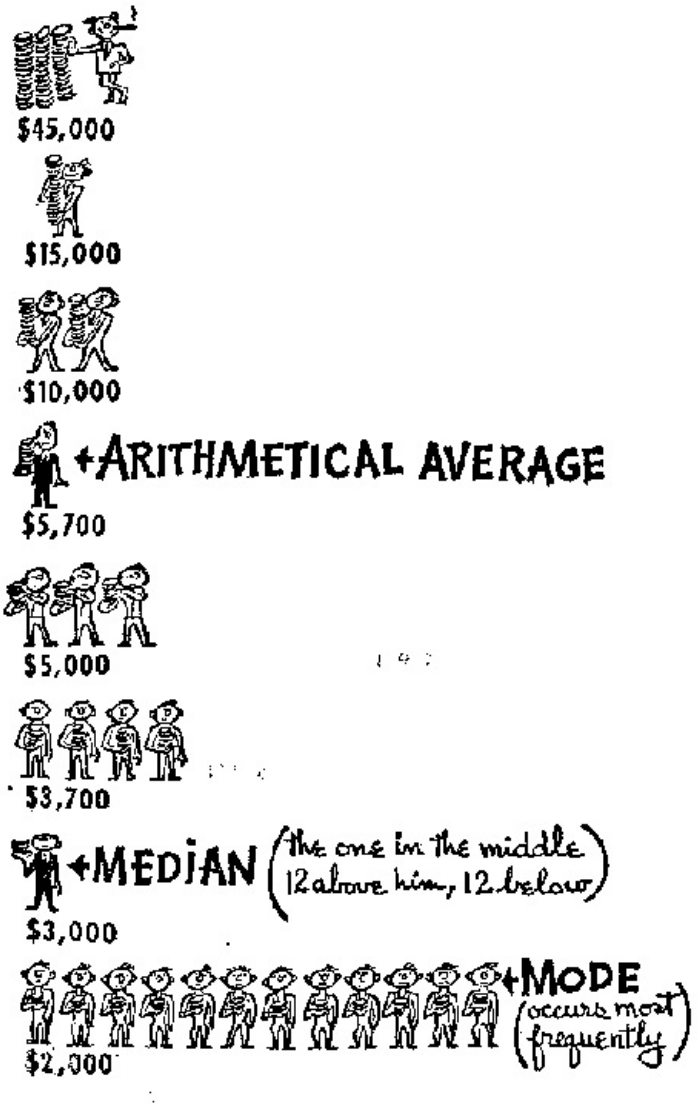
In [4]:

```
df.head(5)
```

Out[4]:

	Id	Title	FullDescription	LocationRaw	LocationNormalized	ContractType	ContractTime	Company	Category	
0	12612628	Engineering Systems Analyst	Engineering Systems Analyst Dorking Surrey Sal...	Dorking, Surrey, Surrey	Dorking	NaN	permanent	Gregory Martin International	Engineering Jobs	3(
1	12612830	Stress Engineer Glasgow	Stress Engineer Glasgow Salary **** to **** We...	Glasgow, Scotland, Scotland	Glasgow	NaN	permanent	Gregory Martin International	Engineering Jobs	3!
2	12612844	Modelling and simulation analyst	Mathematical Modeller / Simulation Analyst / O...	Hampshire, South East, South East	Hampshire	NaN	permanent	Gregory Martin International	Engineering Jobs	4(
3	12613049	Engineering Systems Analyst / Mathematical Mod...	Engineering Systems Analyst / Mathematical Mod...	Surrey, South East, South East	Surrey	NaN	permanent	Gregory Martin International	Engineering Jobs	3(
4	12613647	Pioneer, Miser Engineering Systems Analyst	Pioneer, Miser Engineering Systems Analyst Do...	Surrey, South East, South East	Surrey	NaN	permanent	Gregory Martin International	Engineering Jobs	3(

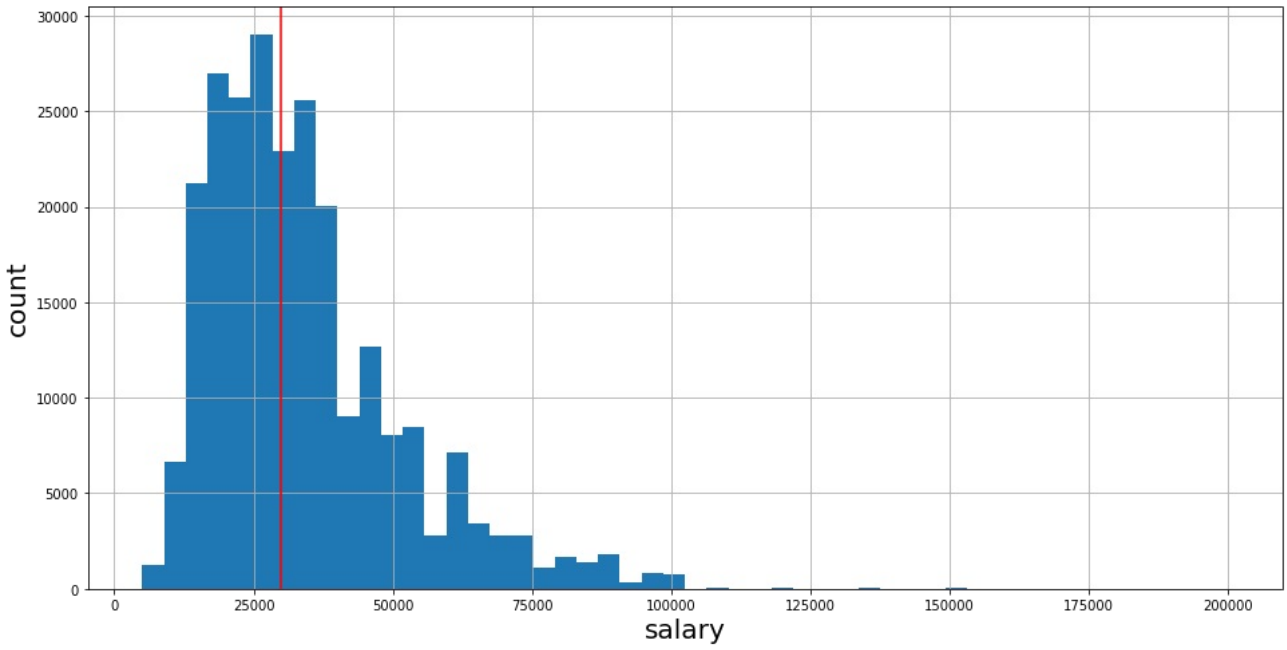
В оригинальной постановке предлагается рассматривать признак SalaryNormalized как целевой и решать задачу регрессии, однако в рамках данного задания мы сведём её к задаче бинарной классификации, разделив объекты на 2 группы: объявления о вакансиях с низкой и высокой зарплатами соответственно.



В качестве порога разбиения объектов на группы будем рассматривать медиану признака SalaryNormalized. Заметим, что таким образом мы автоматически получим задачу классификации со сбалансированными классами:

In [5]:

```
# salary histogramm
median = np.median(df['SalaryNormalized'])
figsize(16,8)
plt.hist(df['SalaryNormalized'], bins=50)
plt.axvline(median, c='r')
plt.xlabel('salary', fontsize=20)
plt.ylabel('count', fontsize=20)
plt.grid()
```



Осуществим последние шаги по подготовке датасета:

- бинаризуем признак SalaryNomalized по описанному ранее порогу;
- исключим из выборки признак SalaryRaw, чтобы устранить утечку целевой переменной в признаки.

In [6]:

```
df['SalaryNormalized'] = (df['SalaryNormalized'] > median).astype(int)
df.drop('SalaryRaw', axis=1, inplace=True)
df.head(5)
```

Out[6]:

	Id	Title	FullDescription	LocationRaw	LocationNormalized	ContractType	ContractTime	Company	Category	S
0	12612628	Engineering Systems Analyst	Engineering Systems Analyst Dorking Surrey Sal...	Dorking, Surrey, Surrey	Dorking	NaN	permanent	Gregory Martin International	Engineering Jobs	
1	12612830	Stress Engineer Glasgow	Stress Engineer Glasgow Salary **** to **** We...	Glasgow, Scotland, Scotland	Glasgow	NaN	permanent	Gregory Martin International	Engineering Jobs	
2	12612844	Modelling and simulation analyst	Mathematical Modeller / Simulation Analyst / O...	Hampshire, South East, South East	Hampshire	NaN	permanent	Gregory Martin International	Engineering Jobs	
3	12613049	Engineering Systems Analyst / Mathematical Mod...	Engineering Systems Analyst / Mathematical Mod...	Surrey, South East, South East	Surrey	NaN	permanent	Gregory Martin International	Engineering Jobs	
4	12613647	Pioneer, Miser Engineering Systems Analyst	Pioneer, Miser Engineering Systems Analyst Do...	Surrey, South East, South East	Surrey	NaN	permanent	Gregory Martin International	Engineering Jobs	

1. (0 баллов) Разбейте получившуюся выборку на обучающую и контрольную в соотношении 70/30 с использованием перемешивания объектов.

При разбиении используйте значение параметра random_state=42.

In [7]:

```
from sklearn.model_selection import train_test_split

#потом, сначала полностью подготовим df
df_train, df_test = train_test_split(df, random_state = 42, train_size = .7)
```

Векторизация

Как правило, модели, используемые в машинном обучении, применяются в предположении, что матрица "объект-признак" является вещественнозначной. Поэтому при работе с категориальными признаками и текстами сперва их необходимо привести к вещественному виду.

Заметим, что в нашей задаче есть признаки, являющиеся текстами произвольной природы (Title, FullDescription), и категориальные признаки, принимающие ограниченное число значений (ContractType, Category и др.).

Самый простой и понятный способ преобразования текстовых данных — векторизация. В этом случае для каждого слова, встречающегося в некотором набре текстов мы создаём отдельный новый признак, который будет равен 1, когда слово встречается в заданном объекте, и 0 – в противном случае.

2. (0.5 балла) Создайте текстовое описание объектов обучающей и контрольной выборок, объединив значения всех признаков каждого объекта выборки через символы пробела. После этого получите признаковое описание объектов, осуществив векторизацию получившихся текстов при помощи CountVectorizer (http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html), обучив его на обучающей выборке и применив на контрольной.

In [8]:

```
df_train['string_concat'] = df_train['Title'].str.cat(df_train['FullDescription'], sep =" ", na_rep = "").str.cat
(df_train['LocationRaw'], sep =" ", na_rep = "").str.cat(df_train['LocationNormalized'], sep =" ", na_rep = "").s
tr.cat(df_train['ContractType'], sep =" ", na_rep = "").str.cat(df_train['ContractTime'], sep =" ", na_rep = "").
str.cat(df_train['Company'], sep =" ", na_rep = "").str.cat(df_train['Category'], sep =" ", na_rep = "")
df_test['string_concat'] = df_test['Title'].str.cat(df_test['FullDescription'], sep =" ", na_rep = "").str.cat(df
_test['LocationRaw'], sep =" ", na_rep = "").str.cat(df_test['LocationNormalized'], sep =" ", na_rep = "").str.ca
t(df_test['ContractType'], sep =" ", na_rep = "").str.cat(df_test['ContractTime'], sep =" ", na_rep = "").str.cat
(df_test['Company'], sep =" ", na_rep = "").str.cat(df_test['Category'], sep =" ", na_rep = "")
df_train.drop(['Id', 'Title', 'FullDescription', 'LocationRaw', 'LocationNormalized', 'ContractType', 'ContractTi
me', 'Company', 'Category', 'SourceName'], axis=1, inplace=True)
df_test.drop(['Id', 'Title', 'FullDescription', 'LocationRaw', 'LocationNormalized', 'ContractType', 'ContractTim
e', 'Company', 'Category', 'SourceName'], axis=1, inplace=True)
```

In [9]:

```
from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer(analyzer='word')

#training vectors
vectors_train = vectorizer.fit_transform(df_train['string_concat'])
#test vectors
vectors_test = vectorizer.transform(df_test['string_concat'])
```

3. (1.5 балла) Обучите следующие модели на обучающей выборке:

- логистическую регрессию (http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html) из модуля sklearn с параметрами по умолчанию;
- логистическую регрессию при помощи Vowpal Wabbit с параметрами по умолчанию.

In [10]:

```
from sklearn.linear_model import LogisticRegression

logreg_clf = LogisticRegression(max_iter=1000000)

#training
logreg_clf.fit(vectors_train, df_train['SalaryNormalized'])
```

Out[10]:

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=1000000,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

In [11]:

```
#predict
predict = logreg_clf.predict(vectors_test)
```

In [12]:

```
from sklearn import metrics
```

In [13]:

```
print(metrics.confusion_matrix(predict, df_test['SalaryNormalized']))
```

```
[[33503  4480]
 [ 4437 31011]]
```

In [14]:

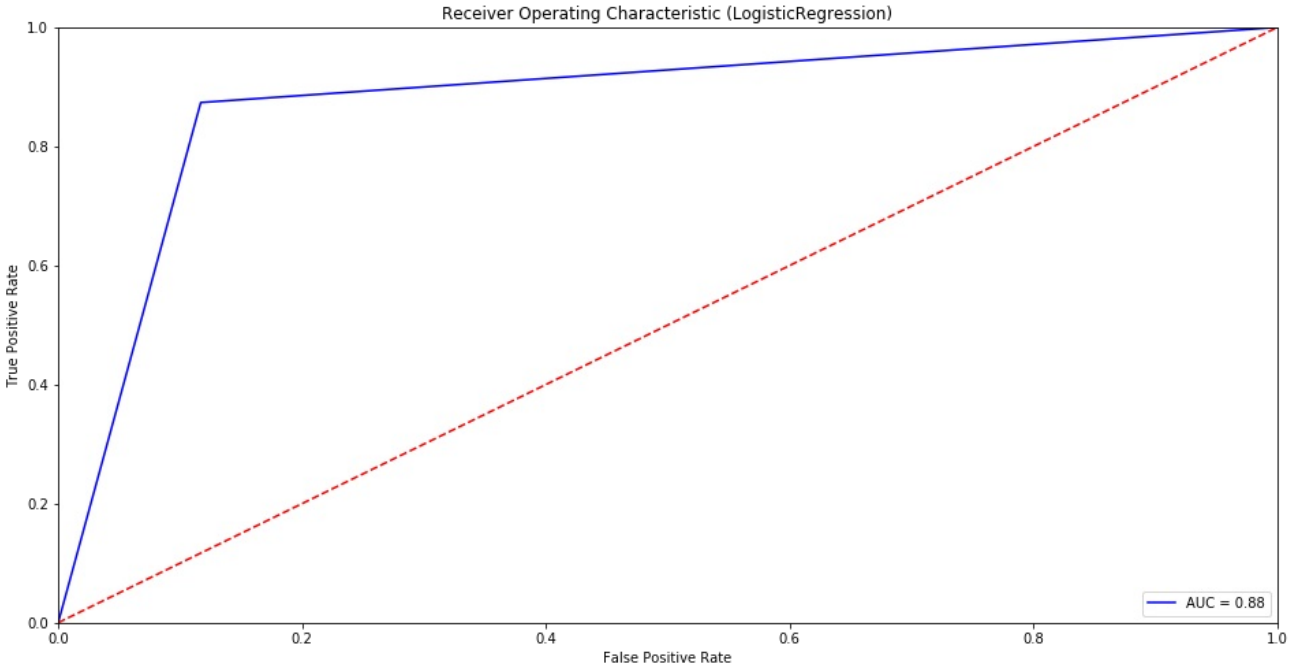
```
print(metrics.classification_report(predict, df_test['SalaryNormalized']))
```

```
              precision    recall  f1-score   support

           0       0.88      0.88      0.88     37983
           1       0.87      0.87      0.87     35448

    accuracy                           0.88     73431
   macro avg       0.88      0.88      0.88     73431
weighted avg       0.88      0.88      0.88     73431
```

4. (0.5 балла) Вычислите значения ROC-AUC, F-меры (http://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html), а также постройте матрицу ошибок (http://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html) для каждой из построенных в п. 3 моделей на контрольной выборке. Сравните построенные модели по качеству их работы.

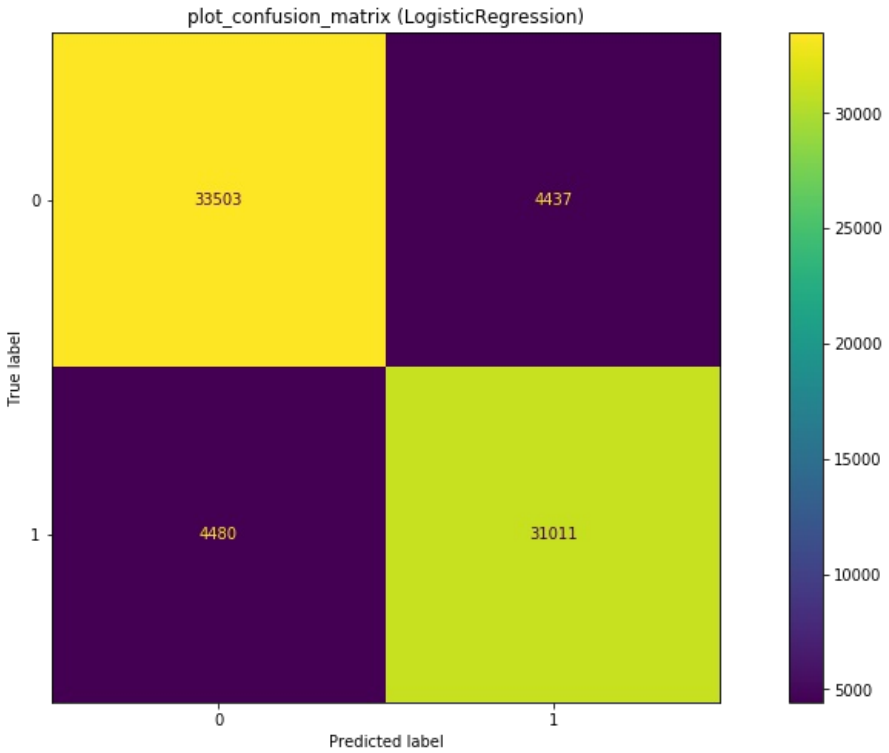
In [15]:

```
#calculate the fpr and tpr for all thresholds of the classification
fpr, tpr, threshold = metrics.roc_curve(df_test['SalaryNormalized'], predict)
roc_auc = metrics.auc(fpr, tpr)

plt.title('Receiver Operating Characteristic (LogisticRegression)')
plt.plot(fpr, tpr, 'b', label = 'AUC = %0.2f' % roc_auc)
plt.legend(loc = 'lower right')
plt.plot([0, 1], [0, 1], 'r--')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```



In [16]:

```
from sklearn.metrics import plot_confusion_matrix

plot_confusion_matrix(logreg_clf, vectors_test, df_test['SalaryNormalized'], values_format = "d")
plt.title('plot_confusion_matrix (LogisticRegression)')
plt.show()
```



5. (1 балл) Отсортируйте веса признаков для модели логистической регрессии из scikit-learn, полученной в п. 2. Какие слова из встречающихся в выборке имеют наибольшее/наименьшее влияние на значение целевой переменной? Проинтерпретируйте полученный результат.

In [17]:

```
#vectorizer
bag_of_words = vectors_train
sum_words = bag_of_words.sum(axis=0)
words_freq = [(word, sum_words[0, idx]) for word, idx in     vectorizer.vocabulary_.items()]
words_freq = sorted(words_freq, key = lambda x: x[1], reverse=True)
```

In [18]:

```
print(words_freq[:10])
```

```
[('and', 1860506), ('the', 1462032), ('to', 1411004), ('of', 1003516), ('in', 725028), ('for', 60610
4), ('with', 508845), ('you', 485590), ('will', 456485), ('be', 437344)]
```

In [19]:

```
print(words_freq[-10:])
```

```
[('rolelocation', 1), ('siclops', 1), ('jwdcm134', 1), ('corporatefocused', 1), ('vickyweareadam', 1
), ('worksteam', 1), ('applicationssupportconsultantsoftwaresupport_job', 1), ('plasticsprocurementd
irector_job', 1), ('oscrum', 1), ('prmarketingofficercharityhospice_job', 1)]
```

6. (0.5 доп. балла) Отсортируйте веса признаков для модели логистической регрессии, полученной в п. 2 при помощи Vowpal Wabbit. Какие слова из встречающихся в выборке имеют наибольшее/наименьшее влияние на значение целевой переменной? Проинтерпретирйте полученный результат.

In [ ]:

```
# Your code here
```

TF-IDF

Ещё один способ работы с текстовыми данными — TF-IDF (https://en.wikipedia.org/wiki/Tf–idf) (**T**erm **F**requency–**I**nverse **D**ocument **F**requency). Рассмотрим коллекцию текстов D. Для каждого уникального слова t из документа $d \in D$ вычислим следующие величины:

1. Term Frequency – количество вхождений слова в отношении к общему числу слов в тексте:
$$\mathrm{tf}(t, d) = \frac{n_{td}}{\sum_{t \in d} n_{td}},$$
где n_{td} — количество вхождений слова t в текст d.
2. Inverse Document Frequency
$$\mathrm{idf}(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|},$$
где $|\{d \in D : t \in d\}|$ – количество текстов в коллекции, содержащих слово t.

Тогда для каждой пары (слово, текст) (t, d) вычислим величину:
$$\text{tf-idf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D).$$

Отметим, что значение $\mathrm{tf}(t, d)$ корректируется для часто встречающихся общеупотребимых слов при помощи значения $\mathrm{idf}(t, D)$.

Признаковым описанием одного объекта $d \in D$ будет вектор $\Big(\text{tf-idf}(t, d, D)\Big)_{t \in V}$, где V – словарь всех слов, встречающихся в коллекции D.

7. (0.5 балла) Создайте текстовое описание объектов обучающей и контрольной выборок, объединив значения всех признаков каждого объекта выборки через символы пробела. После этого получите признаковое описание объектов, вычислив вектор tf-idf для каждого объекта помощи TfidfVectorizer (http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html), обучив его на обучающей выборке и применив на контрольной.

In [ ]:

```
# Your code here
```

8. (0 баллов) Обучите следующие модели на обучающей выборке:

- логистическую регрессию (http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html) из модуля sklearn с параметрами по умолчанию;
- логистическую регрессию при помощи Vowpal Wabbit с параметрами по умолчанию.

In [ ]:

```
# Your code here
```

9. (0.5 балла) Вычислите значения ROC-AUC, F-меры (http://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html), а также постройте матрицу ошибок (http://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html) для каждой из построенных в п. 8 моделей на контрольной выборке. Сравните построенные модели по качеству их работы.

In [ ]:

```
# Your code here
```

10. (0.5 балла) Сравните значения метрик из п. 9 со значениями, полученными в п. 5, и сравните соответствующие модели по качеству из работы.

Ответ:

11. (1 балл) Отсортируйте веса признаков для модели логистической регрессии из scikit-learn, полученной в п. 8. Какие слова из встречающихся в выборке имеют наибольшее/наименьшее влияние на значение целевой переменной? Проинтерпретирйте полученный результат.

In [ ]:

```
# Your code here
```

12. (0.5 доп. балла) Отсортируйте веса признаков для модели логистической регрессии, полученной в п. 8 при помощи Vowpal Wabbit. Какие слова из встречающихся в выборке имеют наибольшее/наименьшее влияние на значение целевой переменной? Проинтерпретирйте полученный результат.

In [ ]:

```
# Your code here
```

Счётчики

Ранее в рамках данного задания при построении моделей мы объединяли значения всех признаков в единую строку, что предполагает равноправность всех признаков. Однако заметим, что в этом случае мы допускаем потерю информации: слово "Glasgow" может по-разному влиять на зарплату, если оно находится в названии объявления и в геолокации. Чтобы устранить этот недостаток, при создании текстового описания объекта будем объединять только значения признаков Title и FullDescription, а остальные будем рассматривать как категориальные. При этом с полученным текстовым описанием объекта будем работать, как раньше (при помощи векторизации или tf-idf), а для кодирования категориальных признаков используем **счётчики**.

Идея этого метода состоит в том, чтобы заменить значение категориального признака на вероятность того, что объект с данным значением признака относится к положительному классу. Опишем эту идею более формально. Пусть у нас есть выборка $X = \{(x_i, y_i)\}_{i=1}^{l}$, и j-ый признак принимает значения из множества $U_j = \{u_{jn}\}_{n=1}^{N_j}$, где N_j — количество различных значений j-ого признака. Пусть $x_{ij} = u_{jn}$, тогда заменим значения j-ого категориального признака объекта x_i на следующую оценку:

$$P(y_i = +1 \quad |$$

Однако заметим, что при таком способе формирования счётчиков мы учитываем в формуле для объекта x_i его метку y_i, тем самым вносим информацию об ответе в признаки. Чтобы устранить этот недостаток, при вычислении счётчика будем исключать из рассмотрения текущий объект, т.е. рассматривать следующую оценку:

$$P(y_i = +1 \quad |$$

13. (0.5 балла) Создайте текстовое описание объектов обучающей и контрольной выборок, объединив значения признаков Title и FullDescription каждого объекта выборки через символ пробела, после чего перейдите к признаковому описанию объектов, вычислив вектор tf-idf аналогично п. 7.

In [ ]:

```
# Your code here
```

14. (1 балл) Закодируйте категориальные признаки (все, кроме Title и FullDescription) при помощи one-hot encoding (http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html), после чего обучите логистическую регрессию (при помощи scikit-learn или Vowpal Wabbit) на обучающей выборке. Вычислите значения ROC-AUC, F-меры (http://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html), а также постройте матрицу ошибок (http://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html) для полученной модели на контрольной выборке.

In [ ]:

```
# Your code here
```

15. (2 балла) Для выборки, полученной в п. 13, закодируйте категориальные признаки (все, кроме Title и FullDescription) при помощи счётчиков, после чего обучите логистическую регрессию (при помощи scikit-learn или Vowpal Wabbit) на обучающей выборке. Вычислите значения ROC-AUC, F-меры (http://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html), а также постройте матрицу ошибок (http://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html) для полученной модели на контрольной выборке.

Уделите внимание оптимальности вычисления счётчиков!

In [ ]:

```
# Your code here
```

16. (0.5 балла) Сравните значения метрик из п. 15 со значениями, полученными в п. 14, и сделайте вывод о качестве классификации для каждого из методов кодирования категориальных признаков.

Ответ:

Подбор гиперпараметров

17. (1.5 доп. балла) Разбейте обучающую выборку, полученную в п. 13, на обучающую и валидационную в отношении 80/20, после чего подберите оптимальное количество фолдов, используемое при кодировании категориальных признаков (всех, кроме Title и FullDescription), путём оптимизации значения accuracy на валидационной выборке. Используйте следующие модели, аналогично также подобрав оптимальные значения указанных гиперпараметров:

- логистическую регрессию из модуля sklearn с подбором коэффициента регуляризации;
- логистическую регрессию при помощи Vowpal Wabbit с подбором следующих гиперпараметров:
  - коэффициент регуляризации (--l2);
  - количество эпох (--passes);
  - длина градиентного шага (-l);
  - длина N-грамм (--ngram).

In [ ]:

```
# Your code here
```

18. (0.5 доп. балла) Обучите указанные выше модели на обучающей выборке для оптимальных значений гиперпараметров, найденных в п. 17, после чего для каждой из моделей вычислите значения ROC-AUC, F-меры, а также постройте матрицу ошибок на контрольной выборке. Как качество классификации при помощи полученных в данном разделе моделей соотносится с моделями, полученными в предыдущих разделах?

In [ ]:

```
# Your code here
```

Здесь вы можете поделиться своими мыслями по поводу этого задания.

А здесь — вставить вашу вторую любимую смешную картинку.

А здесь — посоветовать преподавателям хороший фильм или сериал.

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```